



**UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO**

**FACULTAD DE INGENIERIA**

**IMPLEMENTACION DE UNA PROGRAMOTECA  
DE TIPO INTERACTIVO PARA APOYO A  
LA ASIGNATURA DE DISEÑO LOGICO**

**T E S I S**

**QUE PARA OBTENER EL TITULO DE  
INGENIERO MECANICO ELECTRICISTA**

**P R E S E N T A N**

**MA. GUADALUPE ROSALIA ALDANA JAMAICA**

**FRANCISCO JAVIER LOPEZ ORTIZ**

**DIRECTOR DE TESIS:**

**ING. J. ANTONIO PATIÑO RAMIREZ**

**MEXICO, D. F.**

**1984**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## INDICE GENERAL

T. INTRODUCCION	1
II. LA COMPUTADORA COMO AUXILIAR EN EL PROCESO ENSEÑANZA-APRENDIZAJE	3
III. CARACTERISTICAS DE LOS LENGUAJES USADOS	
III.1 PASCAL	4
III.2 FORTRAN 77	7
IV. MAPAS DE KARNIUGH	
IV.1 Acciones	
IV.1.A PASCAL	8
IV.1.B FORTRAN 77	15
IV.2 Programas fuente y corridas	24.RTS
IV.3 Manual de Usuario	25
V. DITNE MC. CLISKEY	
V.1 Acciones	
V.1.A PASCAL	28
V.1.B FORTRAN 77	33
V.2 Programas fuente y corridas	39.RTS
V.3 Manual de usuario	40

UT. DITNE MC.CIUSEKY MII TTPI ES SAI TRAS

UT.1 Acciones	43
UT.2 Programa fuente y corrida	46.RTS
UT.3 Manual de usuario	47

UTT. SECUENCIALES

UTT.1 Acciones	51
UTT.2 Programa fuente y corrida	58.RTS
UTT.3 Manual de usuario	59

UUTT. CONCLUSIONES

ANEXO A	67
ANEXO B	70
ANEXO C	76

## I. INTRODUCCION

Debido a la importancia de efectuar diseños economicos, es necesario dominar las tecnicas de simplificacion de funciones de conmutacion dada la correlacion basica que existe entre formas algebraicas simples y diseño economico.

Como una forma de apoyo al aprendizaje de las tecnicas de simplificacion de funciones de conmutacion (metodo grafico de Karnaugh y metodo tabular de Quine - Mc.Cluskey) y su aplicacion en la tecnica de reduccion de estados de circuitos secuenciales, se diseña un sistema de programas interactivos para computadora que tiene como objeto proporcionar un medio que guie al usuario en la solucion de ejercicios propuestos por el mismo.

Como no es la finalidad del sistema la enseñanza de los metodos, el usuario debera ya conocer la teoria y usar el sistema para practicar lo aprendido, es decir, podra entonces plantear un problema en el momento mismo de estar frente a la terminal e intentar resolverlo, o bien, proponer uno ya antes resuelto por el y resolverlo ahora bajo la "supervision" de los programas del sistema.

El sistema abarca cuatro subsistemas:

- a) Minimizacion de funciones a traves del metodo grafico de Mapas de Karnaugh. (Capitulo IV MAPAS DE KARNAUGH)
- b) Minimizacion de funciones a traves del metodo tabular de Quine - Mc.Cluskey. (Capitulo V QUINE MC.CLUSKEY)
- c) Minimizacion de funciones a traves del metodo tabular de Quine - Mc.Cluskey para circuitos de salidas multiples. (Capitulo VI QUINE MC.CLUSKEY MULTIPLES SALIDAS)
- d) Reduccion de estados secuenciales e implementacion con Flip Flop JK, SR, D, T. (Capitulo VII SECUENCIALES)

que se describen en los capitulos correspondientes.

Cada capítulo esta dividido en cuatro incisos:

1.- ACCIONES:

- Descripción de la función que lleva a cabo el programa principal y las subrutinas del subsistema correspondiente.
- Diagrama Jerarquico.

2.- Programa(s) fuente y corrida(s) correspondientes.

3.- Manual de usuario

A fin de que el sistema pueda ser versatil para diferentes equipos de computo, se desarrollo en dos versiones, en cuanto a diseño, lenguaje de programación y equipo utilizado. Una de las versiones fue implementada en un equipo BURROUGHS 7800, en el cual se utilizo el lenguaje PASCAL; la otra version se implemento en una BURROUGHS 1885 con el lenguaje FORTRAN 77.

## II. LA COMPUTADORA COMO AUXILIAR EN EL PROCESO ENSEÑANZA-APRENDIZAJE

Ultimamente, la didáctica ha venido ensayando el uso de computadoras en la enseñanza, con resultados promisorios. Así encontramos entre otros trabajos, los realizados por:

- Hellmut Broscha (Instituto de Colaboración Científica Tubingen)
- Sistema Plato II
- Elischa Rhodes Huggins
- Proyecto PLAN (Sistema de Educación Individualizado)
- "ERE" (Edson Responsive Environment)
- Sistema "Bakkalaureus" (Instituto de Colaboración Científica Alemana), con la computadora "NIXDORF 20"
- Distributed Instructional Support System (Lamar University)

Debido a la revolución tecnológica que ha dado lugar a la evolución acelerada de los computadores, es de esperarse una mayor acción sobre el campo educativo.

El computador jamás será un sustituto docente ya que las necesidades educativas van encaminadas a una actividad reflexiva, sin embargo, en lo que respecta a las situaciones de adiestramiento será un auxiliar extraordinario.

Un esquema general de enseñanza por medio de computadoras es:

- a) En el plan de curso de una disciplina se señala lo que será tratado por el profesor en clase y lo que será interactivo con el computador.
- b) El profesor, en clase, motiva la materia a estudiar y provoca la reflexión entre el grupo, llevándolo en el proceso de descubrimiento.
- c) Una vez planteados los mecanismos de acción, el alumno se dirige al computador a verificar sus acciones.

Partiendo de este esquema es como se ha definido este trabajo de tesis.

Nuestro punto de enfoque es el adiestramiento en el proceso de optimización (razón de ser del ingeniero) de funciones obtenidas en el proceso de Diseño Lógico.

A continuación se consignan las ventajas y desventajas más evidentes del uso de las computadoras en la enseñanza.

- a) El costo de la enseñanza por medio de computadoras se calcula en menos de la mitad del de la enseñanza tradicional.
- b) Posibilidad real de individualizar la enseñanza.
- c) Mayor eficiencia en la enseñanza.
- d) Mayor motivación por parte del educando.
- e) Estudio al ritmo propio de cada educando.
- f) Eliminación del inconveniente de la incompatibilidad entre maestro y alumno y entre este y sus discípulos.
- g) Mayor actividad y participación del educando en el proceso de aprendizaje.
- h) Control constante del proceso de aprendizaje por medio de la respuesta inmediata de "CORRECTO" o "ERRONEO" con relación al trabajo del educando, lo cual provoca intensa satisfacción.
- i) Posibilidad de programaciones destinadas a los docentes, tendiendo a un constante perfeccionamiento de las mismas, en un proceso educativo permanente.
- j) Posibilidad de repeticiones de programas.
- k) Posibilidad de que el educando estudie en el horario que le sea conveniente.



**DESVENTAJAS**

- a) Elevado costo de los aparatos, al inicio.
- b) Mayores posibilidades de instruir que de formar
- c) Enseñanza estrictamente individualizada, que no favorece la socialización.

### III. CARACTERISTICAS DE LOS LENGUAJES USADOS

#### III.1 PASCAL

- Es un lenguaje de programación de alto nivel que usa todas las formas básicas de estructuración (la secuencia, la decisión, la estructura repetitiva y el procedimiento).
- Es compacto, esto es, existe un número relativamente pequeño de construcciones básicas, sin embargo, PASCAL es poderoso debido a que pueden combinarse todas ellas de muchas maneras.
- Sus características se definieron sin referirse a una máquina en particular para poder hacer sus programas transportables, aunque no existe un PASCAL estandar universalmente aceptado.

#### FACILIDADES PARA EL SISTEMA ELABORADO.

- Se trataron de usar instrucciones estandar de PASCAL REPORT para hacer transportable el sistema.
- Se aprovecho la facilidad de manejo de conjuntos con que cuenta este lenguaje.
- Se utilizaron los tipos estructurados:
  - a) De arreslos
  - b) De Registro

#### DESVENTAJAS DEL LENGUAJE

- PASCAL proporciona facilidades automáticas de conversión para los archivos de texto, pero no pueden ser usadas en programas reales porque no hay previsión para condiciones de error.
- El tamaño de un arreslo es definido en el tiempo de compilación, esto es, no hay dimensionamiento dinámico.
- No existe construcción de salida de un ciclo, esto es, si se desea salir de un ciclo, la única manera en que se puede hacer es introduciendo variables booleanas falsas o usar proposiciones GOTO.

### III.2 FORTRAN 77

- Es un lenguaje de programación útil en la solución de problemas de tipo científico.

- A diferencia de FORTRAN IV, FORTRAN 77 es estructurado, esto hace que los programas elaborados con este lenguaje, sean bastante más claros que los elaborados con su antecesor.

Facilidades para el sistema elaborado.

- Se pueden leer datos numéricos como alfabéticos y viceversa sin que genere error por esta causa, dejando al programa su verificación.

Desventajas del lenguaje

- No tiene la construcción de conjuntos como sucede con PASCAL.

- No existe la construcción de ciclos de salida.

- No existe dimensionamiento dinámico.

## IV. MAPAS DE KARNAUGH

### IV.1 ACCIONES

#### IV.1.A. PASCAL

En este capítulo se presenta el programa que guía al usuario en la minimización de funciones a través del método de Mapas de Karnaugh, a continuación se describe en forma sencilla la tarea de las subrutinas, del programa principal y la conexión entre ellos. Un diagrama Jerárquico de él se muestra en la Fig. IV.1.A

El programa principal organiza el uso de subrutinas que utilizara para pedir al usuario la función a minimizar, validarla y resolverla. Además valida proposiciones de minimización indicando si son correctas o no. Muestra la solución que obtuvo el usuario, haciéndole saber si es óptima, si no lo fue, tiene la opción de tratar de llegar a ella o bien de conocerla.

El cometido de cada subrutina se describe a continuación, sus parámetros se definen con el siguiente formato:

NOMBRE DEL PARAMETRO: TIPO <CONTENIDO>

#### SUBROUTINAS

##### - TEXTO

Contiene una pequeña introducción al método, la cual puede conocer el usuario si así lo desea.

##### - HORIZONTAL2

Variables de entrada/salida

MAPEO: Entero escalar <mapa con todos los minterminos posibles>

Intercambia los renglones 3 y 4 del mapa original de tal manera que se tensan minterminos adyacentes para poder reducirlos más fácilmente. Se usa cuando se tienen mapas de más de 3 variables.

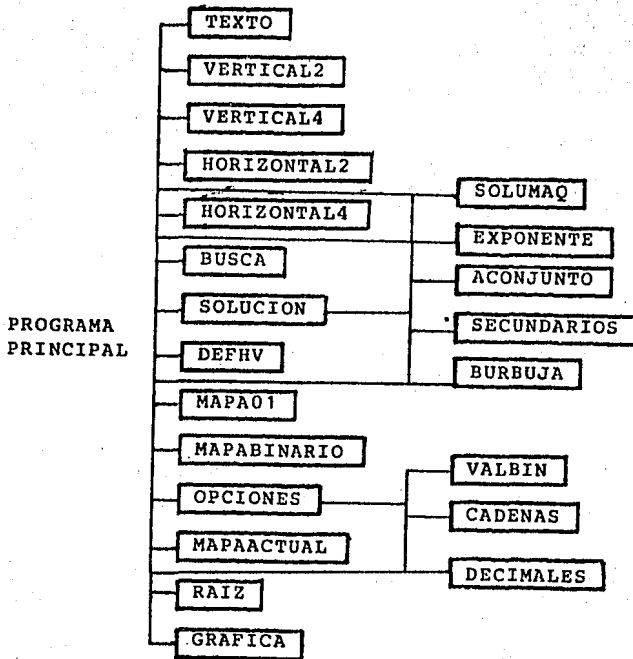


FIG. IV.1.A DIAGRAMA JERARQUICO

## - HORIZONTAL 4

Variable de entrada/salida

MAPEO: Entero escalar &lt;mapa con todos los minterminos posibles&gt;

Intercambia los renglones 7 con 8 del mapa original de tal manera que se tensan minterminos adyacentes para poder reducirlos mas facilmente. Se usa cuando se tienen mapas de 6 variables.

## - VERTICAL2

Variable de entrada/salida

MAPEO: Entero escalar &lt;mapa con todos los minterminos posibles&gt;

Intercambia las columnas 3 y 4 del mapa original de tal manera que se tensan minterminos adyacentes para poder reducirlos mas facilmente. Se usa con mapas de mas de 2 variables.

## - VERTICAL4

Variable de entrada/salida

MAPEO: Entero escalar &lt;mapa con todos los minterminos posibles&gt;

Intercambia las columnas 7 y 8 del mapa original de tal manera que se tensan minterminos adyacentes para poder reducirlos mas facilmente. Se usa con mapas de mas de 2 variables.

## - EXPONENTE

Variable de entrada

N: Entero escalar &lt;exponente de la potenciacion, numero de cubos-0&gt;

Obtiene la funcion  $F(x)=2^{xN}$ . Con este valor se puede saber la numero de elementos que tiene un cubo.

## - RAIZ

Variable de entrada

NUMERO: Entero escalar &lt;numero del que se obtiene la raiz&gt;

Obtiene  $x$  de la funcion  $N^{x(1/x)}=2$ . Se usa para obtener el renglon en que se buscan los cubos que propone el usuario para minimizarse.

## - GRAFICA

Variables de entrada

MAPEO: Entero vectorial <mapa conteniendo todas los posibles minterminos para el numero de variables>  
 MAPAOMIN: Entero vectorial <mapa conteniendo los minterminos de la funcion>  
 CHAPAOMIN: Caracter vectorial <mapa donde se marcan los minterminos que va cubriendo el usuario>

Grafica los mapas de posiciones, minterminos y marcas.

## - DEFHV

Variable de entrada

NVARIA: Entero escalar <numero de variables de la funcion>

Variables de salida

NI: Entero escalar <numero de renglones del mapa>

NJ: Entero escalar <numero de columnas del mapa>

Define el numero de renglones y columnas del mapa de Karnaugh dependiendo del numero de variables de la funcion asi como la posicion de las mismas en el mapa.

## - ACONJUNTO

Variables de entrada

CUBRE: Caracter vectorial <marcas de la tabla de implicantes secundarios>

NDIGS: Entero escalar <numero de minterminos de implicantes secundarios>

NIMP: Entero vectorial < numero de implicantes secundarios>

NN: Entero vectorial <implicantes secundarios>

Variable de salida

CONT: Entero escalar <numero de productos o cubos que cubren los implicantes secundarios>

Si existen implicantes secundarios, esta subrutina es llamada por el Programa Principal para obtenerlos, a traves, del metodo de Petrick (Anexo A), los cuales unidos a los implicantes esenciales, generan todas las posibles soluciones y de ellas se marcan aquellas que dan soluciones optimas.

**- SECUNDARIOS**

Variable de entrada

J: Entero, escalar &lt;posicion de la tabla de los implicantes no esenciales&gt;

Variable de salida

II: Entero, escalar &lt;contador del numero de implicantes secundarios que se van acumulando&gt;

Va construyendo la tabla de implicantes primos secundarios con los renglones J (de la primera tabla) que se envian a esta subrutina y lleva un conteo de ellos.

**- SOLUMAQ**

Variables de entrada

I: Entero escalar &lt;posicion del implicante en la tabla&gt;

MATRIZ: Entero escalar &lt;contiene los implicantes primos que entran en la solucion&gt;

NELEMEN: Entero escalar &lt;numero de cubos del implicante enviado a esta subrutina&gt;

Variable de salida

CONT: Entero escalar &lt;acumula el numero de implicantes de una solucion&gt;

Elimina las variables cuyo valor binario es diferente en los cubos que integran un implicante que contiene la solucion, esto quiere decir que solo aquellas que se mantienen constantes forman parte de la solucion. Cada uno de estos implicantes, en funcion de sus variables se almacena en un renglon (CONT).

**- BURBUJA**

Variables de entrada/salida

DIGITOS: Entero escalar &lt;miniterminos que se ordenan&gt;

NDIGS: Entero escalar &lt;numero de miniterminos que se ordenan&gt;

Ordena n miniterminos por el metodo de burbuja.

**- SOLUCION**

Subrutina fundamental del programa, en ella se resuelve el mapa de Karnaugh; se explica ampliamente en el Anexo A.



## - DECIMALES

Variables de salida

DIGS: Entero escalar <minterminos leídos>

NDIGITOS: Entero escalar <numero de minterminos leídos>

Lee y valida los minterminos de la funcion en forma decimal, ubicandolos en el arreglo DIGS, el total de ellos en NDIGITOS.

## - CADENAS

Variable de entrada

VBIN: entero escalar <valor asociado a las variables por su posicion>

Variables de salida

DIGS: Entero escalar <minterminos leídos ya convertidos a su forma decimal>

NDIGS: Entero escalar <numero de minterminos leídos>

Lee y valida los minterminos de la funcion en forma canonica y los convierte a su equivalente decimal, ubicandolos en el arreglo

## - VALBIN

Variable de entrada

VBIN: entero escalar <valor asociado a las variables por su posicion>

Variables de salida

DIGS: Entero escalar <minterminos leídos ya convertidos a su equivalente decimal>

NDIGS: Entero escalar <numero de minterminos leídos>

Lee y valida los minterminos de la funcion en forma binaria y los convierte a su equivalente decimal

## - OPCIONES

Variables de salida

DIGITOS: Entero escalar <minterminos de la funcion>

NDIGITOS: Entero escalar <numero de minterminos de la funcion>

Explica las formas en que el usuario puede proporcionar los minterminos (decimal, canonica, binaria), se auxilia de las subrutinas DECIMALES, CADENAS, VALBIN para leerlos.

## - MAPAActual

Actualiza las posiciones de los posibles minterminos del mapa de Karnaugh de acuerdo a las posiciones de sus variables.

- MAPABINARIO

Obtiene el valor binario de todos los posibles minterminos del mapa.

- MAPAS01

Define el arreglo MAPASMIN con los minterminos de la funcion y marca en CMAPASMIN los minterminos que va cubriendo el usuario.

- BUSCA

Variables de entrada

INICIO: Entero escalar <acotador inferior para busqueda>

FIN: Entero escalar <acotador superior para busqueda>

T: Entero escalar <renglon en que buscara un cubo>

Variables de salida

UBICA: Entero escalar <columna de la matriz en que encontro el cubo>

En la matriz donde se encuentran todos los cubos, busca aquellos que propone el usuario para minimizarse. En funcion de T ubica los renglones en los que se debe iniciar la busqueda; INICIO y FIN acotan el numero de elementos del cubo buscado. Con UBICA y T se conoce la columna y renglon en que se encontro el cubo. En caso de no encontrarlo, la variable global BANDERA toma el valor FALSO.

## IV.1.B FORTRAN 77

En este inciso se describe otro subsistema que asesora al usuario en la minimización de funciones de conmutación usando el método de Mapas de Karnaugh, empleando otras estructuras de datos, algoritmos y lenguaje de programación.

El subsistema está formado por un programa principal que organiza el uso de diecinueve subrutinas, como se muestra en un Diagrama Jerárquico en la Fig. IV.1.B.1.

La secuencia de llamado a las subrutinas se presenta en la Fig. IV.1.B.2. Se puede observar de esta figura, que el acceso a algunas subrutinas es opcional.

A continuación se describe la función de cada subrutina; se indican las variables de entrada/salida, así como los mensajes usados para comunicarse con el usuario.

## - NUMVAR

Variable de salida: N

Requiere al usuario el número de variables que tiene la función de conmutación que desea minimizar.

Mensajes que requieren respuesta del usuario.

- Cuántas variables tiene la función? ( $2 \leq N \leq 6$ )

Mensajes de error al usuario:

- Tipo de dato erróneo,  $N \Rightarrow$  no numérica.

- Valor fuera de rango.

## - RENCOL

Variable de entrada: N

Variables de salida: NREN, MCOL

Calcula el número de renglones y columnas del mapa de Karnaugh a partir del número de variables de la función mediante las siguientes fórmulas:

$NREN = 2^{**}M$ , donde M = parte entera de  $N/2$

Si existe parte fraccionaria de la división  $N/2$ :

$MCOL = 2^{**}(M+1)$

en caso contrario:

$MCOL = 2^{**}M$

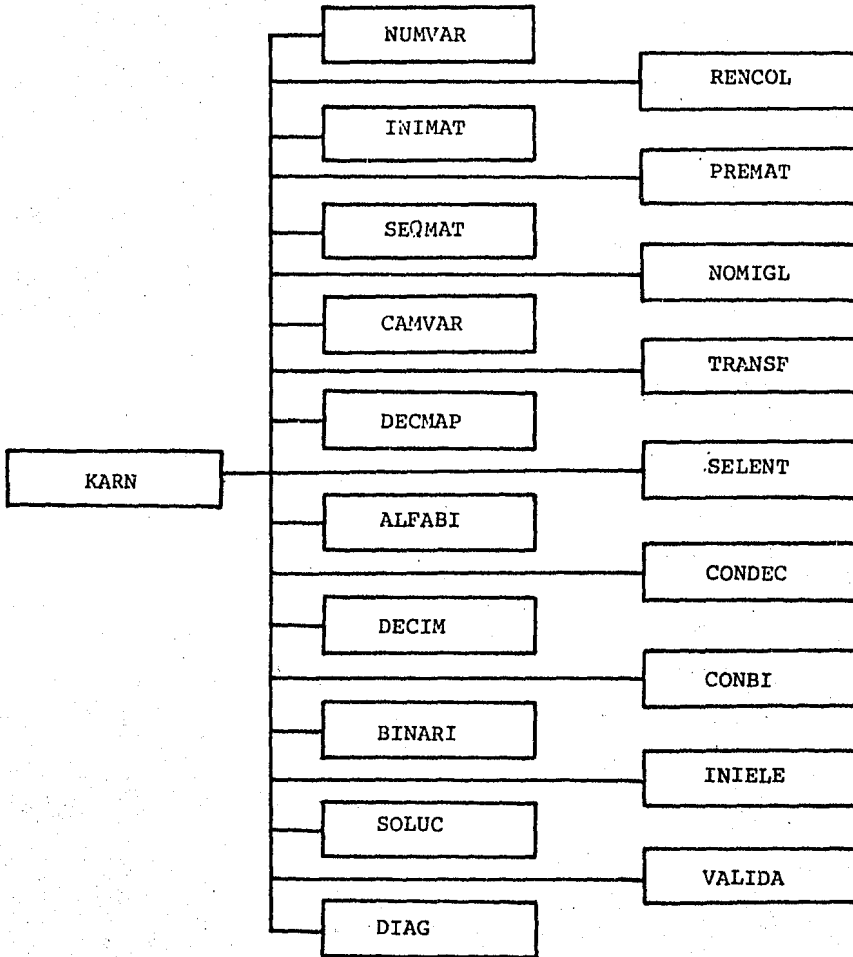


FIG. IV.1.B.1

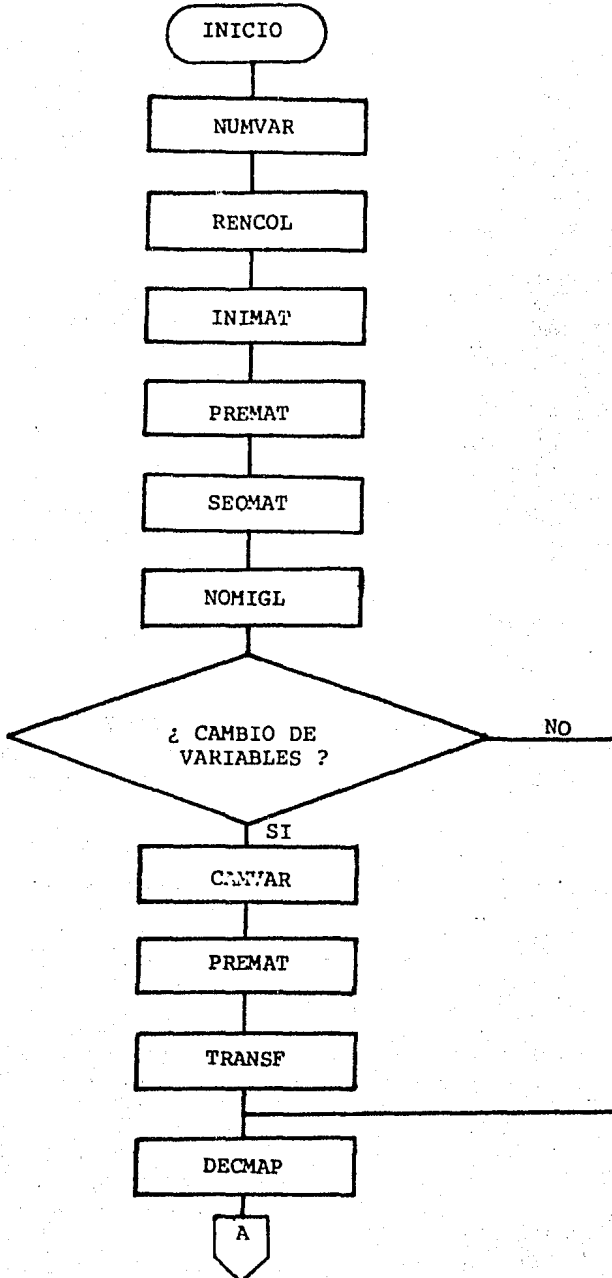
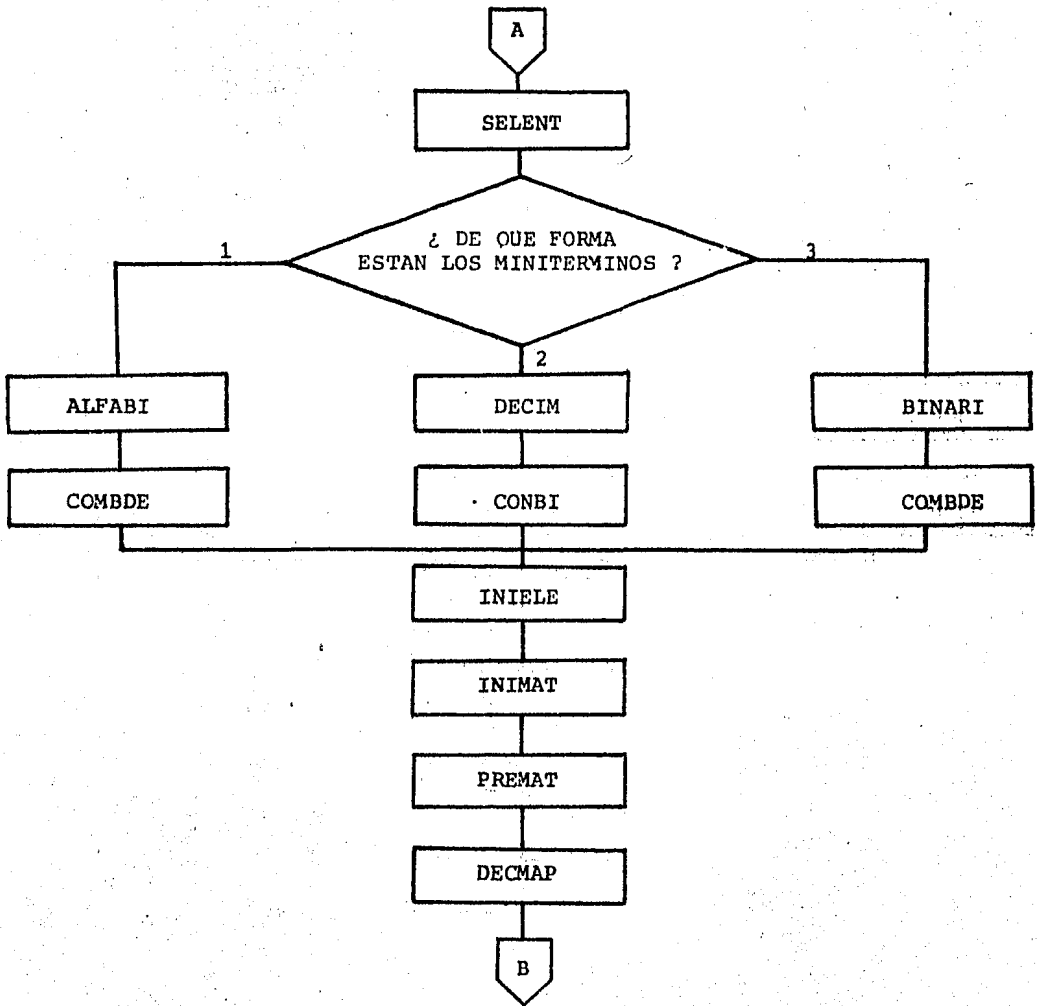
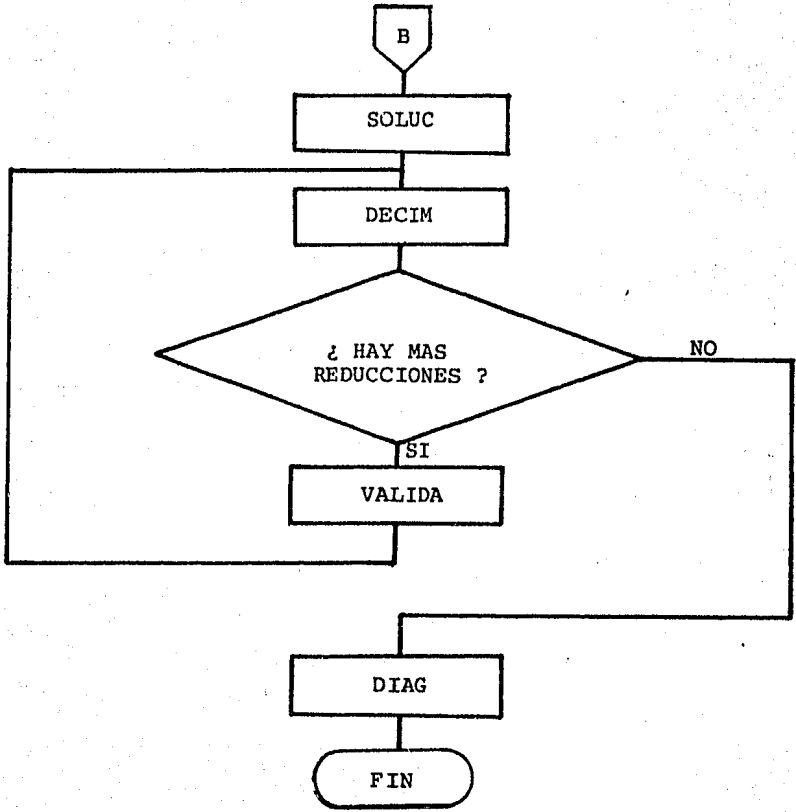


FIG. IV.1.B.2



17.R



## - INIMAT

Variables de entrada: NREN, MCOL, II  
 Variables de salida: IX, VD

Si II=0, inicializa el mapa de Karnaugh colocando un '0' en cada posición de los minterminos que puede contener.

Si II=1, es la segunda llamada a esta subrutina. Ubica '1' en las posiciones correspondientes a los minterminos de la función proporcionada por el usuario.

## - PREMAT

Variables de entrada: N, IBAN, B, IX

Esta subrutina presenta al usuario el mapa de Karnaugh de tres diferentes maneras, dependiendo del valor IBAN:

a) IBAN = 1. Presenta el mapa de Karnaugh generado por el número de variables de la función proporcionada por el usuario, utiliza las letras necesarias de 'A' a 'F' para representar estas variables.

b) IBAN = 2. Esta llamada es opcional, y solo se efectúa si se cambio en el nombre y/o secuencia de las variables de la función. Entonces presenta el mapa de Karnaugh con las variables y/o la secuencia seleccionada por el usuario.

c) IBAN = 3. Presenta el mapa de Karnaugh con '1' en la posición correspondiente a los minterminos de la función.

Mensajes para comunicarse con el usuario:

- Este es el mapa para N variables
- Este es el nuevo nombre o ubicación de las variables.
- Estos son los minterminos que proporcionaste.

## - SEQMAT

Variables de entrada: NREN, MCOL  
 Variables de salida: IMAT, IDIR

Ubica en el mapa de Karnaugh la posición de todos los posibles minterminos para el número de variables de la función, la ubicación se lleva a cabo de acuerdo a la secuencia de las variables. Además, se encarga de que los minterminos estén arreglados de tal forma que cualquier par de ellos inmediatamente adyacentes entre sí (horizontal y verticalmente), deben corresponder a un par de condiciones de entrada que sean lógicamente adyacentes, es decir, difieran solo en una variable.



## - NOMIGL

Variable de salida: IB

Pregunta al usuario si desea cambiar las variables y/o la secuencia de estas en la representación del mapa de Karnaugh.

Las letras A,B,C,D,E y F son usadas para representar las variables de la función. El usuario puede cambiar las letras para representar las variables y/o secuencia.

Mensaje que requiere la respuesta del usuario.

- Deseas cambiar el nombre o ubicación de las variables?  
(SI/NO)

## - CAMBAR

Variable de entrada: N

Variable de salida: B

Efectua el cambio de nombre de las variables y/o su secuencia.

Mensaje que requiere la respuesta del usuario:

- Proporciona el nombre de las variables que sustituyen a los representados en el mapa, de la siguiente forma:  
A=nombre-variable, B=nombre-variable,...;

Con formato libre y termina con el caracter (;)

Mensajes de error:

- Se omitio un signo igual
- Se omitio una coma
- Uno de los caracteres no es alfabetico
- Se omitio fin de texto => (;)
- El numero de cambios no es igual al numero de variables
- Una de las variables a la que se quiere cambiar el nombre, no existe
- Una de las letras esta repetida

## - TRANSF

Variables de entrada: N, B, NREN, MCOL

Variables de salida: IMAT, IDIR

El llamado a esta subrutina es opcional y solo se realiza si el usuario cambio el nombre y/o secuencia de las variables de la función.

Ubica la nueva posición de los minterminos posibles para el número de variables de la función, si la secuencia fue cambiada por el usuario.

Ejemplo para cuatro variables de secuencia ABCD

VARIABLES:	A	B	C	D
Valor asociado				
por posición:	2**3	2**2	2**1	2**0

si la secuencia se cambia a DABC

VARIABLES:	A	B	C	D
Valor asociado				
por posición:	2**3	2**2	2**1	2**0

en el mapa de Karnaugh serán representados los posibles minterminos en ambos casos, en la secuencia ABCD pero con el valor de peso asociado a el nombre de cada variable.

#### - DECMAP

Variables de entrada: VD, NREN, NCOL, IMAT, KBAN  
Variable de salida: BIFF

La subrutina DECMAP puede llamarse en dos ocasiones.

En la primera llamada, crea una estructura que contiene el mapa de Karnaugh con la ubicacion de todos los posibles minterminos para el numero de variables, en forma decimal.

En la segunda llamada, crea una estructura que contiene el mapa de Karnaugh con los minterminos de la funcion proporcionada por el usuario, marcando "1" en la posición correspondiente a cada mintermino. Junto, presenta otro mapa con todos los posibles minterminos para el numero de variables de la funcion, en su representación decimal.

Las estructuras en ambas llamadas son presentadas por la subrutina BECIM.

#### - SELENT

Variable de salida: IFORM

Muestra al usuario las opciones que tiene para proporcionar los minterminos de la funcion (diferentes notaciones) y le pide seleccione una de ellas.

Mensajes para comunicarse con el usuario:

- Descripción de las tres formas o notaciones de que dispone para proporcionar los minterminos

Mensajes que requiere la respuesta del usuario:

- Que forma de proporcionar los minterminos desea?

## - ALFABI

Variables de entrada: N, B  
Variables de salida: M, IFI

Esta subrutina es opcional y es llamada si el usuario selecciona la forma canonica para proporcionar los minterminos de la funcion:

AOBICO para indicar  $\overline{A}BC$ .

Mensajes que requieren la respuesta del usuario:

- Proporciona los minterminos separados por comas y termina con el caracter (?)

Mensajes de error:

- A una de las variables no se le esta asignando su valor binario.
- Un nombre de variable no existe o no es alfabetico.
- Nombre de variable repetido.
- El numero de variables del mintermino no es igual al numero de variables del mapa, o dos comas continuas, o no informacion y coma.

## - BINARI

Variable de entrada: N  
Variables de salida: M, IFI

El llamado a esta subrutina es opcional, y solo se efectua si se selecciona la forma binaria para proporcionar los minterminos:

O10 para indicar  $\overline{A}BC$

Mensajes que requieren la respuesta del usuario:

- Proporciona los minterminos separados por comas y termina con el caracter (?)

Mensajes de error:

- Uno de los caracteres no es dígito binario.
- Se omitio una coma.
- El numero de dígitos no es igual al numero de variables de la funcion.

## - COMBDE

Variables de entrada: N, M, IFI  
Variable de salida: ID

El llamado a esta subrutina es opcional y solo se efectua si se llamaron antes las subrutinas ALFABI o BINARI. Su objetivo es convertir los minterminos a su equivalente decimal.

**- DECIM**

Variables de entrada: N, BUFF, IBAN

Variables de salida: IDD, IFI

Esta subrutina es llamada en dos ocasiones:

a) Si IBAN = 0, requiere los minterminos en forma decimal, es decir:

Para tres variables de secuencia ABC,  $\overline{ABC} = 2$

b) Si IBAN = 1, obtiene reduccion del usuario. Se indican los minterminos que se pueden agrupar para eliminar variables. Los minterminos son indicados en su equivalente decimal. En este proceso se muestra el mapa con los minterminos proporcionados por el usuario ubicados en sus posiciones correspondientes, asociando "1" y un mapa adicional con los valores decimales correspondientes a los minterminos para el numero de variables de la funcion.

Mensajes que requieren la respuesta del usuario:

- Proporciona los minterminos en su equivalente decimal separados por comas y termina con el caracter (#).

- Proporciona tu reduccion, haciendo referencia a los minterminos por su equivalente decimal, separados por comas y termina con el caracter (#)

Mensajes de error:

- Un caracter no es numerico

- Un valor de mintermino sobrepasa maximo valor

- Mas de dos digitos para representar un mintermino

- Se omitio una coma

**- COMBI**

Variables de entrada: N, IFI, ID

Variable de salida: M

El llamado a esta subrutina es opcional y solo se efectua si se opto por la opcion en que el usuario proporciona los minterminos en forma decimal.

Convierte los minterminos de la funcion, proporcionados por el usuario en forma decimal a su equivalente binario.

## - INIELE

Variables de entrada: ID, IFI, IDIR  
Variable de salida: VD

Esta subrutina ubica en la posición correspondiente del mapa de Karnaugh, los minterminos proporcionados por el usuario.

## - VALID

Variables de entrada: IFII, IFI, IDD, ID, N, LIN, LIM

Valida la agrupación de minterminos hecha por el usuario, emitiendo un mensaje de diagnóstico al respecto.

Mensajes de diagnóstico:

- Uno de los elementos proporcionados no es mintermino
- Se sobrepasa valor máximo de minterminos
- Tu agrupación de minterminos es inválida
- Tu agrupación de minterminos es válida
- Esta agrupación ya la proporcionaste antes; ignorada ahora

## - DIAG

Variables de entrada: VEC1 N1P, NOP, TAB, II, DIR, MAT2,  
IFI, IP, NIS, N

Muestra al usuario la función que obtuvo y le indica si es óptima o no.

Si es óptima, pero existen una o varias alternativas de funciones que también son óptimas, se presentan al usuario.

Si no es óptima, además de la función obtenida por el usuario, se presenta(n) la(s) función(es) óptima(s) que exista(n).

Mensajes de diagnóstico:

- Tu función no cubre todos los minterminos.
- La función obtenida es:
- Es una función óptima
- Otras alternativas de funciones óptimas son:

## - SOLUC

Variables de entrada: N, IFI, M, ID  
Variables de salida: LIM, LIN, DIR, IVD, TAB, II, MAT2,  
NOP, NIS, MAT

La subrutina SOLUC utiliza el método de minimización de Quine Mc.Cluskey para obtener todos los cubos posibles, implicantes primos, implicantes primos esenciales, implicantes primos

secundarios y la(s) función(es) óptima(s) que exista(n) para una determinada función proporcionada por el usuario. Se explica ampliamente en el ANEXO B.

100 PRGPRM MAPAKARNAUGH(INPUT,OUTPUT);  
 200 (\* ESTE PROGRAMA RESUELVE MAPAS DE KARNAUGH A TRAVES DE JUINE MC.CLOSKEY  
 300 Y GUIA A EL USUARIO A RESOLVERLO, VALIDANDO TODAS SUS SUGERENCIAS.  
 400 EL PROGRAMA USA PAPA SU SOLUCION, PROCEDURES CUYA FUNCION SE EXPLICA  
 500 A CONTINUACION:  
 600  
 700 - TEXTO: CONTIENE UNA PEQUEZA INTRODUCCION AL METODO.  
 800 - HORIZONTAL2,HORIZONTAL4 : DEFINEN POR RENGLONES LAS POSICIONES DE  
 900 LOS MINTERMINOS EN EL MAPA K.  
 1000 - VERTICAL2,VERTICAL4 : DEFINEN POR COLUMNAS LAS POSICIONES DE LOS  
 1100 MINTERMINOS EN EL MAPA K.  
 1200 - EXPONENTE: OBTIENE LA POTENCIA DE UN NUMERO EN BASE 2.  
 1300 - RAIZ: OBTIENE LA RAIZ N(ESIMA) DE UN NUMERO.  
 1400 - GRAFICA : SE GRAFICA EN EL DISPOSITIVO DE SALIDA EL MAPA K CON  
 1500 LAS POSICIONES DE TODOS LOS POSIBLES MINTERMINOS Y EL  
 1600 Y EL MAPA CON LOS MINTERMINOS DE LA FUNCION  
 1700 - DEFHV : DEFINE EL NUMERO DE RENGLONES Y COLUMNAS DEL MAPA, ASI  
 1800 COMO EL ORDEN DE LAS VARIABLES YA QUE EN FUNCION DE EL  
 1900 SE DEFINE SU PESO  
 2000 - ACONJUNTO: SI HAY MAS DE UNA POSIBLE SOLUCION, OBTIENE TODAS  
 2100 ELLAS A TRAVES DEL METODO DE PETRIC.  
 2200 - SECUNDARIOS : DEFINE LA TABLA DE IMPLICANTES PRIMOS SECUNDARIOS  
 2300 - SOLUCION : RESUELVE A TRAVES DE Q.N. A PARTIR DE TOMAR LOS CUBOS  
 2400 CUYA DIFERENCIA SEA POTENCIA DE 2 Y OBTIENE LA  
 2500 O SOLUCIONES POSIBLES  
 2600 - BURBUJA : ORDENA LOS MINTERMINOS POR EL METODO DE BURBUJA  
 2700 - SOLUCION : PROCEDURE FUNDAMENTAL DEL PROGRAMA YA QUE ES DONDE SE  
 2800 RESUELVE EL MAPA BAJO LA SIGUIENTE SECUENCIA:  
 2900 SEPARA LOS MINTERMINOS POR EL NUMERO DE UNOS DE SU REPRESENTACION  
 3000 BINARIA ALMACENANDOLOS EN SOLUCI.ARREGLO Y EL NUMERO DE ELLOS EN  
 3100 SOLUCI.NELEM  
 3200 HACE LA COMBINACION DE LOS MINTERMINOS CUYA DIFERENCIA SEA POTEN  
 3300 CIA DE 2, MARCANDO SIMULTANEAMENTE LOS QUE SE COMBINAN Y LOS QUE  
 3400 NO, EN EL PRIMER CASO LOS MARCA CON UNA "C", EN CASO CONTRARIO,  
 3500 CON UN "\*".  
 3600 SOLUCI.NELEM ESTA ORGANIZADO DE TAL MANERA QUE LOS RENGLONES IN  
 3700 DICAN EL NUMERO DE UNOS QUE TIENEN LOS CUBOS DE LOS RENGLONES  
 3800 1-10 Y A PARTIR DEL 11 EN ADELANTE, LAS DECENAS INDICAN EL NUMERO  
 3900 DE ELEMENTOS DEL CUBO Y LAS UNIDADES EL RENGLON DEL CUBO QUE SE  
 4000 COMBINO CON EL RENGLON SIGUIENTE. EJEMPLO :  
 4100 EL RENGLON 13 INDICA:  
 4200 2\*\*1 : QUIERE DECIR QUE TIENE DOS ELEMENTOS POR CUBO  
 4300 3 : QUIERE DECIR QUE SE COMBINARON UN CUBO DEL RENGLON 3 CON  
 4400 UNO DEL RENGLON 4.  
 4500 - DECIMALES : VERIFICA LA ENTRADA DE LOS MINTERMINOS DE LA  
 4600 FUNCION EN FORMA DECIMAL.  
 4700 - CADENAS : VERIFICA LA ENTRADA DE LOS MINTERMINOS EN FORMA DE  
 4800 CADENAS BINARIAS  
 4900 - VALBIN : VERIFICA LA ENTRADA DE MINTERMINOS EN FORMA DE BINARIA  
 5000 - OPCIONES : PRESENTA LAS OPCIONES QUE TIENE EL USUARIO PARA PRO  
 5100 PORCIONAR SUS MINTERMINOS, ESTO ES BINARIA, CADENAS  
 5200 Y DECIMALES  
 5300 - MAPABINARIO: OBTIENE EL VALOR BINARIO DE TODAS LAS POSICIONES  
 5400 DE LOS MINTERMINOS DEL MAPA.  
 5500 - MAPAACTUAL : MAPA ORDENADO EN FUNCION DE LA POSICION DE LAS  
 5600 VARIABLES  
 5700 - MAPAS1 : DEFINE EL MAPA PONIENDO 1 EN LAS POSICIONES DE LOS MIN  
 5800 TERMINOS DE LA FUNCION  
 5900 - BUSCA : USADO EN LA SOLUCION DEL USUARIO PARA BUSCAR LOS CUBOS

5901  
5902  
5903  
5904  
6000  
6100  
6200  
6300  
6400  
6500  
6600  
6700  
6800  
6900  
7000  
7100  
7200  
7300  
7400  
7500  
7600  
7700  
7800  
7900  
8000  
8100  
8200  
8300  
8400  
8500  
8600  
8700  
8800  
8900  
9000  
9100  
9200  
9300  
9400  
9500  
9600  
9700  
9800  
9900  
10000  
10100  
10200  
10300  
10400  
10500  
10600  
10700  
10800  
10900  
11000  
11100  
11200  
11300  
11400  
11500  
11600  
11700  
11800  
11900  
12000  
12100

PROPUESTOS POR EL FN EL ARREGLO SOLUCI.ARREGLO

ESTAS SON TODOS LOS PROCEDUPES USADOS EN EL PROGRAMA  
EL PROGRAMA PRINCIPAL COMANDA TODA LA SOLUCION DEL ALUMNO

RESTRICCIONES:

1. DEBIDO A QUE ES UN PROGRAMA DE TIPO EDUCATIVO, ES SUFICIENTE USAR MAPAS DE HASTA 6 VARIABLES.
2. EL NUMERO DE CARACTERES EN LA PANTALLA NO PERMITE VISUALIZAR MAPAS DE MAS DE 6 VARIABLES.

\*)

LABEL 3;  
TYPE CAPAC = ARRAY[0..65] OF CHAR;  
MATRICES = ARRAY[0..70,0..65] OF INTEGER;  
MATRICESC = ARRAY[0..70,0..64] OF CHAR;  
VECTOR = ARRAY[-1..100] OF INTEGER;  
MATENT = ARRAY[0..30,0..30] OF INTEGER;  
MATCARAC = ARRAY[0..64,0..79] OF CHAR;  
NUMEROS = SET OF INTEGER;  
MAPA = ARRAY[0..3,0..8] OF INTEGER;  
CMAPA = ARRAY[0..3,0..8] OF CHAR;  
STRING = ARRAY[0..6] OF CHAR;  
LETRAS = RECORD  
LETRA : STRING; (\* VARIABLES DEL MAPA \*)  
NOASOC : VECTOR; (\* POTENCIAS DE 2 ASOCIADAS CON  
LA POSICION DE LAS VARIABLES \*)  
BINARIO : ARRAY[0..64,0..8] OF INTEGER;  
END;  
SOLUC = RECORD  
ARREGLO : MATRICES;  
ARRMARCA : MATRICESC;  
NOCUBIERTO : MATCARAC;  
MATSOL,IMPSEC,TABIMP : MATENT;  
SOLTOTAL,CUBIERTO : MATCARAC;  
MARC : ARRAY[1..30] OF CHAR;  
NELEM,NMATSOL,NSOLTOTAL,NTASSEC,NTABIMP : VECTOR;  
END;  
VAR  
I1,K,J1,NI,NJ,I,J,II,CONT,CONT1,CONT2,CONT3,CONT4,CONT5,CONT6,CONTAB,  
(\* CONTADORES \*)  
NVARIA (\* NUMERO DE VARIABLES DE LA FUNCION \*),  
NDIGITOS (\* TOTAL DE \*INTERMINOS \*),  
NIMPSEC (\* TOTAL DE IMPLICANTES SECUNDARIOS \*),  
NIMPLIC (\* TOTAL DE IMPLICANTES PRIMOS \*),  
NDIGSEC (\* NUMERO DE MINTERMINOS NO CUBIERTOS EN LA TABLA DE  
IMPLICANTES PRIMOS \*),  
NCASILLAS (\* NUMERO DE MINTERMINOS PARA SOLUCIONES PARCIALES \*),  
TOTSOL (\* TOTAL DE SOLUCIONES FINALES \*),  
POTENCIA (\* NUMERO MAXIMO DE MINTERMINOS PARA NVARIA \*),  
T1 (\* NUMERO DE 1S DE LOS MINTERMINOS \*),  
T (\* DIRECCIONAMIENTO DE RENGLONES \*),  
MAX (\* \*),  
ASECUN,  
DECIMAL (\* MINTERMINO EN FORMA DECIMAL \*)  
: INTEGER;  
MAPAS (\* POSICIONES DE TODOS LOS \*INTERMINOS POSIBLES  
PARA EL NUMERO DE VARIABLES EN EL MAPA \*),  
MAPASMIN (\* MAPA CON 1S EN LOS MINTERMINOS DE LA FUNCION \*)  
: MAPA;  
CYAFASMIN (\* MAPCA EN EL MAPA DE 1S (MAPASMIN), LOS MINTER-  
MINOS QUE VA CUBRIENDO EL USUARIO \*)  
: CMAPA;



```

12101 (* *)
12102 (* *)
12103 (* *)
12104 (* *)
12200 ARRVAR1,ARRVAR2 (* POSICION DE LAS VARIABLES POR PESO *),
CAMBIO,METODO (* LEE DECISIONES DEL USUARIO DE SI O NO *)
12400 : STRING;
12500 SOLUCIONES : MATCARAC; (* TODAS LAS SOLUCIONES DE LA
12600 MAQUINA *)
12700 TABALUM : MATENT;
12800 SIMBLO : LETRAS;
12900 SOLUCI : SOLUC;
13000 NSOLUCIONES (* NUMERO TOTAL DE SOLUCIONES *),
13100 NCONJEXTEN (* NUMERO DE POSIBLES SOLUCIONES PARA IMPL. SEC *),
13200 NCONJUNTO ,
13300 PRUEB (* CURS ENVIADOS A ORDENARSE *),
13400 DIGITOS (* MINTERMINOS *),
13500 DIGSEC (* MINTERMINOS DE LOS IMPLICANTES SECUNDARIOS *),
13600 NTABALUM (* NUMERO DE REDUCCIONES DEL ALUMNO *),
13700 CASILLAS (* CURS MANEJADOS POR EL ALUMNO *)
13800 : VECTOR;
13900 CASDIG, (* REFLEJO DE DIGITOS DONDE VERIFICA LOS MINTERMINOS
14000 QUE SE HAN CUBIERTO Y LOS QUE FALTAN POR CUBRIR *)
14100 CDIGITOS (* REFLEJO DE DIGITOS DONDE SE MARCAN LOS MINTERMINOS
14200 QUE EL USUARIO VA CUBRIENDO EN EL MAPA *) : CARAC;
14300 UNO (* MINTERMINOS CON 1 UNO EN SU REPRESENTACION BINARIA *),
14400 CONJ1,CONJ2 (* CONJUNTOS AUXILIARES PARA VERIFICACION DE INTERSEC
14500 CIONES O IGUALDAD DE CURS *)
14600 : NUMEROS;
14700 CONJUNTO (* ARREGLO DE CONJUNTOS DE CUBOS *)
14800 : ARRAY [0..50,0..150] OF NUMEROS;
14900 SOLALUM : ARRAY [0..200] OF CHAR;
15000 AUXILIAR : STRING; (* AUXILIAR EN EL MANEJO DE LAS VARIABLES
15100 DEL MAPA *)
15200 OPCION,AUX1 : CHAR;
15300 MAPCAMIL: (* MARCA LOS IMPLICANTES SECUNDARIOS QUE DAN SOLUCIONES
15400 OPTIMAS *) ARRAY [0..30] OF CHAR;
15500 BANDERA,BANDERA1,BANDERA2 : BOOLEAN; (* BANDERAS *)
15600
15700 PROCEDURE TEXTO;
15800 BEGIN
15900 WRITELN('ESTE ES UN METODO GRAFICO DE MINIMIZACION, SISTEMATICO ',
16000 'ESTANDAR. ');
16100 WRITELN('ESTE PROGRAMA CONSIDERA LA FORMA DE SUMA DE PRODUCTO PARA ',
16200 'EL MANEJO DE ');
16300 WRITELN('TUS MINTERMINOS. ');
16400 WRITELN('EL PRIMER MAPA TE SERA PRESENTADO CON LAS VARIABLES QUE ',
16500 'EL SISTEMA TE ');
16600 WRITELN('ASIGNA POR OMISION, PUEDES CAMBIARLAS, SI ASI LO DESEAS. ');
16700 WRITELN('LA VARIABLE DE MAYOR PESO ES LA QUE SE PRESENTA MAS A LA ',
16800 'IZQUIERDA, LA ');
16900 WRITELN('DE MENOR PESO, LA QUE SE ENCUENTRA MAS A LA DERECHA. ');
17000 WRITELN('EL OBJETO DE RENOMBRAR LAS VARIABLES PUEDE SER: ');
17100 WRITELN;
17200 WRITELN('1: USAR NOMBRES QUE MAS FRECUENTEMENTE USES ');
17300 WRITELN('2: CAMBIAR EL PESO DE ELLAS. ');
17400 WRITELN('EJEMPLO: ');
17500 WRITELN(' PARA 3 VARIABLES, UN MINTERMINO PUEDE SER: ');
17600 WRITELN(' C1B1AC ');
17700 WRITELN(' DONDE ');
17800 WRITELN(' C,B,A, : NOMBRES DE LAS VARIABLES SIENDO EL PESO A<B<C ');
17900 WRITELN(' 1,1,0 : VALORES BINARIOS ASOCIADOS A LA VARIABLE QUE ',
18000 'LE ANTECEDE, SU ');
18100 WRITELN(' EQUIVALENTE DECIMAL ES: 6 ');
18200 WRITELN('SI CAMBIAMOS DE POSICION LAS VARIABLES CON SU VALOR BINA
18300 'RIO ASOCIADO ');

```

```

18301 (* *)
18302 (* *)
18303 (* *)
18304 (* *)
18400 WRITELN('POR EJEMPLO: 400101');
18500 WRITELN('SU EQUIVALENTE DECIMAL ES: 011 = 3');
18600 WRITELN('CONTINUA...'); READLN;
18700 WRITELN('ASI, UN MAPA QUE POR OMISION PRESENTA LA MAQUINA ES:');
18800 WRITELN(' C?B?');
18900 WRITELN('-----');
19000 WRITELN(' ? 0 ? 1 ? 3 ? 2 ? ');
19100 WRITELN('-----');
19200 WRITELN(' ? 4 ? 5 ? 7 ? 6 ? ');
19300 WRITELN('-----');
19400 WRITELN('UNA FORMA DE RENOMBRAR LAS VARIABLES SERIA:');
19500 WRITELN(' A=B');
19600 WRITELN(' B=C');
19700 WRITELN(' C=A');
19800 WRITELN('QUE GENERARIA EL MAPA:');
19900 WRITELN(' A?C?');
20000 WRITELN('-----');
20100 WRITELN(' ? 0 ? 2 ? 6 ? 4 ? ');
20200 WRITELN('-----');
20300 WRITELN(' ? 1 ? 3 ? 7 ? 5 ? ');
20400 WRITELN('-----');
20500 WRITELN;
20600 WRITELN('COMO VERAS, DONDE ESTABA EL MINTERMINO 3 AHORA ESTA EL 6. ');
20700 WRITELN('CONTINUA...'); READLN;
20800 WRITELN('PUEDES USAR MAPAS DESDE 2 HASTA 6 VARIABLES. ');
20900 WRITELN('LA TEGRIA DE LA MINIMIZACION A TRAVES DE MAPAS DE KARNAUGH,');
21000 ' TE DICE QUE ');
21100 WRITELN('DEBE HABER UN CUADRADO QUE CORRESPONDA A CADA COMBINACION,');
21200 ' DE ENTRADA. ');
21300 WRITELN('OTRO REQUISITO CONSISTE EN QUE LOS CUADRADOS DEBEN ESTAR,');
21400 ' ARREGLADOS EN ');
21500 WRITELN('TAL FORMA QUE, CUALQUIER PAR DE ELLOS INMEDIATAMENTE,');
21600 ' ADYACENTES ENTRE ');
21700 WRITELN('SI (HORIZONTAL O VERTICALMENTE) DEBEN CORRESPONDER,');
21800 ' A UN PAR DE CONDICIO ');
21900 WRITELN('NES DE ENTRADA QUE SEAN LOGICAMENTE ADYACENTES, ES,');
22000 ' DECIR, QUE DIFIERAN ');
22100 WRITELN('EN 1 VARIABLE. POR EJEMPLO: ');
22200 WRITELN('LOS CUADRADOS 2 Y 3 DEL MAPA ANTERIOR, CORRESPONDEN A LAS,');
22300 ' ENTRADAS ');
22400 WRITELN(' 00110 ');
22500 WRITELN(' 00111 ');
22600 WRITELN('QUE SON IDENTICAS EXCEPTO EN A. ');
22700 WRITELN('OBSERVA QUE LOS CUADRADOS DE LOS EXTREMOS DE LAS COLUMNAS,');
22800 ' O LAS HILERAS ');
22900 WRITELN('SON LOGICAMENTE ADYACENTES. ');
23000 WRITELN('DE LO ANTERIOR: ');
23100 WRITELN(' CUALQUIER PAR DE MINTERMINOS DE N VARIABLES QUE SEAN,');
23200 ' ADYACENTES EN EL ');
23300 WRITELN('MAPA DE KARNAUGH, SE PUEDEN COMBINAR EN UN SOLO MINTER,');
23400 ' COMO PRODUCTO DE ');
23500 WRITELN('N-1 LITERALES, PUESTO QUE LOS CONJUNTOS DE DOS MINTER,');
23600 ' MINOS SE COMBINAN ');
23700 WRITELN('MINAN PARA ELIMINAR 1 VARIABLE, LOS CONJUNTOS DE 4 SE COMB,');
23800 ' MINAN PARA ');
23900 WRITELN('ELIMINAR 2; LOS CONJUNTOS DE 8, PARA ELIMINAR 3, ETC. ');
24000 WRITE('AHORA BIEN ');
24100 END; (* TEXT *)
24200
24300 PROCEDURE HORIZONTAL2(VAR MAPEO:MAPA);
24400 VAR AUXILIAR : ARRAY[0..8] OF INTEGER;
24500 BEGIN

```

```

24501 (* *)
245L2 (* *)
245CZ (* *)
245C4 (* *)
246C0
247C0 FOR I:=0 TO (NI-1) DO
248C0 BEGIN
249C0 AUXILIAR[I]:=MAPEO[I,2];
250C0 MAPEO[I,2]:=MAPEO[I,3];
251C0 MAPEO[I,3]:=AUXILIAR[I];
252C0 END;
253C0
254C0 PROCEDURE HORIZONTAL4(VAR MAPEO:MAPA);
255C0 VAR AUXILIAR : ARRAY[0..8] OF INTEGER;
256C0 BEGIN
257C0 FOR I:=0 TO NI-1 DO
258C0 BEGIN
259C0 AUXILIAR[I]:=MAPEO[I,6];
260C0 MAPEO[I,6]:=MAPEO[I,7];
261C0 MAPEO[I,7]:=AUXILIAR[I];
262C0 END;
263C0 END;
264C0
265C0 PROCEDURE VERTICAL2(VAR MAPEO:MAPA);
266C0 VAR AUXILIAR : ARRAY[0..8] OF INTEGER;
267C0 BEGIN
268C0 FOR J:=0 TO NJ-1 DO
269C0 BEGIN
270C0 AUXILIAR[J]:=MAPEO[2,J];
271C0 MAPEO[2,J]:=MAPEO[3,J];
272C0 MAPEO[3,J]:=AUXILIAR[J]
273C0 END
274C0 END;
275C0
276C0 PROCEDURE VERTICAL4(VAR MAPEO:MAPA);
277C0 VAR AUXILIAR : ARRAY[0..8] OF INTEGER;
278C0 BEGIN
279C0 FOR J:=0 TO NJ-1 DO
280C0 BEGIN
281C0 AUXILIAR[J]:=MAPEO[6,J];
282C0 MAPEO[6,J]:=MAPEO[7,J];
283C0 MAPEO[7,J]:=AUXILIAR[J];
284C0 END;
285C0 END;
286C0
287C0 FUNCTION EXPONENTE(N:INTEGER) : INTEGER;
288C0 BEGIN
289C0 EXPONENTE:=ROUND(EXP(N* LN(2)));
290C0 END;
291C0
292C0 FUNCTION RAIZ(NUMERO:INTEGER) : INTEGER;
293C0 VAR I,COCIENTE : INTEGER;
294C0 BEGIN
295C0 COCIENTE:=0;
296C0 I:=NUMERO;
297C0 WHILE I > 1 DO
298C0 BEGIN
299C0 I:=I DIV 2;
300C0 COCIENTE:=COCIENTE+1;
301C0 END;
302C0 RAIZ:=COCIENTE;
303C0 END; (* RAIZ *)
304C0
305C0 PROCEDURE GRAFICA(MAPEO,MAPEOMIN:MAPA; CMAPEOMIN:CMAPA);
306C0 BEGIN
307C0 IF BANDEPA THEN

```

```

30701 (* *)
30702 (* *)
30703 (* *)
30704 (* *)
30800 I1:=2
30900 ELSE
31000 I1:=1;
31100 FOR J:=1 TO I1 DO
31200 BEGIN
31300 FOR I:=2 DOWNT0 2 DO
31400 WRITE(ARRVAR2[I]);
31500 WRITE(' ');
31600 FOR I:=2 DOWNT0 2 DO
31700 WRITE(ARRVAR1[I]);
31800 FOR I:=1 TO NJ DO
31900 WRITE(' ');
32000 END;
32100 WRITELN;
32200 WRITE(' -');
32300 FOR J:=1 TO NJ DO
32400 WRITE('-----');
32500 IF BANDERA THEN
32600 BEGIN
32700 WRITE(' -');
32800 FOR J:=1 TO NJ DO
32900 WRITE('-----');
33000 END;
33100 WRITELN;
33200 FOR I:=0 TO NI-1 DO
33300 BEGIN
33400 WRITE(' ?');
33500 FOR J:=0 TO NJ-1 DO
33600 WRITE(MAPEO[I,J]:2,' ?');
33700 IF BANDERA THEN
33800 BEGIN
33900 WRITE(' ?');
34000 IF BANDEPA2 THEN
34100 FOR J:=0 TO NJ-1 DO
34200 WRITE(MAPEOMIN[I,J]:2,MAPEOMIN[I,J]:1,'?')
34300 ELSE
34400 FOR J:=0 TO NJ-1 DO
34500 WRITE(MAPEOMIN[I,J]:2,' ?');
34600 END;
34700 WRITELN;
34800 WRITE(' -');
34900 FOR J:=1 TO NJ DO
35000 WRITE('-----');
35100 IF BANDERA THEN
35200 BEGIN
35300 WRITE(' -');
35400 FOR J:=1 TO NJ DO
35500 WRITE('-----');
35600 END;
35700 WRITELN;
35800 END;
35900 WRITELN('CONTINUA...'); READLN;
36000 END; (* GRAFICA *)
36100
36200 PROCEDURE DEFHV(VVARIA:INTEGER;VAR NJ,NI:INTEGER);
36300 BEGIN
36400 FOR J:=0 TO 2 DO
36500 BEGIN
36600 ARRVAR1[I]:= ' ';
36700 ARRVAR2[I]:= ' ';
36800 END;
36900 CASE VVARIA OF

```

```

36901 (* *)
36902 (* *)
36903 (* *)
36904 (* *)
37000
37100 1: BEGIN
37200     WRITELN('PARA UNA SOLA VARIABLE, LA SOLUCION ES: ');
37300     WRITE(' SI LA VARIABLE TOMA EL VALOR DE "1", SE ');
37400     WRITE('TIENE UNA TAUTOLOGIA');
37500     WRITELN;
37600     WRITE(' SI LA VARIABLE TOMA EL VALOR DE "0", SE ');
37700     WRITE('TIENE UNA ANTITAUTOLOGIA ');
37800     WRITELN;
37900     ABORT;
38000     END;
38100 2: BEGIN
38200     NI:=2;
38300     NJ:=2;
38400     ARRVAR1[0]:=SIMBOLO.LETRA[1];
38500     ARRVAR2[0]:=SIMBOLO.LETRA[2];
38600     END;
38700 3: BEGIN
38800     NI:=2;
38900     NJ:=4;
39000     ARRVAR1[0]:=SIMBOLO.LETRA[1];
39100     ARRVAR1[1]:=SIMBOLO.LETRA[2];
39200     ARRVAR2[0]:=SIMBOLO.LETRA[3];
39300     END;
39400 4: BEGIN
39500     NI:=4;
39600     NJ:=4;
39700     ARRVAR1[0]:=SIMBOLO.LETRA[1];
39800     ARRVAR1[1]:=SIMBOLO.LETRA[2];
39900     ARRVAR2[0]:=SIMBOLO.LETRA[3];
40000     ARRVAR2[1]:=SIMBOLO.LETRA[4];
40100     END;
40200 5: BEGIN
40300     NI:=4;
40400     NJ:=8;
40500     ARRVAR1[0]:=SIMBOLO.LETRA[1];
40600     ARRVAR1[1]:=SIMBOLO.LETRA[2];
40700     ARRVAR1[2]:=SIMBOLO.LETRA[3];
40800     ARRVAR2[0]:=SIMBOLO.LETRA[4];
40900     ARRVAR2[1]:=SIMBOLO.LETRA[5];
41000     END;
41100 6: BEGIN
41200     NI:=8;
41300     NJ:=8;
41400     ARRVAR1[0]:=SIMBOLO.LETRA[1];
41500     ARRVAR1[1]:=SIMBOLO.LETRA[2];
41600     ARRVAR1[2]:=SIMBOLO.LETRA[3];
41700     ARRVAR2[0]:=SIMBOLO.LETRA[4];
41800     ARRVAR2[1]:=SIMBOLO.LETRA[5];
41900     ARRVAR2[2]:=SIMBOLO.LETRA[6];
42000     END;
42100 END; (* DEFHV *)
42200
42300 PROCEDURE ACONJUNTO (NDIGS, NIMP: INTEGER; NN: VECTOR; CUBRE: MATCARAC;
42400 VAR CONT: INTEGER);
42500 CONST ELIMINA=1000;
42600 VAR
42700     RADEPA: BOOLEAN;
42800     TOPE: VECTOR;
42900     J1, I, J, K, JJ, K1, K2, POSICION, RELATIVO, MINIMO : INTEGER;
43000 BEGIN
43100     K:=-1;

```

```

43101 (* *)
43102 (* *)
43103 (* *)
43104 (* *)
43200 FOR I:=0 TO NDIGS DO
43300 BEGIN
43400 JJ:=0;
43500 FOR J:=1 TO NIMP DO
43600 IF CUBREC[J,I]='A' THEN
43700 BEGIN
43800 CONJUNTO[I,JJ]:=[J];
43900 JJ:=JJ+1
44000 END;
44100 NCONJUNTO[I]:=JJ;
44200 END;
44300 FOR I:=0 TO NDIGS DO
44400 NCONJEXTEN[I]:=NCONJUNTO[I];
44500 (* PREPARA LOS CONJUNTOS PARA UNIRSE *)
44600 FOR I:=0 TO (NDIGS-1) DO
44700 FOR J:=(I+1) TO NDIGS DO
44800 BEGIN
44900 JJ:=0;
45000 FOR K:=NCONJEXTEN[J] TO (NCONJEXTEN[J]+NCONJEXTEN[I]-1) DO
45100 BEGIN
45200 CONJUNTO[J,K]:=CONJUNTO[J,JJ];
45300 JJ:=JJ+1;
45400 END;
45500 NCONJEXTEN[J]:=K;
45600 CONJUNTO[J,K]:=[ELIMINA];
45700 END;
45800 CONJUNTO[J,NCONJUNTO[I]]:=[ELIMINA];
45900 FOR I:=0 TO NDIGS-1 DO
46000 BEGIN
46100 J:=0;
46200 WHILE CONJUNTO[I,J] <> [ELIMINA] DO
46300 BEGIN
46400 FOR K:=J+NCONJUNTO[I+1] TO (NCONJUNTO[I+1]*(1+J) - 1) DO
46500 CONJUNTO[I+1,K]:=CONJUNTO[I+1,K]+CONJUNTO[I,J];
46600 J:=J+1;
46700 END;
46800 END;
46900 JJ:=0;
47000 FOR I:=0 TO (NCONJEXTEN[NDIGS]) DO
47100 BEGIN
47200 BANDERA:=FALSE;
47300 K1:=I;
47400 IF CONJUNTO[NDIGS,I] <> [ELIMINA] THEN
47500 BEGIN
47600 FOR J:=0 TO (NCONJEXTEN[NDIGS]-1) DO
47700 IF (J <> I) AND (CONJUNTO[NDIGS,J] <> [ELIMINA]) AND
47800 (CONJUNTO[NDIGS,J]+CONJUNTO[NDIGS,I] = CONJUNTO[NDIGS,I]) THEN
47900 BEGIN
48000 BANDERA:=TRUE;
48100 CONJUNTO[NDIGS,J]:=[ELIMINA];
48200 END;
48300 FOR J:=0 TO NCONJEXTEN[NDIGS]-1 DO
48400 IF (J <> I) AND (CONJUNTO[NDIGS,J] <> [ELIMINA]) AND
48500 (CONJUNTO[NDIGS,J]+CONJUNTO[NDIGS,I] = CONJUNTO[NDIGS,I]) THEN
48600 BEGIN
48700 BANDERA:=TRUE;
48800 CONJUNTO[NDIGS,I]:=[ELIMINA];
48900 K1:=J;
49000 FOR J1:=0 TO NCONJEXTEN[NDIGS]-1 DO
49100 IF (J1 <> K1) AND (CONJUNTO[NDIGS,J1] <> [ELIMINA]) AND
49200 (CONJUNTO[NDIGS,K1]+CONJUNTO[NDIGS,J1]=CONJUNTO[NDIGS,K1]) THEN
49300 CONJUNTO[NDIGS,J1]:=[ELIMINA];

```

```

49301      (* *)
49302      (* *)
49303      (* *)
49304      (* *)
49400      END; BANDERA:=TRUE; WHILE BANDERA DO BEGIN BANDERA:=FALSE;
49500      FOR J:=0 TO (JJ-1) DO
49600      IF CONJUNTO[ENDIGS+1,J]+CONJUNTO[ENDIGS,K1] = CONJUNTO[ENDIGS,
49700      THEN BEGIN
49800          FOR J1:=J TO JJ-2 DO
49900              CONJUNTO[ENDIGS+1,J1]:=CONJUNTO[ENDIGS+1,J1+1];
50000          JJ:=JJ-1; BANDERA:=TRUE; J:=JJ;
50100      END;
50200      END;
50300      CONJUNTO[ENDIGS+1,JJ]:=CONJUNTO[ENDIGS,K1];
50400      JJ:=JJ+1;
50500      IF K1 <> I THEN
50600          CONJUNTO[ENDIGS,K1]:=ELIMINA;
50700      END;
50800      END;
50900      K2:=JJ-1;
51000      FOR I:=0 TO K2 DO
51100      BEGIN
51200          J:=1;
51300          K1:=0;
51400          TOPE[I]:=0;
51500          CONJ2:=CONJUNTO[ENDIGS+1,I];
51600          WHILE J<=NIMP DO
51700          BEGIN
51800              IF (J IN CONJ2) THEN
51900              BEGIN
52000                  TOPE[I]:=TOPE[I]+SOLUCI.NTABSECC[J];
52100                  SOLUCI.MATSOL[I,K1]:=J;
52200                  K1:=K1+1;
52300              END;
52400              J:=J+1;
52500          END;
52600          SOLUCI.NMATSOLEI[I]:=K1;
52700      END;
52800      (* SE BUSCA EL CONJUNTO CON COSTO MINIMO *)
52900      FOR I:=0 TO K2 DO
53000          MARCAMIN[I]:=' ';
53100          MINIMO:=TOPE[I];
53200          POSICION:=0;
53300          FOR I:=1 TO K2 DO
53400              IF TOPE[I]<MINIMO THEN
53500              BEGIN
53600                  MINIMO:=TOPE[I];
53700                  POSICION:=I;
53800              END;
53900          MARCAMIN[POSICION]:='*';
54000      (* BUSCA TODOS LOS CONJUNTOS CON EL MISMO COSTO MINIMO *)
54100      FOR I:=0 TO K2 DO
54200          IF I<>POSICION THEN
54300          IF TOPE[I]=MINIMO THEN
54400              MARCAMIN[I]:='*';
54500              CONT:=K2;
54600      END ; (* ACONJUNTO *)
54700
54800      PROCEDURE SECUNDARIOS(J:INTEGER;VAR II:INTEGER);
54900      VAR
55000          JJ,CONT,K : INTEGER;
55100      BEGIN
55200      (* SE OBTIENE LA TABLA DE IMPLICANTES SECUNDARIOS *)
55300      FOR K:=0 TO (SOLUCI.NTABIMP[J]-1) DO
55400          SOLUCI.IMPSEC[II,K]:=SOLUCI.TABIMP[J,K];
55500      FOR I:=0 TO ADIGITOS DO

```

```

55501 (* *)
55502 (* *)
55503 (* *)
55504 (* *)
55600 IF (SOLUCI.CUBIERTO[I,J]='&') AND (SOLUCI.CUBIERTO[NIMPLIC+1,I]
55700 = ' ') THEN
55800 BEGIN
55900 JJ:=0;
56000 WHILE DIGITOS[I]<>DIGSFCE[JJ] DO
56100 JJ:=JJ+1;
56200 SOLUCI.NOCUBIERTO[I,JJ]='&';
56300 SOLUCI.NTABSECC[I]:=SOLUCI.NTABIMP[JJ];
56400 END;
56500 II:=II+1;
56600 END ; (* SECUNDARIOS *)
56700
56800 PROCEDURE SOLUMAQ(VAR CONT:INTEGER;NELEMEN,I:INTEGER;MATRIZ:MATENT);
56900 VAR
57000 BINARYO:ARRAY[0..30,0..30] OF CHAR;
57100 LIM2,K,CONT1,CONT2,CONT3,DIR,DIR1,DECIMAL,DELTA,LIM:INTEGER;
57200 AUX: CHAR;
57300 BEGIN
57400 DIR:=0;
57500 FOR CONT1:=0 TO (NELEMEN-1) DO
57600 BEGIN
57700 DECIMAL:=MATRIZ[I,CONT1];
57800 FOR CONT2:=0 TO (NVARIA-1) DO
57900 BEGIN
58000 IF (DECIMAL MOD 2) = 1 THEN
58100 BINARYO[DIR,CONT2]='1'
58200 ELSE
58300 BINARYO[DIR,CONT2]='0';
58400 DECIMAL:=DECIMAL DIV 2;
58500 END;
58600 DIR:=DIR+1;
58700 END;
58800 DELTA:=2;
58900 LIM2:=NELEMEN DIV 2;
59000 LIM:=LIM2;
59100 IF NELEMEN > 1 THEN
59200 FOR CONT3:=1 TO LIM2 DO
59300 BEGIN
59400 WHILE NOT(CONT3 IN UNO) DO
59500 CONT3:=CONT3+1;
59600 FOR CONT1:=0 TO (LIM-1) DO
59700 BEGIN
59800 DIR1:=CONT1*DELTA;
59900 DIR:=DIR1+CONT3;
60000 FOR CONT2:=0 TO (NVARIA-1) DO
60100 IF (BINARYO[DIR,CONT2]) <> (BINARYO[DIR1,CONT2]) THEN
60200 BINARYO[DIR1,CONT2]='X';
60300 END;
60400 LIM:=LIM DIV 2;
60500 DELTA:=DELTA*2;
60600 END;
60700 FOR CONT1:=1 TO NVARIA DO
60800 AUXILIAR[CONT1]:=SIMBOLO.LETRA[CONT1];
60900 FOR CONT1:=1 TO (NVARIA-1) DO
61000 FOR CONT2:=(CONT1+1) TO NVARIA DO
61100 IF AUXILIAR[CONT2] < AUXILIAR[CONT1] THEN
61200 BEGIN
61300 AUX:=AUXILIAR[CONT1];
61400 AUXILIAR[CONT1]:=AUXILIAR[CONT2];
61500 AUXILIAR[CONT2]:=AUX;
61600 END;
61700 CONT1:=0;

```



```

61701 (* *)
61702 (* *)
61703 (* *)
61704 (* *)
61800 FOR K:=(NVARIA-1) DOWNTO 0 DO
61900 IF BINARYOCC,K>>'X' THEN
62000 BEGIN
62100 SOLUCI.SOLTOTAL[CONT,CONT1]:=AUXILIAR[K+1];
62200 SOLUCI.SOLTOTAL[CONT,CONT1+1]:=BINARYOCC,K;
62300 CONT1:=CONT1+2;
62400 SOLUCI.SOLTOTAL[CONT]:=CONT1;
62500 END;
62600 CONT:=CONT+1;
62700 END; (* SOLUMAD *)
62800
62900 PROCEDURE BURBUJA(VAR DIGITOS:VECTOR;NDIGITOS:INTEGER);
63000 VAR
63100 CONT1,AUX,
63200 CONT2 : INTEGER;
63300 BEGIN
63400 FOR CONT2:=0 TO (NDIGITOS-1) DO
63500 FOR CONT1:=(CONT2+1) TO NDIGITOS DO
63600 IF DIGITOSE[CONT1] < DIGITOSE[CONT2] THEN
63700 BEGIN
63800 AUX:=DIGITOSE[CONT2];
63900 DIGITOSE[CONT2]:=DIGITOSE[CONT1];
64000 DIGITOSE[CONT1]:=AUX;
64100 END;
64200 END; (* DE BURBUJA *)
64300
64400 PROCEDURE SOLUCION;
64500 VAR
64600 MODUL,AUX,MED,DIFERENCIA,J1,K2,L,T,I,J,K,I1 : INTEGER;
64700 CONTCURR,ESTADO,I3,I2,K1,CONT,CONT1,DIR,
64800 T1:INTEGER;
64900 BEGIN
65000 BANDERA:=TRUE;
65100 FOR I:=0 TO 64 DO
65200 BEGIN
65300 FOR J:=0 TO 50 DO
65400 BEGIN
65500 SOLUCI.ARREGLOC[J]:=0;
65600 SOLUCI.AR2MARCA[I,J]:='0';
65700 END;
65800 SOLUCI.NELEM[I]:=0;
65900 SOLUCI.NSOLTOTAL[I]:=0;
66000 END;
66100 FOR I:=1 TO 30 DO
66200 BEGIN
66300 FOR J:=0 TO 50 DO
66400 BEGIN
66500 SOLUCI.VOCUBIERTO[I,J]:=' ';
66600 SOLUCI.CUBIERTO[I,J]:=' ';
66700 END;
66800 SOLUCI.MARCA[I]:=' ';
66900 END;
67000 FOR I:=0 TO NDIGITOS DO
67100 CASE DIGITOSE[I] OF
67200 0: SOLUCI.NELEM[0]:=1;
67300 1,2,4,8,16,32,64: BEGIN
67400 SOLUCI.ARREGLOC[1,SOLUCI.NELEM[1]]:=DIGITOSE[I];
67500 SOLUCI.NELEM[1]:=SOLUCI.NELEM[1]+1;
67600 END;
67700 3,5,6,9,10,12,17,18,20,24,33,34,36,40,49: BEGIN
67800 SOLUCI.ARREGLOC[2,SOLUCI.NELEM[2]]:=DIGITOSE[I];
67900 SOLUCI.NELEM[2]:=SOLUCI.NELEM[2]+1;

```

```

67901 (* *)
67902 (* *)
67903 (* *)
67904 (* *)
68000      END;
68100      7,11,13,14,19,21,22,25,26,28,35,37,38,41,42,44,49,50,52,56: BEGIN
68200          SOLUCI.ARREGLOC3,SOLUCI.NELEM[3]:=DIGITOS[1];
68300          SOLUCI.NELEM[3]:=SOLUCI.NELEM[3]+1;
68400      END;
68500      15,23,27,29,30,39,43,45,46,51,53,54,57,59,60: BEGIN
68600          SOLUCI.ARREGLOC4,SOLUCI.NELEM[4]:=DIGITOS[1];
68700          SOLUCI.NELEM[4]:=SOLUCI.NELEM[4]+1;
68800      END;
68900      31,47,55,59,61,62: BEGIN
69000          SOLUCI.ARREGLOC5,SOLUCI.NELEM[5]:=DIGITOS[1];
69100          SOLUCI.NELEM[5]:=SOLUCI.NELEM[5]+1;
69200      END;
69300      63: BEGIN
69400          SOLUCI.ARREGLOC6,SOLUCI.NELEM[6]:=DIGITOS[1];
69500          SOLUCI.NELEM[6]:=SOLUCI.NELEM[6]+1;
69600      END;
69700      END; (+ CASE +)
69800      IF SOLUCI.NELEM[1]=1 THEN
69900          IF SOLUCI.NELEM[1]>0 THEN
70000              BEGIN
70100                  SOLUCI.ARRMARCA[0,0]='C';
70200                  FOR I:=0 TO (SOLUCI.NELEM[1]+2-1) DO
70300                      BEGIN
70400                          SOLUCI.ARREGLOC[1,I]:=0;
70500                          I:=I+1;
70600                          MED:=I DIV 2;
70700                          SOLUCI.ARREGLOC[1,I]:=SOLUCI.ARREGLOC[1,MED];
70800                          SOLUCI.ARRMARCA[1,MED]='C';
70900                      END;
71000                      SOLUCI.NELEM[1]:=SOLUCI.NELEM[1]+2;
71100              END
71200              ELSE
71300                  SOLUCI.ARRMARCA[0,0]='*';
71400                  K:=11;
71500                  FOR I:=1 TO (NVARIA-1) DO
71600                      BEGIN
71700                          K:=K+1;
71800                          L:=0;
71900                          IF ((SOLUCI.NELEM[I]>0) AND (SOLUCI.NELEM[I+1]>0)) THEN
72000                              BEGIN
72100                                  FOR I:=L TO (SOLUCI.NELEM[I]-1) DO
72200                                      FOR J:=0 TO (SOLUCI.NELEM[I+1]-1) DO
72300                                          IF ((SOLUCI.ARREGLOC[I+1,J] - SOLUCI.ARREGLOC[I,I]) IN UNO)
72400                                              AND (SOLUCI.ARREGLOC[I+1,J]-SOLUCI.ARREGLOC[I,I] > 0) THEN
72500                                                  BEGIN
72600                                                      SOLUCI.ARREGLOC[K,L]:=SOLUCI.ARREGLOC[I,I];
72700                                                      SOLUCI.ARREGLOC[K,L+1]:=SOLUCI.ARREGLOC[I+1,J];
72800                                                      L:=L+2;
72900                                                      SOLUCI.ARRMARCA[I,I]='C';
73000                                                      SOLUCI.ARRMARCA[I+1,J]='C';
73100                                                  END
73200                                                  ELSE
73300                                                      IF SOLUCI.ARRMARCA[I,I] <> 'C' THEN
73400                                                          SOLUCI.ARRMARCA[I,I]='*';
73500                                                  END
73600                                                  ELSE
73700                                                      FOR I:=0 TO SOLUCI.NELEM[I]-1 DO
73800                                                          IF SOLUCI.ARRMARCA[I,I] <> 'C' THEN
73900                                                              SOLUCI.ARRMARCA[I,I]='*';
74000                                                              SOLUCI.NELEM[K]:=L;
74100              END;

```

```

74101 (* *)
74102 (* *)
74103 (* *)
74104 (* *)
74200 IF SOLUCI.NELEM[NVARIA] > 0 THEN
74300 FOR I:=0 TO SOLUCI.NELEM[NVARIA]-1 DO
74400 IF SOLUCI.ARRMARCA[NVARIA,I] <> 'C' THEN
74500 SOLUCI.ARRMARCA[NVARIA,I]:='*';
74600 K:=21;
74700 T:=11;
74800 K2:=1;
74900 T1:=EXPONENTE(K DIV 10);
75000 WHILE (SOLUCI.NELEM[T]=0) AND (T<70) DO
75100 BEGIN
75200 T:=T+1;
75300 K:=K+1;
75400 END;
75500 WHILE SOLUCI.ARREGLOCT,I] <> 0 DO
75600 BEGIN
75700 WHILE (SOLUCI.ARREGLOCT+1,I] <> 0) OR (SOLUCI.ARREGLOCT,I] <> 0)
75800 IF SOLUCI.ARREGLOCT+1,I] <> 0 THEN
75900 BEGIN
76000 K1:=0;
76100 BANDERA:=TRUE;
76200 T1:=EXPONENTE(K DIV 10);
76300 FOR I:=0 TO (SOLUCI.NELEM[T]-1) DO
76400 BEGIN
76500 FOR J:=0 TO (SOLUCI.NELEM[T+1]-1) DO
76600 BEGIN
76700 DIFERENCIA:=SOLUCI.ARREGLOCT+1,J]-SOLUCI.ARREGLOCT,I];
76800 FOR J1:=1 TO K2 DO
76900 BEGIN
77000 J:=J+1;
77100 I:=I+1;
77200 AUX:=SOLUCI.ARREGLOCT+1,J]-SOLUCI.ARREGLOCT,I];
77300 IF (AUX=DIFERENCIA) AND (AUX>0) AND (AUX IN UNO) THEN
77400 BANDERA:=BANDERA AND TRUE
77500 ELSE
77600 BANDERA:=FALSE;
77700 END;
77800 I:=I-K2;
77900 IF BANDERA THEN
78000 BEGIN
78100 J:=J-K2;
78200 FOR I1:=I TO I+K2 DO
78300 SOLUCI.ARRMARCA[T,I1]:='C';
78400 FOR I1:=J TO J+K2 DO
78500 SOLUCI.ARRMARCA[T+1,I1]:='C';
78600 I1:=I;
78700 I2:=J;
78800 FOR I3:=K1 TO (K1+K2) DO
78900 BEGIN
79000 PRURB[I3-K1]:=SOLUCI.ARREGLOCT,I1];
79100 PBURB[I3-K1+K2+1]:=SOLUCI.ARREGLOCT+1,I2];
79200 I1:=I1+1;
79300 I2:=I2+1;
79400 END;
79500 BURBUJA(PRURB,T1-1);
79600 IF K1>0 THEN
79700 BEGIN
79800 FOR I3:=0 TO ((K1 DIV T1) -1) DO
79900 BEGIN
80000 BANDERA:=TRUE;
80100 CONT:=0;
80200 FOR I1:=T1*I3 TO T1*(I3+1)-1 DO
80300 BEGIN

```

```

80301 (* *)
80302 (* *)
80303 (* *)
80304 (* *)
80400 IF PBURB[CONT] = SOLUCI.ARREGLOK,I1] THEN
80500 BANDERA:=BANDERA AND TRUE
80600 ELSE
80700 BANDERA:=FALSE;
80800 CONT:=CONT+1;
80900 END;
81000 IF BANDERA THEN
81100 I3:=K1;
81200 END;
81300 IF NOT BANDERA THEN
81400 BEGIN
81500 FOR I3:=K1 TO (K1+T1-1) DO
81600 SOLUCI.ARREGLOK,I3]:=PBURB[I3-K1];
81700 SOLUCI.NELEM[K1]:=SOLUCI.NELEM[K1]+T1;
81800 K1:=SOLUCI.NELEM[K1];
81900 END;
82000 ELSE
82100 BEGIN
82200 FOR I3:=0 TO (T1-1) DO
82300 SOLUCI.ARREGLOK,I3]:=PBURB[I3];
82400 SOLUCI.NELEM[K1]:=T1;
82500 K1:=T1;
82600 END;
82700 J:=J+K2;
82800 END
82900 ELSE
83000 FOR I1:=I TO I+K2 DO
83100 IF SOLUCI.ARRMARCACT,I1] <> 'C' THEN
83200 SOLUCI.ARRMARCACT,I1]:='*';
83300 BANDERA:=TRUE;
83400 END;
83500 I:=I+K2;
83600 END;
83700 T:=T+1;
83800 K:=K+1;
83900 END
84000 ELSE
84100 BEGIN
84200 FOR I:=0 TO SOLUCI.NELEM[T]-1 DO
84300 IF SOLUCI.ARRMARCACT,I] <> 'C' THEN
84400 SOLUCI.ARRMARCACT,I]:='*';
84500 T:=T+2; K:=K+2;
84600 END;
84700 K2:=T1-1;
84800 T:=(T+10) DIV 10+1;
84900 K:=(K+10) DIV 10+1;
85000 WHILE (SOLUCI.NELEM[T]=0) AND (T<70) DO
85100 BEGIN
85200 T:=T+1;
85300 K:=K+1;
85400 END;
85500 END;
85600 BURBUJA(DIGITOS,NDIGITOS);
85700 NIMPLIC:=1;
85800 T:=70;
85900 WHILE (SOLUCI.NELEM[T] = 0) AND (T>0) DO
86000 T:=T-1;
86100 MAX:=EXPOENTE(T DIV 10);
86200 WHILE (T >= 0) AND (SOLUCI.NELEM[T] <> 0) DO
86300 BEGIN
86400 MODUL:=(T DIV 10) + 1;
86500

```

```

86501 (+ *)
86502 (* *)
86503 (* +)
86504 (* *)
86600
86700 WHILE (T > MODUL) OR (T = 0) DO
86800 BEGIN
86900 J:=0;
87000 T1:=EXPONENTE(T DIV 10);
87100 WHILE J < SOLUCI.NELEM[T] DO
87200 BEGIN
87300 WHILE (SOLUCI.ARREGLOACT,J) <> '*' AND (J<SOLUCI.NELEM[T]) DO
87400 J:=J+1;
87500 IF SOLUCI.ARREGLOACT,J] = '*' THEN
87600 BEGIN
87700 FOR I:=0 TO (T1-1) DO
87800 BEGIN
87900 SOLUCI.TABIMP[NIMPLIC,I]:=SOLUCI.ARREGLOACT,J];
88000 J:=J+1;
88100 END;
88200 SOLUCI.NTABIMP[NIMPLIC]:=T1;
88300 NIMPLIC:=NIMPLIC+1;
88400 END;
88500 IF T>=0 THEN
88600 T:=T-1;
88700 WHILE (SOLUCI.NELEM[T]=J) AND (T>MODUL) DO
88800 T:=T-1;
88900 END;
89000 IF T >= 0 THEN
89100 T:=T-1;
89200 WHILE (SOLUCI.NELEM[T]=0) AND (T>0) DO
89300 T:=T-1;
89400 END;
89500 NIMPLIC:=NIMPLIC-1;
89600 FOR I:=1 TO NIMPLIC DO
89700 FOR J:=0 TO NDIGITOS DO
89800 FOR K:=0 TO (SOLUCI.NTABIMP[I]-1) DO
89900 IF SOLUCI.TABIMP[I,K] = DIGITOSE[J] THEN
90000 SOLUCI.CUBIERTO[I,J]:='0';
90100
90200 FOR I:=0 TO NDIGITOS DO
90300 BEGIN
90400 CONTCUBRE:=0;
90500 FOR J:=1 TO NIMPLIC DO
90600 IF SOLUCI.CUBIERTO[J,I] = '0' THEN
90700 BEGIN
90800 CONTCUBRE:=CONTCUBRE+1;
90900 ESTADO:=J;
91000 END;
91100 IF CONTCUBRE = 1 THEN
91200 BEGIN
91300 SOLUCI.MARCESTADO1:='*';
91400 FOR K:=0 TO NDIGITOS DO
91500 IF SOLUCI.CUBIERTO[ESTADO,K] = '0' THEN
91600 SOLUCI.CUBIERTO[NIMPLIC+1,K]:='0';
91700 END; (* IF *)
91800 END; (* FOR *)
91900 CONTCUBRE:=0;
92000 FOR I:=0 TO NDIGITOS DO
92100 IF SOLUCI.CUBIERTO[NIMPLIC+1,I] = '0' THEN
92200 CONTCUBRE:=CONTCUBRE+1;
92300 II:=1;
92400 ASOLN:=0;
92500 IF CONTCUBRE > 0 THEN
92600 BEGIN
92700 I:=I;

```

```

92701 (* *)
92702 (* *)
92703 (* *)
92704 (* *)
92800 FOR J:=0 TO NDIGITOS DO
92900 IF SOLUCI.CUBIERTO(NIMPLIC+1,J) = ' ' THEN
93000 BEGIN
93100 DIGSEC[I]:=DIGITOS[J];
93200 I:=I+1;
93300 END;
93400 NDIGSEC:=I-1;
93500 FOR J:=1 TO NIMPLIC DO
93600 IF SOLUCI.MARCC[J] <> '*' THEN
93700 SECUNDARIOS(J,I);
93800 NIMPSEC:=II-1;
93900 ACONJUNTO(NDIGSEC,NIMPSEC,SOLUCI.NTABSEC,SOLUCI.NOCUBIERTO,NSECUN);
94000 END; (* IF *)
94100 CONT:=1;
94200 FOR I:=1 TO (NIMPLIC) DO
94300 IF SOLUCI.MARCC[I] = '*' THEN
94400 SOLUMAQ(CONT,SOLUCI.NTABIMP[I],I,SOLUCI.TABIMP);
94500 FOR I:=0 TO NSECUN DO
94600 BEGIN
94700 K:=0;
94800 FOR J:=1 TO CONT-1 DO
94900 BEGIN
95000 J1:=0;
95100 WHILE J1<SOLUCI.NSOLTOTAL[J] DO
95200 BEGIN
95300 SOLUCIONES[I,K]:=SOLUCI.SOLTOTAL[J,J1];
95400 K:=K+1;
95500 J1:=J1+1;
95600 END;
95700 SOLUCIONES[I,K]:='*';
95800 K:=K+1;
95900 END;
96000 IF K < 2 THEN
96100 NSOLUCIONES[I]:=0
96200 ELSE
96300 NSOLUCIONES[I]:=K-1;
96400 END;
96500 IF CONTCUBRE > 0 THEN
96600 BEGIN
96700 FOR I:=0 TO NSECUN DO
96800 FOR J:=0 TO SOLUCI.NMATSOLEI]-1 DO
96900 BEGIN
97000 SOLUMAQ(CONT,SOLUCI.NTABSEC[SOLUCI.MATSOLEI,J],SOLUCI.MATSOLEI,J,
97100 SOLUCI.IMPSEC);
97200 IF NSOLUCIONES[I] > J THEN
97300 BEGIN
97400 SOLUCIONES[I,NSOLUCIONES[I]]:='*';
97500 J1:=NSOLUCIONES[I]+2;
97600 END
97700 ELSE
97800 J1:=0;
97900 K:=0;
98000 WHILE K<SOLUCI.NSOLTOTAL[CONT-1] DO
98100 BEGIN
98200 SOLUCIONES[I,J1]:=SOLUCI.SOLTOTAL[CONT-1,K];
98300 K:=K+1;
98400 J1:=J1+1;
98500 END;
98600 NSOLUCIONES[I]:=J1;
98700 END;
98800 END;
98900 TOTSCI:=CONT;

```

```

98901      (* *)
98902      (* *)
98903      (* *)
98904      (* *)
99000      END; (* DE SOLUCION *)
99100
99200      PROCEDURE DECIMALES(VAR DIGS:VECTOR;VAR NDIGS:INTEGER);
99300      LABEL 60;
99400      VAR
99500          AUX1,CNUMERO,CCERO,I,I1,T1,J,I2:INTEGER;
99600          MINTER:CHAR;
99700
99800      BEGIN
99900          60:
100000          READLN;
100100          READ(MINTER);
100200          J:=0;
100300          CCERO:=ORD('0');
100400          WHILE MINTER <> ',' DO
100500              BEGIN
100600                  WHILE MINTER = ' ' DO
100700                      READ (MINTER);
100800                      I:=1;
100900                      AUX1:=0;
101000                      IF ((MINTER <='9') AND (MINTER >='0')) THEN
101100                          BEGIN
101200                              WHILE ((MINTER<='9') AND (MINTER >='0')) DO
101300                                  BEGIN
101400                                      CNUMERO:=ORD(MINTER);
101500                                      AUX1:=AUX1*I+(CNUMERO - CCERO);
101600                                      I:=10*I;
101700                                      READ(MINTER);
101800                                  END;
101900                                  IF (AUX1>=POTENCIA) OR (AUX1<0) THEN
102000                                      BEGIN
102100                                          WRITE('UN NUMERO DE LOS PROPORCIONADOS, NO ESTA EN UN');
102200                                          WRITELN(' RANGO PERMISIBLE, VUELVE ');
102300                                          WRITELN('A DARLOS');
102400                                          GOTO 60;
102500                                      END
102600                                      ELSE
102700                                          IF J>0 THEN
102800                                              FOR I2:=J-1 DOWNT0 0 DO
102900                                                  IF DIGS[I2] = AUX1 THEN
103000                                                      BEGIN
103100                                                          WRITELN('HAY UN MINTERMINO REPETIDO,VUELVE A DARLOS');
103200                                                          GOTO 60;
103300                                                      END;
103400                                                      DIGS[J]:=AUX1;
103500                                                      J:=J+1;
103600                                                      WHILE MINTER = ' ' DO
103700                                                          READ(MINTER);
103800                                                          IF MINTER = ',' THEN
103900                                                              READ (MINTER);
104000                                                      END
104100                                                      ELSE
104200                                                          BEGIN
104300                                                              WRITELN('HAY UN CARACTER NO VALIDO, VUELVE A DARMELOS');
104400                                                              GOTO 60;
104500                                                          END;
104600                                                      END;
104700                                                      NDIGS:=J-1;
104800                                                  END;
104900          PROCEDURE CADENAS(VAR DIGS:VECTOR;VAR NDIGS:INTEGER;VBIN:VECTOR);
105000          LABEL 70;
105100          VAR

```

```

105101 (* *)
105102 (* *)
105103 (* *)
105104 (* *)
105200 AUX1,MINTER :CHAR;
105300 CCERO,CNUMERO,I,I1,J : INTEGER;
105400 ALFA :CAPAC;
105500 BAND : ARRAY [1..10] OF INTEGER;
105600 BEGIN
105700 70:
105800   FOR I:=0 TO 30 DO
105900     DIGSC[I]:=0;
106000   FOR I:=1 TO 10 DO
106100     BAND[I]:=0;
106200   CCERO:=ORD('0');
106300   I:=0;
106400   READLN;
106500   READ(MINTER);
106600   WHILE MINTER<>' ' DO
106700     BEGIN
106800       J:=0;
106900       WHILE (MINTER <> ',') AND (MINTER<>' ') DO
107000         BEGIN
107100           WHILE MINTER = ' ' DO
107200             READ(MINTER);
107300           IF (MINTER >= SIMBOLO.LETRAC[I]) AND (MINTER<=SIMBOLO.LETRAC[NVARIA])
107400             THEN
107500               BEGIN
107600                 READ (AUX1);
107700                 WHILE AUX1 = ' ' DO
107800                   READ(AUX1);
107900                 IF ((AUX1 = '1') OR (AUX1 = '0')) THEN
108000                   BEGIN
108100                     I1:=1;
108200                     WHILE ((MINTER<>SIMBOLO.LETRAC[I1]) AND (I1 <NVARIA+1)) DO
108300                       I1:=I1+1;
108400                     IF I1>NVARIA THEN
108500                       BEGIN
108600                         WRITELN('NO ESTA DEFINIDA UNA DE LAS VARIABLES');
108700                         GOTO 70;
108800                       END
108900                     ELSE
109000                       IF BAND[I1]=0 THEN
109100                         BEGIN
109200                           BAND[I1]:=1;
109300                           ALFA[I1]:=AUX1;
109400                           READ(MINTER);
109500                         END
109600                       ELSE
109700                         BEGIN
109800                           WRITELN('VARIABLE DOBLEMENTE DEFINIDA');
109900                           GOTO 70;
110000                         END;
110100                       END
110200                     ELSE
110300                       BEGIN
110400                         WRITELN('ALGUNO DE LOS VALORES QUE PROPORCIONASTE NO ES BINARIO');
110500                         GOTO 70;
110600                       END;
110700                     END
110800                   ELSE
110900                     BEGIN
111000                       WRITELN('ALGUNA VARIABLE NO CORRESPONDE CON LAS ASIGNADAS');
111100                       GOTO 70;
111200                     END;
111300                   J:=J+1;

```



```

111301 (* *)
111302 (* *)
111303 (* *)
111304 (* *)
111400 END;
111500 IF J <> NVARIA THEN
111600 BEGIN
111700 WRITELN('CADENA INCOMPLETA');
111800 GOTO 70;
111900 END;
112000 DIGS[I]:=0;
112100 FOR I1:=0 TO NVARIA-1 DO
112200 BEGIN
112300 CNUMERO:=ORD(ALFA[I1+1]);
112400 DIGS[I1]:=DIGS[I1] +(CNUMERO-CCERO)*VBINENVARIA-1-I1];
112500 END;
112600 IF (DIGS[I] >= POTENCIA) OR (DIGS[I] < 0) THEN
112700 BEGIN
112800 WRITELN('UNO DE LOS DIGS NO ES COMPATIBLE CON EL NUMERO DE',
112900 ' VARIABLES');
113000 GOTO 70;
113100 END
113200 ELSE
113300 IF I > C THEN
113400 FOR I1:=I-1 DOWNTO C DO
113500 IF DIGS[I1] = DIGS[I] THEN
113600 BEGIN
113700 WRITELN('HAY UN MINTERMINO REPETIDO, VUELVE A DARMELOS');
113800 GOTO 70;
113900 END;
114000 FOR I1:=1 TO 10 DO
114100 BAND[I1]:=0;
114200 I:=I+1;
114300 IF MINTER<>',' THEN
114400 READ(MINTER);
114500 END;
114600 NDIGS:=I-1;
114700 END;
114800
114900 PROCEDURE VALBIN(VAR DIGS:VECTOR;VAR NDIGS:INTEGER;VBIN:VECTOR);
115000 LABEL 80;
115100 VAR
115200 MINTER : CHAR;
115300 CCERO,CNUMERO,I1,I2,J,I : INTEGER;
115400 CADBIN : CARAC;
115500 BEGIN
115600 80:
115700 READLN;
115800 READ (MINTER);
115900 CCERO:=ORD('0'); J:=0;
116000 WHILE (MINTER <> ',') DO
116100 BEGIN
116200 I:=1;
116300 WHILE (MINTER<>',' ) AND (MINTER<>',' ) DO
116400 BEGIN
116500 WHILE MINTER = ' ' DO
116600 READ (MINTER);
116700 IF ((MINTER = '0') OR (MINTER = '1')) THEN
116800 BEGIN
116900 CADBIN[I]:=MINTER;
117000 I:=I+1;
117100 READ(MINTER);
117200 END
117300 ELSE
117400 BEGIN
117500 WRITELN ('ERROR, CARACTER NO RECONOCIDO');

```

```

117501 (* *)
117502 (* *)
117503 (* *)
117504 (* *)
117600 GOTO 81;
117700 END;
117800 IF I=(NVARIA+1) THEN
117900 BEGIN
118000 DIGS[J]:=0;
118100 I:=1;
118200 FOR I2:=0 TO NVARIA-1 DO
118300 BEGIN
118400 CAUMERO:=ORD(CADRIN[I2+1]);
118500 DIGS[I2]:=DIGS[J]+(CAUMERO-CERO)*VRIN[NVARIA-1-I2];
118600 END;
118700 IF (DIGS[J] >= POTENCIA) OR (DIGS[J] < 0) THEN
118800 BEGIN
118900 WRITELN('UNO DE LOS DIGS NO ES COMPATIBLE CON EL NUMERO DE',
119000 ' VARIABLES');
119100 GOTO 80;
119200 END
119300 ELSE
119400 IF J> THEN
119500 FOR I2:=(J-1) DOWNTO 0 DO
119600 IF DIGS[I2] = DIGS[J] THEN
119700 BEGIN
119800 WRITELN('UNO DE LOS MINTERMINOS SE REPITE');
119900 GOTO 80;
120000 END;
120100 FOR I2:=1 TO 5 DO
120200 CADRIN[I2]:=' ';
120300 J:=J+1;
120400 END
120500 ELSE
120600 BEGIN
120700 WRITELN('ERROR, MINTERMINO NO COMPLETO O EXCEDIDO');
120800 GOTO 80;
120900 END;
121000 IF MINTER <> ' ' THEN
121100 RFAC (MINTER);
121200 WRITELN(MINTER);
121300 END;
121400 NDIGS:=J-1;
121500 END;
121600
121700 PROCEDURE OPCIONES(VAR DIGITOS:VECTOR; VAR NDIGITOS: INTEGER);
121800 BEGIN
121900 WRITELN('TIENES LAS SIGUIENTES OPCIONES PARA PROPORCIONAR LOS ',
122000 'MINTERMINOS');
122100 WRITELN(' 1: POR MEDIO DE CADENAS DE CARACTERES COMO A CONTINUA',
122200 'CION SE DESCRIBEN');
122300 WRITELN(' POR CADA VARIABLE SE PROPORCIONAN DOS CARACTERES:');
122400 WRITELN(' EL PRIMER CARACTER INDICA EL NOMBRE DE LA ',
122500 'VARIABLE, EL SEGUNDO ');
122600 WRITELN(' CARACTER INDICA EL VALOR BINARIO ASOCIADO A ',
122700 'ESTA VARIABLE. ');
122800 WRITELN(' EJEMPLO:');
122900 WRITELN(' *** PARA TRES VARIABLES DE SECUENCIA CBA, UN ',
123000 'MINTERMINO PODRIA');
123100 WRITELN(' SEP 011, ES DECIR, C=0, B=1, A=1, PARA',
123200 ' INDICAR ESTO, SE DEBE ');
123300 WRITELN(' PROPORCIONAR C0B1A1');
123400 WRITELN(' 2: POR MEDIO DE UNA CADENA DE CARACTERES BINARIOS,');

```

```

123701 (* *)
123702 (* *)
123703 (* *)
123704 (* *)
123800 ' TENIENDO EN CUENTA ');
123900 WRITELN(' QUE CADA DIGITO BINARIO ESTA ASOCIADO A UNA ',
124000 'VARIABLE POR SU ');
124100 WRITELN(' POSICION. EJEMPLO:');
124200 WRITELN(' **** PARA TRES VARIABLES DE SECUENCIA BAC, SI SE ',
124300 'PROPORCIONA B10,');
124400 WRITELN(' SIGNIFICA B=0, A=1, C=0');
124500 WRITELN;
124600 WRITELN(' 3: PROPORCIONANDO EL VALOR DECIMAL DE LOS MINTERMIOS',
124700 'EJEMPLO:');
124800 WRITELN(' **** PARA INDICAR EN UN MAPA DE CUATRO VARIA',
124900 'BLES DE SECUENCIA ');
125000 WRITELN(' HACD EL MINTERMINO B=C, A=0, C=1, D=1, ES ',
125100 'DECIP, SE DEBE ');
125200 WRITELN(' PROPORCIONAR EL EQUIVALENTE DECIMAL DE (011, ',
125300 'ESTO ES, 3');
125400 WRITELN;
125500 WRITELN('LOS MINTERMIOS DEBERAN SEPARARSE POR COMAS, TERMINAR',
125600 'CON ";" Y RETURN');
125700 REPEAT
125800 WRITELN(' * AHORA PROPORCIONA EL NUMERO DE LA OPCION FLEGIDA');
125900 READLN; READ(OPCION);
126000 WRITELN;
126100 IF OPCION = '1' THEN
126200 BEGIN
126300 WRITELN('RECUERDA QUE POR CADA VARIABLE SE PROPOR',
126400 'CIONAN DOS CARACTERES: ');
126500 WRITELN('LA VARIABLE Y SU VALOR BINARIO ASOCIADO');
126600 Cadenas(DIGITOS,NDIGITOS,SIMBOLO,NOASOC)
126700 END
126800 ELSE
126900 IF OPCION = '3' THEN
127000 BEGIN
127100 WRITELN('PROPORCIONA LOS MINTERMIOS EN SU EQUIVALENTE ',
127200 'DECIMAL SEPARADOS POR ');
127300 WRITELN('COMAS, PARA FINALIZAR TRANSMITE (;)');
127400 Decimales(DIGITOS,NDIGITOS);
127500 END
127600 ELSE
127700 IF OPCION = '2' THEN
127800 BEGIN
127900 WRITELN('PROPORCIONA LOS MINTERMIOS EN FORMA BINARIA');
128000 VALPIN(DIGITOS,NDIGITOS,SIMBOLO,NOASOC)
128100 END
128200 ELSE
128300 WRITELN('OPCION INCORRECTA');
128400 UNTIL (OPCION > '4') AND (OPCION < '1');
128500 IF NDIGITOS = -1 THEN
128600 BEGIN
128700 WRITELN('TU SOLUCION ES UNA ANTI-AUTOLOGIA, ES DECIR, 0');
128800 APORT;
128900 END;
129000 END; (* OPCIONES *)
129100
129200 PROCEDURE MAPABINARIO;
129300 VAR I,J,K,L,DIR: INTEGER;
129400 BEGIN
129500 DIR:=0;
129600 FOR I:=0 TO NI-1 DO
129700 FOR J:=0 TO NJ-1 DO
129800 BEGIN
129900 L:="APAS[I,J];

```

```

129901 (* *)
129902 (* *)
129903 (* *)
129904 (* *)
130000 FOR K:=1 TO NVARIA-1 DO
130100 BEGIN
130200 SIMBOLO.BINARIO[DIR,K]:=L MOD 2;
130300 L:=L DIV 2;
130400 END;
130500 DIR:=DIR+1;
130600 END;
130700 END; (* MAPBINARIO *)
130800
130900 PROCEDURE MAPACTUAL;
131000 VAR PARAMAPA,I,J,I1,J1 : INTEGER;
131100 AUX1 : CHAR;
131200 BEGIN
131300 FOR I:=1 TO NVARIA DO
131400 AUXILIAR[I]:=SIMBOLO.LETRA[I];
131500 FOR I:=1 TO NVARIA-1 DO
131600 FOR J:=I+1 TO NVARIA DO
131700 IF AUXILIAR[I] > AUXILIAR[J] THEN
131800 BEGIN
131900 AUX1:=AUXILIAR[I];
132000 AUXILIAR[I]:=AUXILIAR[J];
132100 AUXILIAR[J]:=AUX1;
132200 END;
132300 I:=1;
132400 FOR J:=0 TO (NVARIA-1) DO
132500 BEGIN
132600 WHILE SIMBOLO.LETRA[J+1] <> AUXILIAR[I] DO
132700 I:=I+1;
132800 CASE I OF
132900 1 : SIMBOLO.NOASOC[J]:=1;
133000 2 : SIMBOLO.NOASOC[J]:=2;
133100 3 : SIMBOLO.NOASOC[J]:=4;
133200 4 : SIMBOLO.NOASOC[J]:=8;
133300 5 : SIMBOLO.NOASOC[J]:=16;
133400 6 : SIMBOLO.NOASOC[J]:=32;
133500 END;
133600 I:=1;
133700 END;
133800 I:=0;
133900 FOR I1:=0 TO NI-1 DO
134000 FOR J1:=0 TO NJ-1 DO
134100 BEGIN
134200 PARAMAPA:=0;
134300 FOR J:=0 TO (NVARIA-1) DO
134400 PARAMAPA:=PARAMAPA+SIMBOLO.NOASOC[J]*SIMBOLO.
134500 BINARIO[I,J];
134600 MAPAS[I1,J1]:=PARAMAPA;
134700 I:=I+1;
134800 END;
134900 END; (* MAPACTUAL *)
135000
135100 PROCEDURE MAPASC1;
135200 BEGIN
135300 FOR I:=0 TO NI-1 DO
135400 FOR J:=0 TO NJ-1 DO
135500 FOR K:=0 TO NDIGITOS DO
135600 IF DIGITOS[K] = MAPAS[I,J] THEN
135700 MAPASYN[I,J]:=1;
135800 IF BANDERA2 THEN
135900 FOR I:=0 TO NI-1 DO
136000 FOR J:=0 TO NJ-1 DO
136100 FOR K:=0 TO NDIGITOS DO

```

```

136101 (* *)
136102 (* *)
136103 (* *)
136104 (* *)
136200 IF (DIGITOS[K] = MAPAS[I,J]) AND (CDIGITOS[K] = 'X') THEN
136300   CMAPAS[MICI,J]:= 'X';
136400 END; (* MAPAS *)
136500
136600 PROCEDURE BUSCA(VAR T,UBICA:INTEGER; INICIO,FIN:INTEGER);
136700 VAR
136800   CONT,CONT1: INTEGER;
136900 BEGIN
137000   WHILE (SOLUCI.NELEMEN[T] = 0) AND (T > (T1+10)) DO
137100     T:=T-1;
137200     BANDERA:=FALSE;
137300     IF T > T1+10 THEN
137400       WHILE (NOT BANDERA) AND ((T>T1+10) OR (T=0)) DO
137500         BEGIN
137600           FOR CONT:=0 TO SOLUCI.NELEMEN[T]-1 DO
137700             BEGIN
137800               IF SOLUCI.AREGLO[T,CONT] = CASILLAS[INICIO] THEN
137900                 BEGIN
138000                   BANDERA:=TRUE;
138100                   FOR CONT1:=INICIO TO FIN DO
138200                     BEGIN
138300                       IF SOLUCI.AREGLO[T,CONT]=CASILLAS[CONT1] THEN
138400                         BANDERA:=BANDERA AND TRUE
138500                       ELSE
138600                         BANDERA:=FALSE;
138700                       CONT:=CONT+1;
138800                     END;
138900                   CONT:=CONT-1;
139000                 END
139100               ELSE
139200                 BEGIN
139300                   BANDERA:=FALSE;
139400                   CONT:=CONT+T1;
139500                 END;
139600               IF BANDERA THEN
139700                 BEGIN
139800                   UBICA:=CONT-T1;
139900                   CONT:=SOLUCI.NELEMEN[T];
140000                 END;
140100             END;
140200           T:=T-1;
140300         END
140400       END; (* BUSCA *)
140500
140600 (* PROGRAMA PRINCIPAL *)
140700 BEGIN
140800   WRITELN('ESTE PROGRAMA TE GUIARA EN LA MINIMIZACION DE ',
140900     'FUNCIONES POR MEDIO DEL');
141000   WRITELN('METODO DE KARNAUGH. ');
141100   WRITELN('DESEAS CONOCER UNA PEQUERA INTRODUCCION A ESTE ',
141200     'METODO? ');
141300   READ(CAMBIO[1],CAMBIO[2]);
141400   WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
141500     ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
141600     BEGIN
141700       WRITELN('REPITE SI DESEAS O NO HACERLO');
141800       READLN; READ(CAMBIO[1],CAMBIO[2]);
141900     END;
142000   IF CAMBIO[1] = 'S' THEN
142100     TEXTO;
142200   DIGITOS[0]:=0;
142300   NDIGITOS:=0;

```

```

142301 (* *)
142302 (* *)
142303 (* *)
142304 (* *)
142400 ANDPAP:=FALSE;
142500 DIM:=(1,2,4,15,32,64);
142600 FOR I:=0 TO R DO
142700 FOR J:=0 TO R DO
142800 BEGIN
142900 MAPASMIN[I,J]:=' ';
143000 MAPASMIN[I,JJ]:=1;
143100 END;
143200 WRITELN('CUANTAS VARIABLES DESEAS QUE TENGA TU FUNCION?');
143300 READLN; READ(NVARIA);
143400 WHILE ((NVARIA < 1) OR (NVARIA > 6)) DO
143500 BEGIN
143600 IF NVARIA < 1 THEN
143700 WRITELN('CON ESTE PROGRAMA PUEDES RESOLVER FUNCIONES ',
143800 "REALES", REPITE POR FAVOR');
143900 ELSE
144000 BEGIN
144100 WRITELN('ESTE PROGRAMA TIENE FINES DIDACTICOS Y ESTA ',
144200 'SUJETO AL ESPACIO DE LA ');
144300 WRITELN('PANTALLA, POR LO CUAL RESUELVE FUNCIONES CON EN MAXI',
144400 'MO DE 6 VARIABLES');
144500 WRITELN('DESEAS PROPONER ALGUNA FUNCION QUE PUEDAS ',
144600 'RESOLVER A TRAVES DE ESTE ');
144700 WRITELN('PROGRAMA?');
144800 READLN; READ(CAMBIO[1],CAMBIO[2]);
144900 WHILE NOT (((CAMBIO[1]='S')AND(CAMBIO[2]='I')) OR ((CAMBIO[1]
145000 ='N') AND (CAMBIO[2]='O'))) DO
145100 BEGIN
145200 WRITELN('REPITE SI DESEAS PROPONER OTRA FUNCION');
145300 READLN; READ(CAMBIO[1],CAMBIO[2]);
145400 END;
145500 IF ((CAMBIO[1]='N') AND (CAMBIO[2]='O')) THEN
145600 ABORT
145700 END;
145800 WRITELN('CUANTAS VARIABLES ENTONCES DESEAS USAR?');
145900 READLN; READ(NVARIA);
146000 WRITELN;
146100 END;
146200 SIMBLO.LETRAC1:='A';
146300 FOR I:=2 TO NVARIA DO
146400 BEGIN
146500 SIMBLO.LETRACI:=SUCC(SIMBLO.LETRAC[I-1]);
146600 AUXILIARCI:=SIMBLO.LETRACI;
146700 END;
146800 DEFHV(NVARIA,NJ,NI);
146900 FOR I:=0 TO NI-1 DO
147000 FOR J:=0 TO NJ-1 DO
147100 MAPAS[I,J]:=NJ*I + J;
147200 IF NJ>2 THEN
147300 BEGIN
147400 HORIZONTAL2(MAPAS);
147500 IF (NJ>4) THEN
147600 BEGIN
147700 HORIZONTAL4(MAPAS);
147800 VERTICAL2(MAPAS);
147900 END
148000 ELSE IF NI>2 THEN
148100 VERTICAL2(MAPAS);
148200 IF (NI>4) THEN
148300 VERTICAL4(MAPAS);
148400 END;
148500 MAPASIMAPID;

```

```

148501 (* *)
148502 (* *)
148503 (* *)
148504 (* *)
148600 POTENCIA:=EXPONENTE(NVARIA);
148700 BANDFRA:=FALSE;
148800 GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
148900 WRITELN;
149000 WRITELN('EL MAPA REPRESENTA LAS POSICIONES DE TODOS LOS',
149100 ' MINTERMINOS POSIBLES PARA ');
149200 WRITELN('EL NUMERO DE VARIABLES QUE ELEGISTE ');
149300 WRITELN;
149400 WRITELN('DESEAS RENOMBRAR LAS VARIABLES? ');
149500 READLN; READ(CAMBIO[1], CAMBIO[2]);
149600 WRITELN;
149700 WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
149800 ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
149900 BEGIN
150000 WRITELN('REPITE SI DESEAS RENOMBRARLAS O NO');
150100 READLN; READ(CAMBIO[1],CAMBIO[2]);
150200 END;
150300 BANDERA:=FALSE;
150400 IF (CAMBIO[1]='S') AND (CAMBIO[2]='I') THEN
150500 BEGIN
150600 WHILE NOT BANDERA DO
150700 BEGIN
150800 BANDERA:=TRUE;
150900 WRITELN('SE TE DARA A CONTINUACION CADA VARIABLE,',
151000 ' ENSEGUIDA DEBERAS ASIGNARLE EL ');
151100 WRITELN('NUEVO (UN CARACTER)');
151200 FOR I:=1 TO NVARIA DO
151300 BEGIN
151400 WRITE(SIMBOLO.LETRAC[I], '=');
151500 READLN; READ(AUXILIAR[I]);
151600 IF (AUXILIAR[I]=' ') THEN
151700 AUXILIAR[I]:=SIMBOLO.LETRAC[I];
151800 WRITELN;
151900 IF (AUXILIAR[I] > 'Z') OR (AUXILIAR[I] < 'A') THEN
152000 BANDERA:=FALSE;
152100 END;
152200 BANDERA:=TRUE;
152300 FOR I:=1 TO (NVARIA-1) DO
152400 FOR J:=(I+1) TO NVARIA DO
152500 IF AUXILIAR[I]=AUXILIAR[J] THEN
152600 BANDERA:=FALSE;
152700 IF NOT BANDERA THEN
152800 BEGIN
152900 WRITELN('REPITE TUS VARIABLES POR FAVOR, HAY UN ',
153000 ' CARACTER NO CONOCIDO O BIEN UNA DE');
153100 WRITELN('TUS VARIABLES ESTA PEPETIDA');
153200 END;
153300 END;
153400 FOR I:=1 TO NVARIA DO
153500 SIMBOLO.LETRAC[I]:=AUXILIAR[I];
153600 WRITELN('EL MAPA GENERADO CON TUS NUEVAS VARIABLES ',
153700 ' TIENE LA SIGUIENTE ESTRUCTURA');
153800 DEFPV(NVARIA,NJ,'I');
153900 MAPAFACTUAL;
154000 BANDFRA:=FALSE;
154100 GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
154200 END;
154300 MAPAFACTUAL;
154400 OPCIONES(DIGITOS,NDIGITOS);
154500 3: FOR CONT:=0 TO NDIGITOS DO
154600 CDIGITOS[CONT]:=' ';
154700 FOR CONT:=0 TO NT-1 DO

```

```

154701 (* *)
154702 (* *)
154703 (* *)
154704 (* *)
154801 FOR CONT1:=1 TO 'J'-1 DO
154901   CMAPASMIN[CONT,CONT1]:= ' ';
155001   WRITELN;
155101   WRITELN('CON LAS POSICIONES DE LOS MINTERMINOS, EL MAPA ',
155201     'DEFINIDO SE PRESENTA A LA ');
155301   WRITELN('DIRECCION');
155401   MAPAS1;
155501   BANDERA:=TRUE;
155601   GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
155701   SOLUCION;
155801   WRITELN;
155901   WRITELN('DE ACUERDO AL MAPA ANTERIOR, PROPORCIONA LOS MINTERMINOS '
156001     'QUE SE PUEDEN');
156101   WRITELN('COMBINAR, CADA UNO SEPARADO POR (,) Y TERMINAR CON (;)');
156201   WRITELN;
156301   WRITELN('EXISTEN MINTERMINOS QUE SE COMBINEN?');
156401   READLN; READ(CAMBIO[1],CAMBIO[2]);
156501   WHILE NOT (((CAMBIO[1] = 'S') AND (CAMBIO[2] = 'I')) OR
156601     ((CAMBIO[1] = 'N') AND (CAMBIO[2] = 'O')))) DO
156701     BEGIN
156801       WRITELN('REPITE SI HAY MINTERMINOS QUE SE COMBINEN');
156901       READLN; READ(CAMBIO[1],CAMBIO[2]);
157001     END;
157101     CONTAB:=1;
157201     CONT2:=0;
157301     WHILE ((CAMBIO[1] = 'S') OR (CAMBIO[1]='N')) DO
157401       IF CAMBIO[1] = 'S' THEN
157501         WHILE CAMBIO[1]='S' DO
157601           BEGIN
157701             WRITELN('PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA',
157801               'FORMA INDICADA');
157901             CASILLAS[0]:=0;
158001             NCASILLAS:=0;
158101             DECIMALES(CASILLAS,NCASILLAS);
158201             IF NCASILLAS > 0 THEN
158301               IF (NCASILLAS+1) IN UNO THEN
158401                 BEGIN
158501                   BURBUJA(C/CASILLAS,NCASILLAS);
158601                   T1:=PAIZ(NCASILLAS+1);
158701                   T:=T1*10 + 10;
158801                   WHILE (SOLUCI.NFLEM[T] = 0) AND (T > (T1*10)) DO
158901                     T:=T-1;
159001                   BANDERA:=FALSE;
159101                   IF T > T1*10 THEN
159201                     WHILE (NOT BANDERA) AND (T>T1*10) DO
159301                       BEGIN
159401                         FOR CONT:=0 TO SOLUCI.NELEM[T]-1 DO
159501                           BEGIN
159601                             IF SOLUCI.ARREGLO[T,CONT] = CASILLAS[0] THEN
159701                               BEGIN
159801                                 BANDERA:=TRUE;
159901                                 FOR CONT1:=1 TO NCASILLAS DO
160001                                   BEGIN
160101                                     IF SOLUCI.ARREGLO[T,CONT]=CASILLAS[CONT1] THEN
160201                                       BANDERA:=BANDERA AND TRUE
160301                                     ELSE
160401                                       BANDERA:=FALSE;
160501                                     CONT:=CONT+1;
160601                                   END;
160701                                 CONT:=CONT-1;
160801                               END
160901                             ELSE

```



```

160971 (* *)
160982 (* *)
160993 (* *)
160994 (* *)
161000 BEGIN
161100     BANDERA:=FALSE;
161200     CONT:=CONT+NCASILLAS;
161300     END;
161400     IF BANDERA THEN
161500         CONT:=SOLUCI.NELEMCT];
161600     END;
161700     T:=T-1;
161800 END;
161900 IF BANDERA THEN
162000 BEGIN
162100     FOR CONT1:=0 TO NCASILLAS DO
162200     FOR CONT2:=0 TO NDIGITOS DO
162300         IF CASILLAS[CONT1] = DIGITOS[CONT2] THEN
162400             IF CASDIG[CONT2] <> 'C' THEN
162500                 BANDERA:=FALSE;
162600     IF NOT BANDERA THEN
162700     BEGIN
162800         FOR CONT:=0 TO NCASILLAS DO
162900             TABALUM[CONTAB,CONT]:=CASILLAS[CONT];
163000             NTABALUM[CONTAB]:=NCASILLAS+1;
163100             CONTAB:=CONTAB+1;
163200         FOR CONT1:=0 TO NCASILLAS DO
163300         FOR CONT:=0 TO NDIGITOS DO
163400             IF CASILLAS[CONT1] = DIGITOS[CONT] THEN
163500                 CASDIG[CONT]:='C';
163600             BANDERA:=TRUE;
163700             WRITELN('BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN');
163800         FOR CONT:=0 TO NCASILLAS DO
163900         FOR CONT1:=0 TO NDIGITOS DO
164000             IF CASILLAS[CONT1] = DIGITOS[CONT] THEN
164100                 CDIGITOS[CONT1]:='X';
164200             BANDERA2:=TRUE;
164300             MAPAS1;
164400             GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
164500     END
164600     ELSE
164700     BEGIN
164800         WRITELN('TODOS LOS MINTERMINOS QUE PRPONEN YA ESTAN CUBI',
164900             'ERTOS NO LOGRAS *MINIMIZAR*');
165000         WRITELN('MAS TU FUNCION');
165100     END;
165200     WRITELN('AUN HAY MINTERMINOS QUE SE COMBINEN?');
165300     READLN; READ(CAMBIO[1],CAMBIO[2]);
165400     WHILE NOT (((CAMBIO[1] = 'S') AND (CAMBIO[2] = 'I')) OR
165500         ((CAMBIO[1] = 'N') AND (CAMBIO[2] = 'O'))) DO
165600     BEGIN
165700         WRITELN('REPITE SI AUN HAY MINTERMINOS QUE SE COMBINEN O NO');
165800         READLN; READ(CAMBIO[1],CAMBIO[2]);
165900     END
166000     ELSE
166100     BEGIN
166200     BEGIN
166300         BANDERA1:=TRUE;
166400         FOR CONT:=0 TO NCASILLAS DO
166500         BEGIN
166600             BANDERA:=TRUE;
166700             FOR CONT1:=0 TO NDIGITOS DO
166800             IF CASILLAS[CONT] = DIGITOS[CONT1] THEN
166900                 BANDERA:=FALSE;
167000             IF BANDERA THEN
167100                 BANDERA1:=FALSE;

```

```

167101 (+ *)
167102 (+ *)
167103 (+ *)
167104 (+ *)
167200 END;
167300 IF NOT BANDERA1 THEN
167400 WRITELN('UNO DE LOS NUMEROS QUE PROPORCIONASTE NO ES',
167500 ' MINTERMINO')
167600 ELSE
167700 WRITELN('LOS MINTERMINOS QUE PROPORCIONASTE NO SE PUEDEN',
167800 ' COMBINAR');
167900 BANDERA:=FALSE;
168000 END;
168100 IF CAMBIO[1] = 'N' THEN
168200 BEGIN
168300 FOR CONT:= 1 TO NDIGITOS DO
168400 IF CAMBIO[CONT] = 'C' THEN
168500 BANDERA:=BANDERA AND TRUE
168600 ELSE
168700 BANDERA:=FALSE;
168800 IF NOT BANDERA THEN
168900 BEGIN
169000 WRITELN('AHN HAY MINTERMINOS POR CUBRIR, INTENTALO');
169100 WRITELN('SE PUEDEN COMBINAR ENTRE ELLOS',
169200 ' LOS MINTERMINOS QUE NO SE HAN CUBIERTO?');
169300 READLN; READ(CAMBIO[1],CAMBIO[2]);
169400 WHILE NOT ((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
169500 ((CAMBIO[1]='N') AND (CAMBIO[2]='O')) DO
169600 BEGIN
169700 WRITELN('SE PUEDEN COMBINAR ENTRE ELLOS O NO?');
169800 READLN; READ(CAMBIO[1],CAMBIO[2]);
169900 END;
170000 END
170100 ELSE
170200 CAMBIO[1]='F';
170300 END
170400 ELSE
170500 BEGIN
170600 BANDERA1:=TRUE;
170700 FOR CONT:=1 TO NDIGITOS DO
170800 IF CDIGITOS[CONT] <> 'X' THEN
170900 BANDERA1:=FALSE;
171000 IF BANDERA1 THEN
171100 BEGIN
171200 WRITE('TODOS LOS MINTERMINOS YA ESTAN CUBIERTOS, POR ',
171300 ' LO CUAL ');
171400 CAMBIO[1]='F';
171500 END;
171600 END;
171700 END
171800 ELSE
171900 BEGIN
172000 WRITELN('EL NUMERO DE MINTERMINOS NO ES POTENCIA DE 2,',
172100 ' POR LO CUAL NO SE PUEDEN ');
172200 WRITELN('COMBINAR');
172300 END
172400 ELSE
172500 BEGIN
172600 WRITELN('SI SOLO SE TRATA DE UN MINTERMINO, QUIERE DECIR QUE NO',
172700 ' SE PUEDE COMBINAR, ');
172800 WRITELN('REPITFLO');
172900 CAMBIO[1]='N';
173000 END
173100 END (* WHILE DE SI *)
173200 ELSE
173300 BEGIN

```

```

173301 (* *)
173302 (* *)
173303 (* *)
173304 (* *)
173400 WRITELN('PROPORCIONA CADA MINTERMINO QUE NO SE COMBINO, ',
173500 '1 A LA VEZ');
173600 WHILE CAMBIO[1]='M' DO
173700 BEGIN
173800 DECIMALES(CASILLAS,NCASILLAS);
173900 WHILE NCASILLAS <> 0 DO
174000 BEGIN
174100 WRITELN('SOLO 1 A LA VEZ, REPITE');
174200 DECIMALES(CASILLAS,NCASILLAS);
174300 END;
174400 T1:=0;
174500 T:=10;
174600 WHILE ((SOLUCI.NELEM[T]=0) AND (T>=0)) DO
174700 T:=T-1;
174800 BANDERA:=FALSE;
174900 WHILE (NOT BANDERA) AND (T>=0) DO
175000 BEGIN
175100 FOR CONT:=0 TO SOLUCI.NELEM[T]-1 DO
175200 IF SOLUCI.ARREGLO[T,CONT]=CASILLAS[0] THEN
175300 IF SOLUCI.ARRMARCA[T,CONT]='*' THEN
175400 BEGIN
175500 CONT:=SOLUCI.NELEM[T];
175600 BANDERA:=TRUE;
175700 END
175800 ELSE
175900 BEGIN
176000 WRITELN('EL MINTERMINO ',CASILLAS[0]:2,' SI SE PUEDE ',
176100 'COMBINAR, INTENTALO');
176200 CAMBIO[1]='S';
176300 T:=0;
176400 END;
176500 T:=T-1;
176600 END;
176700 IF BANDERA THEN
176800 BEGIN
176900 FOR CONT1:=0 TO NCASILLAS DO
177000 FOR CONT2:=0 TO NDIGITOS DO
177100 IF CASILLAS[CONT1] = DIGITOS[CONT2] THEN
177200 IF CASDIG[CONT2] <> 'C' THEN
177300 BANDERA:=FALSE;
177400 IF NOT BANDERA THEN
177500 BEGIN
177600 FOR CONT:=0 TO NCASILLAS DO
177700 TABALUM[CONTAB,CONT]:=CASILLAS[CONT];
177800 NTARALUM[CONTAB]:=NCASILLAS+1;
177900 CONTAB:=CONTAB+1;
178000 FOR CONT1:=0 TO NCASILLAS DO
178100 FOR CONT:=0 TO NDIGITOS DO
178200 IF CASILLAS[CONT1] = DIGITOS[CONT] THEN
178300 CASDIG[CONT1]='C';
178400 WRITELN('BIEN, EL MINTERMINO NO SE COMBINA, POR LO CUAL ',
178500 'NO SE REDUCE');
178600 BANDERA:=TRUE;
178700 FOR CONT:=0 TO NCASILLAS DO
178800 FOR CONT1:=0 TO NDIGITOS DO
178900 IF CASILLAS[CONT] = DIGITOS[CONT1] THEN
179000 CDIGITOS[CONT1]='X';
179100 BANDERA2:=TRUE;
179200 MAPAS01;
179300 GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
179400 END
179500 ELSE

```

```

179501 (* *)
179502 (* *)
179503 (* *)
179504 (* *)
179505 BEGIN
179506     WRITELN('TODOS LOS MINTERMINOS QUE PROPONES YA ESTAN ',
179507             'CUBIERTOS, NO LOGRAS MINIMIZAR');
179508     WRITELN('MAS TU FUNCION');
179509 END;
179510 WRITELN('AUN HAY MINTERMINOS QUE NO SE COMBINEN?');
179511 READLN; READ(METODO[1],METODO[2]);
179512 WHILE NOT (((METODO[1] = 'S') AND (METODO[2] = 'I')) OR
179513             ((METODO[1] = 'N') AND (METODO[2] = 'O'))) DO
179514 BEGIN
179515     WRITELN('REPITE SI AUN HAY MINTERMINOS QUE NO SE COMBINEN');
179516     READLN; READ(METODO[1],METODO[2]);
179517 END;
179518 IF METODO[1]='S' THEN
179519 BEGIN
179520     BANDERA1:=TRUE;
179521     FOR CONT:=0 TO NDIGITOS DO
179522     IF CDIGITOS[CONT] <> 'C' THEN
179523         BANDERA1:=FALSE;
179524     IF BANDERA1 THEN
179525     BEGIN
179526         WRITE('TODOS LOS MINTERMINOS YA ESTAN CUBIERTOS, POR ',
179527               'LO CUAL ');
179528         CAMBIO[1]:='F';
179529     END
179530     ELSE
179531     BEGIN
179532         WRITELN('PROPORCIONAME EL SIGUIENTE');
179533         CAMBIO[1]:='N'
179534     END;
179535 END
179536 ELSE
179537 BEGIN
179538     FOR CONT:=0 TO NDIGITOS DO
179539     IF CADIGITOS[CONT] = 'C' THEN
179540         BANDERA:=BANDERA AND TRUE
179541     ELSE
179542         BANDERA:=FALSE;
179543     IF NOT BANDERA THEN
179544     BEGIN
179545         WRITELN('AUN HAY MINTERMINOS POR CUBRIR, INTENTALO');
179546         WRITELN('SE PUEDEN COMBINAR ENTRE ELLOS LOS MINTERMINOS ',
179547                 'QUE FALTAN POR CUBRIR?');
179548         READLN; READ(CAMBIO[1],CAMBIO[2]);
179549         WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
179550                   ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
179551         BEGIN
179552             WRITELN('SE PUEDEN COMBINAR ENTRE ELLOS O NO?');
179553             READLN; READ(CAMBIO[1],CAMBIO[2]);
179554         END;
179555     END
179556     ELSE CAMBIO[1]:='F';
179557 END;
179558 END;
179559 IF (NOT BANDERA) AND (T<0) AND (CAMBIO[1]='N') THEN
179560 WRITELN('EL PROPUESTO NO ES MINTERMINO, DAME OTRO QUE SI LO SEA');
179561 END
179562 END;
179563 IF BANDERA THEN
179564 BEGIN
179565     CONT:=IGTOS;
179566     CONT:=1;

```

```

185701 (* *)
185702 (* *)
185703 (* *)
185704 (* *)
185800 FOR CONT1:=1 TO CONTAB-1 DO
185900 BEGIN
186000 SOLUMAB(CONT,NTABALUM[CONT1],CONT1,TABALUM);
186100 CONT4:=0;
186200 FOR CONT3:=CONT2 TO (SOLUCI.NSOLTOTAL[CONT-1]+CONT2) DO
186300 BEGIN
186400 SOLALUM[CONT3]:=SOLUCI.SOLTOTALE[CONT-1,CONT4];
186500 CONT4:=CONT4+1;
186600 END;
186700 SOLALUM[CONT3]:='+';
186800 CONT2:=CONT3+1;
186900 END;
187000 WRITELN('TU SOLUCION ES:');
187100 CONT1:=0;
187200 CONT3:=0;
187300 IF (CONTAB=2) AND (SOLUCI.NSOLTOTAL[CONT-1]=0) THEN
187400 WRITELN('UNA TAUTOLOGIA, ES DECIR, 1')
187500 ELSE
187600 WHILE (CONT1<(CONT2-1)) DO
187700 BEGIN
187800 WHILE (CONT1 < (CONT2-1)) AND (CONT3 < 75) DO
187900 BEGIN
188000 WRITE(SOLALUM[CONT1]:1);
188100 CONT1:=CONT1+1;
188200 CONT3:=CONT3+1;
188300 END;
188400 WRITELN;
188500 CONT3:=0;
188600 END;
188700 WRITELN;
188800 END;
188900 IF NSECUN = 0 THEN
189000 IF NSOLUCIONES[0]>=(CONT2-1) THEN
189100 WRITELN('PERFECTO, OBTUVISTE LA SOLUCION OPTIMA')
189200 ELSE
189300 BEGIN
189400 WRITELN('TU SOLUCION ES CORRECTA PERO NO LA OPTIMA ',
189500 'DESEAS INTENTAR LLEGAR A ELLA?');
189600 READLN; READ(CAMBIO[1],CAMBIO[2]);
189700 WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
189800 ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
189900 BEGIN
190000 WRITELN('REPITE SI DESEAS INTENTAR LLEGAR A LA SOLUCION ',
190100 'OPTIMA O NO');
190200 READLN; READ(CAMBIO[1],CAMBIO[2]);
190300 END;
190400 IF CAMBIO[1]='N' THEN
190500 BEGIN
190600 WRITELN('DESEAS CONOCER ANTES LA SOLUCION OPTIMA?');
190700 READLN; READ(CAMBIO[1],CAMBIO[2]);
190800 WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
190900 ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
191000 BEGIN
191100 WRITELN('REPITE SI DESEAS CONOCER LA SOLUCION OPTIMA O NO');
191200 READLN; READ(CAMBIO[1],CAMBIO[2]);
191300 END;
191400 IF CAMBIO[1] = 'S' THEN
191500 BEGIN
191600 CONT1:=0;
191700 CONT3:=0;
191800 IF NSOLUCIONES[0] = 0 THEN
191900 WRITELN('UNA TAUTOLOGIA, ES DECIR, 1')

```

```

191901 (* *)
191902 (* *)
191903 (* *)
191904 (* *)
192000
192100     ELSE
192200     WHILE (CONT1<NSOLUCIONES[0]) DO
192300     BEGIN
192400         WHILE (CONT1 < NSOLUCIONES[0]) AND (CONT3 < 75) DO
192500         BEGIN
192600             WRITE(SOLUCIONES[0],CONT1);1);
192700             CONT1:=CONT1+1;
192800             CONT3:=CONT3+1;
192900         END;
193000         WRITELN;
193100         CONT3:=0;
193200     END;
193300     WRITELN;
193400     END;
193500     ELSE
193600     GOTO 3;
193700 END
193800 ELSE
193900 BEGIN
194000     CONT:=0;
194100     BANDERA:=FALSE;
194200     WHILE (CONT <= NSECUN) AND (NOT BANDERA) DO
194300     BEGIN
194400         IF MARCAMIN[CONT] = '*' THEN
194500             IF (CONT2-1)<=NSOLUCIONES[CONT] THEN
194600             BEGIN
194700                 WRITELN('PERFECTO, ORTUVIDE LA SOLUCION OPTIMA');
194800                 BANDERA:=TRUE;
194900             END;
195000             CONT:=CONT+1;
195100     END;
195200     IF NOT BANDERA THEN
195300     BEGIN
195400         WRITELN('TU SOLUCION ES CORRECTA PERO NO OPTIMA, DESEAS ',
195500             'LLEGAR A ELLA?');
195600         READLN; READ(CAMBIO[1],CAMBIO[2]);
195700         WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
195800             ((CAMBIO[1]='N') AND (CAMBIO[2] = 'O')))) DO
195900         BEGIN
196000             WRITELN('REPITE SI DESEAS INTENTAR LLEGAR A LA SOLUCION ',
196100                 'OPTIMA O NO?');
196200             READLN; READ(CAMBIO[1],CAMBIO[2]);
196300         END;
196400         IF CAMBIO[1] = 'N' THEN
196500         BEGIN
196600             WRITELN('DESEAS CONOCER LA(S) SOLUCION(ES) OPTIMA(S)?');
196700             READLN; READ(CAMBIO[1],CAMBIO[2]);
196800             WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
196900                 ((CAMBIO[1]='N') AND (CAMBIO[2]='O')))) DO
197000             BEGIN
197100                 WRITELN('REPITE SI DESEAS CONOCER LA SOLUCION OPTIMA');
197200                 READLN; READ(CAMBIO[1],CAMBIO[2]);
197300             END;
197400             IF CAMBIO[1]='S' THEN
197500             FOR CONT1:=0 TO NSECUN DO BEGIN
197600                 IF MARCAMIN[CONT1] = '*' THEN
197700                 FOR CONT2:=0 TO NSOLUCIONES[CONT1] DO
197800                     WRITE(SOLUCIONES[CONT1,COUNT2]);
197900                 WRITELN;
198000             END;
198100     END

```

```
198101 (* *)
198102 (* *)
198103 (* *)
198104 (* *)
198200      FLSE
198300      GOTO 3;
198400      END;
198500      END;
198600      END.
```

R KARNAUGH  
#RUNNING 4984

ESTE PROGRAMA TE GUIARA EN LA MINIMIZACION DE FUNCIONES POR MEDIO DEL METODO DE KARNAUGH.  
DESEAS CONOCER UNA PEQUEÑA INTRODUCCION A ESTE METODO?

#?

SI

ESTE ES UN METODO GRAFICO DE MINIMIZACION, SISTEMATICO ESTANDAR.  
ESTE PROGRAMA CONSIDERA LA FORMA DE SUMA DE PRODUCTO PARA EL MANEJO DE TUS MINTERMINOS.

EL PRIMER MAPA TE SERA PRESENTADO CON LAS VARIABLES QUE EL SISTEMA TE ASIGNA POR OMISION, PUEDES CAMBIARLAS, SI ASI LO DESEAS.

LA VARIABLE DE MAYOR PESO ES LA QUE SE PRESENTA MAS A LA IZQUIERDA, LA DE MENOR PESO, LA QUE SE ENCUENTRA MAS A LA DERECHA.

EL OBJETO DE RENOMBRAR LAS VARIABLES PUEDE SER:

1: USAR NOMBRES QUE MAS FRECUENTEMENTE USES

2: CAMBIAR EL PESO DE ELLAS.

EJEMPLO:

PARA 3 VARIABLES, UN MINTERMINO PUEDE SER:

C1B1A0

DONDE

C,B,A,: NOMBRES DE LAS VARIABLES SIENDO EL PESO A<D<C

1,1,0 : VALORES BINARIOS ASOCIADOS A LA VARIABLE QUE LE ANTECEDE,SU

EQUIVALENTE DECIMAL ES: 6

SI CAMBIAMOS DE POSICION LAS VARIABLES CON SU VALOR BINARIO ASOCIADO

POR EJEMPLO: A0C1B1

SU EQUIVALENTE DECIMAL ES: 011 = 3

CONTINUA...

ASI, UN MAPA QUE POR OMISION PRESENTA LA MAQUINA ES:

C\BA

-----  
| 0 | 1 | 1 | 3 | 2 |  
-----

| 4 | 5 | 7 | 6 |  
-----

UNA FORMA DE RENOMBRAR LAS VARIABLES SERIA:

A=B

B=C

C=A

QUE GENERARIA EL MAPA:

A\CB

-----  
| 0 | 2 | 6 | 4 |  
-----

| 1 | 3 | 7 | 5 |  
-----

COMO VERAS, DONDE ESTABA EL MINTERMINO 3 AHORA ESTA EL 6.  
CONTINUA...



PUEDES USAR MAPAS DESDE 2 HASTA 6 VARIABLES.

LA TEORIA DE LA MINIMIZACION A TRAVES DE MAPAS DE KARNAUGH TE DICE QUE DEBE HABER UN CUADRADO QUE CORRESPONDA A CADA COMBINACION DE ENTRADA; OTRO REQUISITO CONSISTE EN QUE LOS CUADRADOS DEBEN ESTAR ARREGLADOS EN TAL FORMA QUE, CUALQUIER PAR DE ELLOS INMEDIATAMENTE ADYACENTES ENTRE SI (HORIZONTAL O VERTICALMENTE) DEBEN CORRESPONDER A UN PAR DE CONDICIONES DE ENTRADA QUE SEAN LOGICAMENTE ADYACENTES, ES DECIR, QUE DIFIERAN EN 1 VARIABLE. POR EJEMPLO:

LOS CUADRADOS 2 Y 3 DEL MAPA ANTERIOR, CORRESPONDEN A LAS ENTRADAS  
COB1A0  
COB1A1

QUE SON IDENTICAS EXCEPTO EN A.

OBSERVA QUE LOS CUADRADOS DE LOS EXTREMOS DE LAS COLUMNAS O LAS HILERAS SON LOGICAMENTE ADYACENTES.

DE LO ANTERIOR:

CUALQUIER PAR DE MINTERMINOS DE N VARIABLES QUE SEAN ADYACENTES EN EL MAPA DE KARNAUGH, SE PUEDEN COMBINAR EN UN SOLO MINTERMINO PRODUCTO DE N-1 LITERALES, PUESTO QUE LOS CONJUNTOS DE DOS MINTERMINOS SE COMBINAN PARA ELIMINAR 1 VARIABLE, LOS CONJUNTOS DE 4 SE COMBINAN PARA ELIMINAR 2; LOS CONJUNTOS DE 8, PARA ELIMINAR 3, ETC.

AHORA BIEN CUANTAS VARIABLES DESEAS QUE TENGA TU FUNCION?

4

DC\ BA

1	0	1	1	3	2	1
1	4	5	7	6	1	
1	12	113	115	114	1	
1	8	9	111	110	1	

CONTINUA...

EL MAPA REPRESENTA LAS POSICIONES DE TODOS LOS MINTERMINOS POSIBLES PARA EL NUMERO DE VARIABLES QUE ELEGISTE

DESEAS RENOMBRAR LAS VARIABLES?

SI

SE TE DARA A CONTINUACION CADA VARIABLE, ENSEGUIDA DEBERAS ASIGNARLE EL NUEVO (UN CARACTER)

A=C

B=D

C=A

D=B

EL MAPA GENERADO CON TUS NUEVAS VARIABLES TIENE LA SIGUIENTE ESTRUCTURA  
BA\ DC

```
-----  
| 0 | 4 | 12 | 8 |  
-----  
| 1 | 5 | 13 | 9 |  
-----  
| 3 | 7 | 15 | 11 |  
-----  
| 2 | 6 | 14 | 10 |  
-----
```

CONTINUA...

TIENES LAS SIGUIENTES OPCIONES PARA PROPORCIONAR LOS MINTERMINOS

1: POR MEDIO DE CADENAS DE CARACTERES COMO A CONTINUACION SE DESCRIBEN  
POR CADA VARIABLE SE PROPORCIONAN DOS CARACTERES:  
EL PRIMER CARACTER INDICA EL NOMBRE DE LA VARIABLE, EL SEGUNDO  
CARACTER INDICA EL VALOR BINARIO ASOCIADO A ESTA VARIABLE.  
EJEMPLO:

\*\*\* PARA TRES VARIABLES DE SECUENCIA CBA, UN MINTERMINO PODRIA  
SER 011; ES DECIR, C=0, B=1, A=1, PARA INDICAR ESTO, SE DEBE  
PROPORCIONAR C0B1A1

2: POR MEDIO DE UNA CADENA DE CARACTERES BINARIOS, TENIENDO EN CUENTA  
QUE CADA DIGITO BINARIO ESTA ASOCIADO A UNA VARIABLE POR SU  
POSICION. EJEMPLO:

\*\*\* PARA TRES VARIABLES DE SECUENCIA BAC, SI SE PROPORCIONA 010,  
SIGNIFICA B=0, A=1, C=0

3: PROPORCIONANDO EL VALOR DECIMAL DE LOS MINTERMINOS. EJEMPLO:

\*\*\* PARA INDICAR EN UN MAPA DE CUATRO VARIABLES DE SECUENCIA  
BACD EL MINTERMINO B=0, A=0, C=1, D=1, ES DECIR, SE DEBE  
PROPORCIONAR EL EQUIVALENTE DECIMAL DE 0011, ESTO ES, 3

LOS MINTERMINOS DEBERAN SEPARARSE POR COMAS, TERMINAR CON ";" Y RETURN  
\* AHORA PROPORCIONA EL NUMERO DE LA OPCION ELEGIDA

3

PROPORCIONA LOS MINTERMINOS EN SU EQUIVALENTE DECIMAL SEPARADOS POR  
COMAS, PARA FINALIZAR TRANSMITE (;)

0,2,8,12,17;

UN NUMERO DE LOS PROPORCIONADOS, NO ESTA EN UN RANGO PERMISIBLE, VUELVE  
A DARLOS

0,2,8,12,13;

CON LAS POSICIONES DE LOS MINTERMINOS, EL MAPA DEFINIDO SE PRESENTA A LA  
DERECHA

BA\ DC

```
-----  
| 0 | 4 | 12 | 8 |  
-----  
| 1 | 5 | 13 | 9 |  
-----  
| 3 | 7 | 15 | 11 |  
-----  
| 2 | 6 | 14 | 10 |  
-----
```

BA\ DC

```
-----  
| 1 | 0 | 1 | 1 | 1 |  
-----  
| 0 | 0 | 1 | 1 | 0 |  
-----  
| 0 | 0 | 0 | 0 | 0 |  
-----  
| 1 | 0 | 0 | 0 | 0 |  
-----
```

CONTINUA...

DE ACUERDO AL MAPA ANTERIOR, PROPORCIONA LOS MINTERMINOS QUE SE PUEDEN COMBINAR, CADA UNO SEPARADO POR (,) Y TERMINAR CON (,)

EXISTEN MINTERMINOS QUE SE COMBINEN?

SI

PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA 0,2;

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

BA\ DC

BA\ DC

-----  
| 0 | 4 | 12 | 8 |

-----  
| 1X | 0 | 1 | 1 |

-----  
| 1 | 5 | 13 | 9 |

-----  
| 0 | 0 | 1 | 0 |

-----  
| 3 | 7 | 15 | 11 |

-----  
| 0 | 0 | 0 | 0 |

-----  
| 2 | 6 | 14 | 10 |

-----  
| 1X | 0 | 0 | 0 |

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINEN?

SI

PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA 8,9;

UNO DE LOS NUMEROS QUE PROPORCIONASTE NO ES MINTERMINO

PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA 8,12;

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

BA\ DC

BA\ DC

-----  
| 0 | 4 | 12 | 8 |

-----  
| 1X | 0 | 1X | 1X |

-----  
| 1 | 5 | 13 | 9 |

-----  
| 0 | 0 | 1 | 0 |

-----  
| 3 | 7 | 15 | 11 |

-----  
| 0 | 0 | 0 | 0 |

-----  
| 2 | 6 | 14 | 10 |

-----  
| 1X | 0 | 0 | 0 |

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINEN?

SI

PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA 12,13;

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

BA\ DC

BA\ DC

-----  
| 0 | 4 | 12 | 8 |

-----  
| 1X | 0 | 1X | 1X |

-----  
| 1 | 5 | 13 | 9 |

-----  
| 0 | 0 | 1X | 0 |

-----  
| 3 | 7 | 15 | 11 |

-----  
| 0 | 0 | 0 | 0 |

-----  
| 2 | 6 | 14 | 10 |

-----  
| 1X | 0 | 0 | 0 |

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINEN?

SI

TODOS LOS MINTERMINOS YA ESTAN CUBIERTOS, POR LO CUAL TU SOLUCION ES:  
D0C0A0+D1B0A0+D1C1B0

PERFECTO, OBTUVISTE LA SOLUCION OPTIMA

#ET=9:13.0 PT=2.5 IO=0.2

FILE	8(ACCESS='SEQUENTIAL',BLOCKSIZE=192),MYUSE='IO',	00000100
	.FORM='FORMATTED',RECL=192),KIND='REMOTE',STATUS='OLD')	00000200
FILE	9(ACCESS='SEQUENTIAL',BLOCKSIZE=84),MYUSE='OUT',	00000300
	.FORM='FORMATTED',RECL=84,KIND='PRINTER',STATUS='NEW')	00000400
	DIMENSION IX(960),IMAT(8,8),M(64),VD(64),IDIR(64),IO(64)	00000500
	,BJF(640),IDD(64),LIM(10),LIN(10),DIR(15),IVD(1000),TAB(40),	00000600
	.MAT2(70),VECI(64),IP(2)),MAT(10,64)	00000610
	COMMON CONTRL(28),CURSOR(8)	00000620
	COMMON /C/ IMPZ,ICH	00000625
	CHARACTER CONTRL*1,CURSOR*1	00000630
	CHARACTER IX*1,B*3,M*6,VD*1,BUF*1,TAB*14,MAT2*30,CONTIN*8	00000700
	,MAT*14	00000705
	INTEGER DIR,VECI	00000709
	DATA IPRV,IIP/0,)/	00000710
	DATA CONYRL/Z00,Z03,Z11,Z05,Z08,Z16,Z3C,Z0C,Z0D,Z25,Z13,	00000711
	Z0E,Z0F,Z18,Z19,Z3F,Z1E,Z1C,Z1D,Z1F,Z1E,Z1Z,	00000712
	Z27,Z05,Z06,ZE6,ZE7,Z7F/	00000713
	DATA CURSOR/Z40,Z4F,Z7F,Z7B,Z58,Z6C,Z50,Z7D,Z4D,Z5D,Z5C,	00000714
	Z4E,Z68,Z6J,Z4B,Z61,ZF0,ZF1,ZF2,ZF3,ZF4,ZF5,	00000715
	ZF6,ZF7,ZF8,ZF9,Z7A,Z5E,Z4C,Z7E,Z6E,Z6F,Z7C,	00000716
	ZC1,ZC2,ZC3,ZC4,ZC5,ZC6,ZC7,ZC8,ZC9,ZD1,ZD2,	00000717
	ZD3,ZD4,ZD5,ZD6,ZD7,ZD8,ZD9,ZE2,ZE3,ZE4,ZE5,	00000718
	ZE6,ZE7,ZE8,ZE9,Z4A,ZF0,Z5A,Z5F,Z6D,Z79,Z81,	00000719
	Z82,Z83,Z84,Z85,Z86,Z87,Z88,Z89,Z91,Z92,Z93,	00000720
	Z94,Z95,Z96/	00000721
	DATA B(1:6)/'ABCDEF'/	00000800
	WRITE(8,3)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00000810
3	FORHAT(4A1,'IMPRESSION ? 51=1')	00000820
	READ(8,4)IMPZ	00000830
4	FORHAT(I1)	00000840
	CALL NUMVAR(N)	00000900
	CALL RENCOL(N,NREN,MCOL)	00001000
	INDOP=0	00001100
	CALL INIHAT(NREN,MCOL,IX,VD,INDOP)	00001200
	IBAN=0	00001300
	CALL PREHAT(N,IBAN,B,IX)	00001400
	CALL SEQHAT(NREN,MCOL,IMAT,IDIR)	00001500
	CALL NOMIGL(IB)	00001600
	IF(IB.NE.1)GO TO 150	00001700
	CALL CAMVAR(N,B)	00001800
	IBAN=1	00001900
	CALL PREHAT(N,IBAN,B,IX)	00002000
	CALL TRANSF(N,B,NREN,MCOL,IMAT,IDIR)	00002100
	WRITE(8,5)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00002110
5	FORHAT(4A1,'CONTINUA')	00002115
	READ(8,6)CONTIN	00002120
6	FORHAT(1A8)	00002125
150	CONTINUE	00002200
	CALL DECHAP(VD,NREN,MCOL,BUF,IMAT,KBAN)	00002300
	CALL SELENT(IX,IFORM)	00002400
	GO TO (10,30,20),IFORM	00002500
10	CALL ALFABI(N,B,M,IFI)	00002600
	CALL CONDEC(N,M,IFI,ID)	00002700
	GO TO 40	00002800
20	JBAN=0	00003000
	CALL DECIM(N,ID,IFI,BUF,JBAN,IPRV,IB)	00003100
	IPRV=0	00003110
	CALL COMBI(N,IFI,ID,M)	00003200
	GO TO 40	00003300
30	CALL BINARI(N,M,IFI)	00003400
	CALL CONDEC(N,M,IFI,ID)	00003500
40	CALL INTELE(ID,IFI,IDIR,VD)	00003600
	INDOP=1	00003700
	CALL INIHAT(NREN,MCOL,IX,VD,INDOP)	00003800

IBAN=3	00003900
CALL PREMAT(N, IBAN, B, IX)	00004000
JBAN=1	00004200
KBAN=1	00004300
CALL DECHAP(VD, NREN, HCOL, BUF, IMAT, KBAN)	00004400
WRITE(8, 48) CONTRL(23), CONTRL(28), CURSOR(1), CURSOR(19)	00004405
48 FORMAT(5X, 4A1, '**** EN PROCESO ****')	00004409
CALL SOLUC(N, IFI, H, ID, LIM, LIN, DIR, IVD, TAB, III, MAT2, NOP, NIS	00004410
., MAT)	00004415
WRITE(8, 49) CONTRL(23), CONTRL(20), CURSOR(1), CURSOR(19)	00004420
49 FORMAT(5X, 4A1, '++++ FIN DE PROCESO +++++')	00004440
NIP=1	00004450
50 CALL DECIM(N, IDD, IFII, BJF, JBAN, IPRV, IB)	00004600
IF( (IB.EQ.1) GO TO 60	00004700
CALL VALIDA(IFII, IFI, IDD, ID, N, LIM, LIM, IVD, VECL, NIP, IP, IIP, MAT)	00004800
IF(IMPZ.EQ.1) CALL IMPR	00004810
GO TO 50	00004900
63 CALL DIAG(VECL, NIP, NOP, TAB, III, DIR, MAT2, IFI, IP, NIS, N)	00005000
END	00005100
*****	00005110
SUBROUTINE NUMVAR(N)	00005200
COMMON CONTRL(28), CURSOR(80)	00005210
COMMON /C/ IMPZ, ICH	00005230
CHARACTER CONTRL*1, CJRSOR*1	00005240
WRITE(8, 5) CONTRL(8)	00005250
5 FORMAT(A1)	00005260
6 WRITE(8, 8) CONTRL(23), CONTRL(28), CURSOR(1), CURSOR(19)	00005270
., CONTRL(23), CONTRL(28), CURSOR(1), CURSOR(1), CONTRL(19), CONTRL(21),	00005280
., CONTRL(23), CONTRL(26)	00005290
8 FORMAT(4A1, 'CUANTAS VARIABLES TIENE LA FUNCION (2 <= N <= 6)'	00005300
I, 4A1, ' DATOS: ', A1, 1X, 3A1)	00005310
READ(8, 9, ERR=1) N	00005320
9 FORMAT(I1)	00005330
IF( (N.LE.2 .OR. N.GT.6) GO TO 14	00005340
IF(IMPZ.EQ.1) CALL IMPR	00005350
RETURN	00005360
10 WRITE(8, 12) CONTRL(23), CONTRL(28), CURSOR(1), CURSOR(21)	00005370
12 FORMAT(4A1, 'ERROR: TIPO DE DATO ERRONEO; N => NO NUMERICO')	00005380
IF(IMPZ.EQ.1) CALL IMPR	00005390
CALL LIMPIA	00005400
GO TO 6	00005410
14 WRITE(8, 16) CONTRL(23), CONTRL(28), CURSOR(1), CURSOR(21)	00005420
16 FORMAT(4A1, 'ERROR: VALOR FUERA DE RANGO; RECUERDA 2 <= N <= 6)'	00005430
IF(IMPZ.EQ.1) CALL IMPR	00005440
CALL LIMPIA	00005450
GO TO 6	00005460
END	00005470
*****	00005475
SUBROUTINE NOHIGL(IB)	00006300
COMMON /C/ IMPZ, ICH	00006310
COMMON CONTRL(28), CURSOR(80)	00006320
CHARACTER CONTRL*1, CJRSOR*1	00006330
CHARACTER RESP*80, SPA*1	00006340
DATA SPA/' '	00006350
IB=0	00006360
8 CALL LIMPIA	00006370
WRITE(8, 140) CONTRL(23), CONTRL(28), CURSOR(1), CURSOR(19)	00006800
., CONTRL(23), CONTRL(28), CURSOR(1), CJRSOR(1)	00006810
140 FORMAT(1X, 4A1, 'DESEAS CAMBIAR EL NOMBRE O UBICACION DE LAS VARIABLE	00006820
1ES (SI/NO) ', 4A1, ' DATOS: '	00007100
READ(8, 110) RESP	00007600
110 FORMAT(A80)	00007700
IF(IMPZ.EQ.1) CALL IMPR	00007910
DO 120 I=7, 80	00008000

```

IF(RESP(I)=1).EQ.SPA)GO TO 121
IF(.NOT.(RESP(I+1).EQ.*SI*))GO TO 112
I8=1
GO TO 121
112 IF(.NOT.(RESP(I+1).EQ.*NO*))GO TO 8
I8=0
GO TO 121
120 CONTINUE
GO TO 8
121 RETURN
END
*****
SUBROUTINE RENCOL(N,NREN,MCOL)
R=FLOAT(N)/2.
M=R
MF=(R-H)*10.
NREN=2.**M
IF(MF.EQ.0)MCOL=2.**M
IF(MF.NE.0)MCOL=2.**(M+1)
RETURN
END
*****
SUBROUTINE INIMAT(NREN,MCOL,IX,VD,INDOP)
DIMENSION IX(960),IND(8),VD(64)
CHARACTER IX*1,VD*1
DATA (IND(I),I=1,8)/197,277,357,437,517,597,677,757/
IF(II.EQ.1)GO TO 30
DO 10 IH=1,960
10 IX(IH)=* '
II=0
DO 20 IJ=1,NREN
DO 20 IK=1,MCOL
IF=IJ+1
20 VD(IF)=* '
30 CONTINUE
IVD=0
DO 60 IU=1,NREN
I1=IND(IU)
DO 55 JJ=1,MCOL
IVD=IVD+1
IX(I1)=VD(IVD)
55 I1=I1+2
60 CONTINUE
RETURN
END
*****
SUBROUTINE PREMAT(N,IOAN,8,IX)
DIMENSION IX(960),AV(63),IV(63),JP(5),JF(5)
COMMON /C/ IHPZ,ICH
COMMON CONTRL(28),CJRSOR(83)
CHARACTER CONTRL*1,CURSJR*1
CHARACTER IX*1,AV*1,D*3)
DATA (IV(I),I=1,55)/119,275,121,123,275,359,361,121
1,123,285,355,365,435,519,521,45,47,49,51,121,123,129
2,131,293,355,373,435,519,521,527,529,45,47,49,51,121,123
3,129,131,293,353,373,433,515,595,613,673,675,693,753
4,755,839,841,847,849/
DATA (AV(I),I=1,55)/'D','A','B','A','C','C','C','C'
1,'B','A','B','A','D','D','C','C','C','C','D','D','D','D'
2,'B','A','B','A','E','E','E','E','D','D','D','D','E','E'
3,'E','E','C','B','A','A','C','B','A','C','B','A'
4,'F','F','F','F'/
DATA (JP(I),I=1,5)/1,3,8,16,32/
DATA (JF(I),I=1,5)/2,5,8,16,24/
00008100
00008200
00008300
00008400
00008500
00008600
00008700
00008800
00008900
00009000
00009100
00009200
00009300
00009400
00009500
00009600
00009700
00009800
00009900
00010000
00010100
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00011000
00011100
00011200
00011300
00011400
00011500
00011600
00011700
00011800
00011900
00012000
00012100
00012200
00012300
00012310
00012400
00012500
00012504
00012505
00012506
00012600
00012700
00012800
00012900
00013000
00013100
00013200
00013300
00013400
00013500
00013600
00013700
00013800

```

```

      IP=JP(N-1)
      IPP=JP(N-1)+JF(N-1)-1
      IF(IBAN.EQ.3)GO TO 100)
      IF(IBAN.EQ.1)GO TO 800
      DO 700 IU=IP,IPP
      IO=IV(IU)
      IX(ID)=AV(IU)
700 CONTINUE
      WRITE(8,11)CONTRL(8)
      WRITE(8,12)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19),N
12 FORMAT(4A1,'ESTE ES EL MAPA DE KARNAUGH PARA ',I1,' VARIABLES')
      GO TO 1010
800 NT=N*2
      DO 900 IU=IP,IPP
      IO=IV(IU)
      DO 850 IK=1,NT,2
      IF(AV(IU).NE.B(İK:İK))GO TO 850
      IX(IO)=B(İK+1:İK+1)
      GO TO 860
850 CONTINUE
860 CONTINUE
900 CONTINUE
      WRITE(8,13)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
13 FORMAT(4A1,'ESTE ES EL NUEVO NOMBRE O UBICACION DE LAS VARIABLES')
      GO TO 1010
1000 CALL LIMPIA
      WRITE(8,25)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
25 FORMAT(1X,4A1,'ESTOS SON LOS MINTERMINOS QUE PROPORCIONASTE')
1010 WRITE(8,10)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(2),
      .(IX(I),I=1,960)
10 FORMAT(4A1,6(16DA1))
11 FORMAT(A1)
      IF(IMPZ.EQ.1)CALL IMPZ
      RETURN
      END
*****
SUBROUTINE CAHVAR(N,B)
DIMENSION DEF(6)
COMMON /C/ IMPZ,ICH
COMMON CONTRL(28),CURSOR(80)
CHARACTER CONTRL=1,CURSOR=1
CHARACTER A=80 , B=3) , SPA=1 , IGL=1 , COM=1 , FIN=1
CHARACTER DEF=1
DATA SPA,IGL,COM,FIN/ ' ',' ',' ',' ',' ',' ',' ',' /
DATA (DEF(I),I=1,6)/'A','B','C','D','E','F'/
5 CALL LIMPIA
      WRITE(8,100)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
      ,CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)
100 FORMAT(1X,4A1,'PROPORCIONA EL NOMBRE DE LAS VARIABLES QUE SUTITUYE
IN',/,'IX',A LAS REPRESENTADAS EN EL MAPA , DE LA SIGUIENTE FORMA '
2/,1X,'A=NOMBRE-VARIABLE , B=NOMBRE-VARIABLE... ; ',4A1,'DATOS:')
      READ(8,110)A
110 FORMAT(A80)
      IF(IMPZ.EQ.1)CALL IMPZ
      CALL LIMPIA
      K=1
      J=1
      IBAN=0
      DO 50 I=7,80
      IF(A(I:1).EQ.SPA)GO TO 30)
      IF(J.NE.2)GO TO 10
      J=J+1
      IF(A(I:1).EQ.IGL)GO TO 30)
      WRITE(8,400)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)

```

```

00013900
00014000
00014100
00014200
00014300
00014400
00014500
00014600
00014610
00014620
00014630
00014700
00014800
00014900
00015000
00015100
00015200
00015300
00015400
00015500
00015600
00015700
00015710
00015711
00015712
00015715
00015720
00015725
00015800
00015810
00015900
00015910
00015915
00016000
00016800
00016810
00016900
00017000
00017005
00017010
00017015
00017100
00017200
00017300
00017400
00017410
00017500
00017510
00017700
00017800
00017900
00018000
00018300
00018305
00018310
00018400
00018500
00018510
00018600
00018700
00018800
00018900
00019000
00019100

```



```

400 FORMAT(IX,4A1,'*** ERROR: SE OMITIO UN SIGNO IGUAL ***')
      IBAN=1
      GO TO 60
10 IF(J.NE.4)GO TO 20
      J=1
      IF(A(I:1).EQ.FIN)GO TO 70
      IF(A(I:1).EQ.COM)GO TO 50
      WRITE(8,41)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
410 FORMAT(IX,4A1,'*** ERROR: SE OMITIO UNA COMA ***')
      IBAN=1
      GO TO 60
20 IF(A(I:1).GE.'A'.AND.A(I:1).LE.'Z')GO TO 25
      WRITE(8,42)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
420 FORMAT(IX,4A1,'*** ERROR: UNO DE LOS CARACTERES NO ES
1,' ALFABETICO ***')
      IBAN=1
      GO TO 60
25 B(K:K)=A(I:1)
      K=K+1
      J=J+1
50 CONTINUE
      WRITE(8,43)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
431 FORMAT(IX,4A1,'*** ERROR: LA CADEMA NO TIENE FIN DE MENSAJE ***')
      IBAN=0
60 IF(IMPZ.EQ.1)CALL INPR
      IF(IBAN.EQ.1) GO TO 5
70 IF(N.EQ.(K-1)/2) GO TO 72
      WRITE(8,44)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
440 FORMAT(IX,4A1,'*** ERROR: EL NUMERO DE CAMBIOS NO ES IGUAL AL
1,' NUMERO DE VARIABLES ***')
      GO TO 5
72 NP=3
      N2=K-1
      DO 154 IG=1,2
      NPP=NP
      DO 75 I=IG,N2,2
      DO 74 J=NPP,N2,2
74 IF(B(I:1).EQ.B(J:J))GO TO 80
      NPP=NPP+2
75 CONTINUE
154 NP=NP+1
      NL=K-1
      DO 155 I=1,NL,2
      IF(B(I:1).GE.DEF(1).AND.B(I:1).LE.DEF(N))GO TO 155
      WRITE(8,465)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
465 FORMAT(IX,4A1,'*** ERROR: UNA VARIABLE A LA QUE SE LE
1,' QUIERE CAMBIAR EL NOMBRE NO EXISTE **')
      IF(IMPZ.EQ.1)CALL INPR
      GO TO 5
155 CONTINUE
      RETURN
80 WRITE(8,455)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
455 FORMAT(IX,4A1,'*** ERROR: UNA DE LAS LETRAS ESTA REPETIDA ***')
      IF(IMPZ.EQ.1)CALL INPR
      GO TO 5
      END
*****
SUBROUTINE SEOMAT(NREN,NCOL,IMAT,IDIR)
DIMENSION IMAT(0,8),IDIR(64)
NUM=0
DO 10 I=1,NREN
DO 10 J=1,NCOL
IMAT(I,J)=NUM
10 NUM=NUM+1

```

K=3	00026000
15 IF(K.GT.MCOL)GO TO 25	00026100
DD 20 J=1,NREN	00026200
IT=IMAT(J,K)	00026300
IMAT(J,K)=IMAT(J,K+1)	00026400
20 IMAT(J,K+1)=IT	00026500
K=K+4	00026600
GO TO 15	00026700
25 CONTINUE	00026800
K=3	00026900
28 IF(K.GT.NREN)GO TO 35	00027000
DD 30 J=1,MCOL	00027100
IT=IMAT(K,J)	00027200
IMAT(K,J)=IMAT(K+1,J)	00027300
IMAT(K+1,J)=IT	00027400
30 CONTINUE	00027500
K=K+4	00027600
GO TO 28	00027700
35 CONTINUE	00027800
IC=0	00027900
DD 40 I=1,NREN	00028000
DD 40 J=1,MCOL	00028100
IC=IC+1	00028200
ILK=IMAT(I,J)+1	00028300
40 IDIR(ILK)=IC	00028400
RETURN	00028900
END	00029000
*****	
SUBROUTINE TRANSF(N,B,NREN,MCOL,IMAT,IDIR)	00029100
DIMENSION IMAT(8,8),IEXPP(6),ID(6),MB(8,8),IDIR(64)	00029200
CHARACTER B*30,B1*30,MB*6,UND*1,CERO*1,TI*1,TII*2	00029300
DATA UND,CERO /'1',')/'	00029400
NS=N*2-2	00029500
3 IBA=0	00029600
DD 6 I=1,NS+2	00029700
IF(B(I:1).LE.B(I+2:I+2))GO TO 6	00029800
TII=B(I:I+1)	00029900
B(I:I+1)=B(I+2:I+3)	00030000
B(I+2:I+3)=TII	00030100
IBA=1	00030200
6 CONTINUE	00030300
IF(IBA.EQ.1)GO TO 3	00030400
J=1	00030500
DD 10 I=1,N	00030600
J=J+2	00030700
10 B(I:I)=B(J:J)	00030800
DD 16 I=1,N	00030900
16 B1(I:I)=B(I:I)	00031000
NS=N-1	00031100
20 IBA=0	00031200
DD 30 J=1,NS	00031300
IF(B1(J:J).LE.B1(J+1:J+1))GO TO 30	00031400
TI=B1(J:J)	00031500
B1(J:J)=B1(J+1:J+1)	00031600
B1(J+1:J+1)=TI	00031700
IBA=1	00031800
30 CONTINUE	00031900
IF(IBA.EQ.1)GO TO 20	00032000
DD 40 J=1,N	00032100
IEXPP(J)=2**NS	00032200
40 NS=NS-1	00032300
DD 50 K=1,N	00032400
DD 50 L=1,N	00032500
50 IF(B(K:K).EQ.B1(L:L))ID(K)=IEXPP(L)	00032600

```

DO 100 I=1,NREN
DO 100 J=1,NCOL
KB=N
IR=IHAT(I,J)
IF(IR.EQ.0)GO TO 54
52 IF(IR.EQ.1)GO TO 58
RR=FLOAT(IR)/2.
IR=RR
FR=RR-IR
IF(FR.EQ.0.0)HB(I,J)(KB:KB)=CERO
IF(FR.NE.0.0)HB(I,J)(KB:KB)=UNO
KB=KB+1
GO TO 52
54 IF(KB.EQ.0)GO TO 7)
HB(I,J)(KB:KB)=CERO
KB=KB+1
GO TO 54
58 HB(I,J)(KB:KB)=UNO
KB=KB+1
GO TO 54
70 CONTINUE
100 CONTINUE
DO 200 I=1,NREN
DO 200 J=1,NCOL
KAL=0
DO 300 K=1,N
IF(HB(I,J)(K:K).EQ.CERO)GO TO 300
KAL=KAL+ID(K)
300 CONTINUE
IHAT(I,J)=KAL
200 CONTINUE
IC=0
DO 230 I=1,NREN
DO 230 J=1,NCOL
IC=IC+1
ILK=IHAT(I,J)+1
230 IJIR(ILK)=IC
RETURN
END
SUBROUTINE SELENT(IX,IFORM)
COMMON CNTRL(28),CURSOR(80)
COMMON /C/ IHPZ,ICH
CHARACTER CNTRL+1,CURSOR+1
CHARACTER BUF1+240,BUF2+240,BUF3+240,BUF4+240,IX(960)+1
*BUF5+240,BUF6+240,BUF7+240
BUF1(1:80)=' AHORA PROPORCIONA TUS MINTERMINOS DE ALGUNA DE LAS SIGUE'
*BUENTES FORMAS:
BUF1(81:160)='
*BUF1(161:240)='1.' POR MEDIO DE UNA CADENA DE CARACTERES COMO A CONTINUA
*CONTINUACION SE DESCRIBE:
BUF2(1:8)='
*BUF2(81:16)=' EL PRIMER CARACTER INDICA EL NOMBRE DE LA VARIABLE.
*ASOCIADO A ESTA VARIABLE.
*UN MINTERMINO PODRIA SER
*AR ESTO SE DEBE PROPORCIONAR
*BUF3(1:16)='
*BUF3(161:240)='
*BUF4(1:80)='

```

```

00032700
00032800
00032900
00033000
00033100
00033200
00033300
00033400
00033500
00033600
00033700
00033800
00033900
00034000
00034100
00034200
00034300
00034400
00034500
00034600
00034700
00034800
00034900
00035000
00035100
00035200
00035300
00035400
00035500
00035600
00035700
00035800
00035900
00036000
00036100
00036200
00036300
00036400
00036500
00036600
00036700
00036800
00036900
00037000
00037005
00037006
00037007
00037010
00037011
00037012
00037013
00037014
00037015
00037016
00037017
00037018
00037019
00037020
00037021
00037022
00037023
00037024
00037025
00037026
00037027
00037028
00037029
00037030
00037031
00037032
00037033
00037034
00037035
00037036
00037037
00037038
00037039
00037040
00037041
00037042
00037043
00037044
00037045
00037046
00037047
00037048
00037049
00037050
00037051
00037052
00037053
00037054
00037055
00037056
00037057
00037058
00037059
00037060
00037061
00037062
00037063
00037064
00037065
00037066
00037067
00037068
00037069
00037070
00037071
00037072
00037073
00037074
00037075
00037076
00037077
00037078
00037079
00037080
00037081
00037082
00037083
00037084
00037085
00037086
00037087
00037088
00037089
00037090
00037091
00037092
00037093
00037094
00037095
00037096
00037097
00037098
00037099
00038000
00038010
00038020
00038022

```

```

00038224
BUF4(81:160)= *2.- POR MEDIO DE UNA CADENA DE CARACTERES BINARIOS, 00038300
* TENIENDO EN CUENTA QUE 00038400
* BUF4(161:240)= * CADA DIGITO BINARIO ESTA ASOCIADO A UNA VARIABLE 00038500
* E POR SU POSICION 00038510
* BUF5(1:80)= *... EJEMPLO: PARA TRES VARIABLES DE SECUENCIA BAC, 00038600
* SI SE PROPORCIONA 01) 00038700
* BUF5(81:160)= * SIGNIFICA B=0 A=1 C=0 00038710
00038720
* BUF5(161:240)= * 00038730
00038740
BUF6(1:80)= *3.- PROPORCIONANDO EL VALOR DECIMAL DE LOS HINTERMINOS 00038800
* NOS 00038810
* BUF6(81:160)= *... EJEMPLO: PARA INDICAR EN UN MAPA DE CUATRO VAR 00038900
* IABLES DE SECUENCIA BACD 00038910
* BUF6(161:240)= * EL HINTERMINO B=0 A=0 C=1 D=1 ; ES DEC 00039000
* IR 0011, SE DEBE PROPORCIONAR * 00039010
* BUF7(1:80)= * EL EQUIVALENTE DECIMAL DE 0011, ESTO E 00039100
* S 3. 00039200
* BUF7(81:160)= * 00039300
00039310
DO 50 I=1,240 00039610
IX(I)=BUF1(I:I) 00039615
IX(I+240)=BUF2(I:I) 00039616
IX(I+480)=BUF3(I:I) 00039617
50 IX(I+720)=BUF4(I:I) 00039618
WRITE(8,81)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(2), 00039620
*(IX(I),I=1,960) 00039625
DO 60 I=1,240 00039630
IX(I)=BUF5(I:I) 00039631
IX(I+240)=BUF6(I:I) 00039645
60 IX(I+480)=BUF7(I:I) 00039650
WRITE(8,80)(IX(I),I=1,640) 00039651
15 WRITE(8,16)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1), 00039655
CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(22), 00039656
CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(22), 00039657
CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1) 00039658
16 FORMAT(4A1, * 00039659
*4A1, * 00039660
*4A1, *QUE FORMA DE PROPORCIONAR LOS HINTERMINOS DESEAS ? 1=2*3 * 00039661
* 4A1, *INPUT: * 00039668
80 FORMAT(6(160A1)) 00039670
81 FORMAT(4A1,6(160A1)) 00039675
READ(8,20)NF 00039700
21 FORMAT(1X, *NF=*, I1) 00039900
20 FORMAT(6X, I1) 00040000
IF(IMPZ.EQ.1)CALL IMPR 00040010
IF(NF.LT.1.OR.NF.GT.3)GO TO 15 00040010
IFORN=NF 00040200
DO 30 IT=1,960 00040210
30 IX(IT)= * 00040220
RETURN 00040300
END 00040400
***** 00040410
SUBROUTINE ALFABI(N,8,M,IFI) 00040500
DIMENSION IREP(6),H(64),H1(4) 00040600
COMMON CONTRL(28),CURSOR(80) 00040610
COMMON /C/ IMPZ,ICH 00040612
CHARACTER CONTRL*1,CURSOR*1 00040615
CHARACTER M*6,M1*6,B*5,COM*1,FIN*1,SPA*1,A*80,UNO*1,CERO*1 00040700
DATA SPA,COM,FIN,JNO,CERO/' ',' ',' ',' ',' ',' ',' ',' ','1','0'/ 00040800
DATA (H1(IL),IL=1,40)/4) * // 00040900
NM=1 00041000
WRITE(8,5)CONTRL(8) 00041005

```

5	FORMAT(1A1)	00041007
10	CALL LIMPIA	00041010
	L=1	00041100
	MN1=1	00041200
	NC=0	00041300
	0J 11 IL=1,6	00041400
11	IREP(IL)=0	00041500
	WRITE(8,299)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19),	00041600
	CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00041610
299	FORMAT(1X,4A1,'*PROPORCIONA LOS MINTERMINOS SEPARADOS POR COMAS',/,	00041800
	1X,'*PARA INDICAR FIN DE ENTRADA TRANSMITE UN ',/,4A1,'*DATOS:')	00041900
	READ(8,250)A	00042000
250	FORMAT(AB0)	00042100
	IF(IMPZ.EQ.1)CALL IMPR	00042305
	CALL LIMPIA	00042310
	DO 50 I=7,80	00042400
	IF(A(I:1).EQ.SPA)GO TO 48	00042500
	IF(A(I:1).EQ.COM.OR.A(I:1).EQ.FIN)GO TO 35	00042600
	IF(L.EQ.1)GO TO 16	00042700
	IF(A(I:1).EQ.CERO.OR.A(I:1).EQ.UNO)GO TO 15	00042800
	WRITE(8,393)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00042900
393	FORMAT(1X,4A1,'*** ERROR: A UNA DE LAS VARIABLES NO SE LE ESTA ASI	00043100
	*GNANDO UN VALOR BINARIO')	00043200
	GO TO 80	00043300
15	NC=NC+1	00043400
	M1(MN1)(K:K)=A(I:1)	00043500
	L=1	00043600
	IF(NC.GT.6)GO TO 39	00043700
	GO TO 50	00043800
16	0J 20 J=1,N	00044100
	IF(A(I:1).NE.B(J:J))GO TO 20	00044200
	L=0	00044300
	K=J	00044400
	IF(IREP(J).EQ.1)GO TO 22	00044500
	IREP(J)=1	00044600
	GO TO 50	00044700
22	CONTINUE	00044800
	WRITE(8,310)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00044900
310	FORMAT(1X,4A1,'*** ERROR: UN NUMBRE DE VARIABLE NO EXISTE O NO ES	00045100
	*ALFABETICO')	00045200
	GO TO 80	00045300
22	WRITE(8,312)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00045400
312	FORMAT(1X,4A1,'*** ERROR: UN NUMBRE DE VARIABLE ESTA REPETIDO')	00045600
	GO TO 80	00045700
35	IF(NC.EQ.0.AND.A(I:1).EQ.COM)GO TO 39	00045800
	IF(NC.EQ.0.AND.A(I:1).EQ.FIN)GO TO 60	00045900
	IF(NC.NE.N)GO TO 39	00046000
	0J 38 IL=1,N	00046100
38	IREP(IL)=0	00046200
	NC=0	00046300
	MN1=MN1+1	00046400
	IF(A(I:1).EQ.FIN)GO TO 6)	00046500
	GO TO 50	00046600
39	WRITE(8,314)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00046700
314	FORMAT(1X,4A1,'*** ERROR: EL NUMERO DE VARIABLES DEL MINTERMINO NO	00046900
	*ES IGUAL AL',/,1X,'*NUMERO DE VARIABLES DEL MAPA',/,1X,'*D DOS COMAS	00047000
	*S CONTINUAS, O NO INFORMACION Y COMA')	00047100
	GO TO 80	00047200
40	IF(L.EQ.59.AND.NC.NE.3)GO TO 35	00047300
59	CONTINJE	00047400
60	IF(MN1.EQ.1)GO TO 55	00047500
	0J 54 IL=1,MN1-1	00047600
	M(NH)=M1(IL)	00047700
54	NN=NN+1	00047800

```

55 IF(1.GT.80)GO TO 10          00047900
   IFI=NM-1                    00048000
   IF(NM.EQ.1)GO TO 75        00048100
75 RETURN                       00048500
80 CALL IMPR                   00048520
   GO TO 10                    00048600
350 FORMAT(1X,6A6)            00048700
   END                          00048710
*****
SUB ROUTINE DECIM(N,IDD,IFI,BUF,JBAN,IPRV,IB) 00048800
DIMENSION IDD(64),B(4),IHAT(8,8),BUF(64)    00048900
COMMON CONTRL(28),CURSOR(8)                00048925
COMMON /C/ IMPZ,ICH                        00048950
CHARACTER CONTRL*1,CJRSOR*1                00048970
CHARACTER A*8),FIN*1,COM*1,SPA*1,B*2,BF*2,BUF*1 00049000
DATA SPA,COM,FIN/' ',' ',' ',' ',' '/      00049100
NS=2.0**N                                   00049110
IFI=0                                         00049200
IF(1=0)                                       00049300
IF(IPRV.EQ.0)WRITE(8,4)CONTRL(8)            00049310
4  FORMAT(4A1)                               00049315
   D3 5 I=1,64                               00049400
5  IDD(I)=0.                                  00049500
6  CALL LIMPIA                               00049510
   IF(JBAN.EQ.0)GO TO 10                    00049600
   WRITE(8,31)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19) 00049700
   .      .CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(20)    00049715
   .      .CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(21)    00049730
   .      .CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)      00049710
310 FORMAT(1X,4A1,'PROPORCIONA TU REDUCCION, HACIENDO REFERENCIA A LOS 00049900
   . MINTERMINOS POR SJ',4A1,'EQUIVALENTE DECIMAL, SEPARADOS POR COMAS 00050000
   . Y PARA TERMINAR EL CARACTER ( )',4A1,'-PARA TERMINAR DE PROPORCIO 00050100
   . NAR REDUCCIONES DIGITA LA PALABRA => FIN',4A1,'DATOS:') 00051100
   GO TO 17                                  00050200
10  WRITE(8,30)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19) 00050300
   . CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)            00050310
17  IF(IPRV.EQ.1)GO TO 2                    00050410
   IPRV=1                                       00050420
   WRITE(8,14)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(2), 00050500
   . (B(I),I=1,64)                                       00050515
14  FORMAT(4A1,4(160A1))                    00050600
   WRITE(8,4)CONTRL(23),CONTRL(28),CURSOR(7),CURSOR(1) 00050610
300  FORMAT(1X,4A1,' PRUPORCIONA LOS MINTERMINOS EN SU EQUIVALENTE DECI 00051100
   . HAL',/,', SEPARADOS POR COMAS , PARA FIN DE ENTRADAS UN ',4A1,'DAT 00051200
   . OS:')                                         00051210
2  READ(8,299)A                                00051300
299  FORMAT(A80)                                00051500
   IF(A(7:9).EQ.'FIN')THEN                    00051600
     IB=1                                       00051605
     RETURN                                    00051606
   ENDIF                                       00051607
   IF(IMPZ.EQ.1)CALL IMPR                     00051608
   CALL LIMPIA                                00051610
   ICOM=1                                     00051700
   J=1                                        00051800
   IIR=0                                      00051900
   INUM=0                                    00052000
   MNJM=0                                    00052100
   IZ=0                                      00052200
   K=0                                       00052300
   D3 50 I=7,8)                              00052400
   IF(A(I:1).EQ.FIN.AND.(INJM.EQ.1.OR.IZ.EQ.1))GO TO 18 00052500
   IF(A(I:1).EQ.FIN)GO TO 60                 00052600
   IF(A(I:1).EQ.SPA)GO TO 49                 00052700

```

IF(A(I:1).EQ.COM)GO TO 2)	00052800
IF(A(I:1).GE.'0'.AND.A(I:1).LE.'9')GO TO 25	00052900
WRITE(8,305)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00053000
305 FORMAT(1X,4A1,'*** ERROR: UN CARACTER NO ES NUMERICO')	00053200
GO TO 630	00053300
18 IFIN=1	00053400
19 ICDH=0	00053500
20 IF(ICDH.EQ.1)GO TO 5)	00053600
IF(IZ.EQ.0)GO TO 22	00053700
A(J:J+1)='00'	00053800
J=J+2	00053900
IZ=0	00054000
GO TO 24	00054100
22 CONTINUE	00054200
IF(K.NE.1)GO TO 23	00054300
A(J:J)=A(J-1:J-1)	00054400
A(J-1:J-1)='0'	00054500
J=J+1	00054600
23 BF(1:2)=A(J-2:J-1)	00054700
READ(BF,320)IS	00054800
320 FORHAT(IZ)	00054900
IF(IS.GE.NS)GO TO 26	00055000
24 ICDH=1	00055100
IIN=0	00055200
K=0	00055300
INUH=0	00055400
NNUH=NNUH+1	00055500
IF(IFIN.EQ.1)GO TO 6)	00055600
GO TO 50	00055700
25 IF(IIN.EQ.1)GO TO 51	00055800
IF(A(I:1).EQ.'0'.AND.INUH.EQ.0)GO TO 48	00055900
K=K+1	00056000
IF(K.GT.2)GO TO 27	00056100
A(J:J)=A(I:1)	00056200
J=J+1	00056300
ICDH=0	00056400
IZ=0	00056500
INUH=1	00056600
GO TO 50	00056700
26 WRITE(8,338)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00056800
338 FORHAT(1X,4A1,'*** ERROR: UN VALDR DE MINTERMINO SOBREPASA MAXIMO	00057000
'VALOR')	00057100
GO TO 630	00057200
27 WRITE(8,340)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00057300
340 FORHAT(1X,4A1,'*** ERROR: HAS DE DOS DIGITO PARA REPRESENTAR UN N	00057500
'TERMINO')	00057600
GO TO 630	00057700
48 IZ=1	00057800
ICDH=0	00057900
GO TO 50	00058000
49 IF(I.EQ.80.AND.(INUH.EQ.1.OR.IZ.EQ.1))GO TO 19	00058100
IF(INUH.EQ.0.AND.IZ.EQ.0)GO TO 50	00058400
IIN=1	00058500
GO TO 50	00058600
51 WRITE(8,365)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00058700
365 FORMAT(1X,4A1,'*** ERROR: SE OMITIO UNA COMA')	00058900
GO TO 630	00059000
50 CONTINUE	00059100
IF(NNUH.EQ.0)GO TO 6	00059400
INI=IFI+1	00059500
IFI=IFI+NNUH	00059600
IKK=1	00059700
DO 66 IK=1,NNUH	00059800
66 IKK=A(IKK:IKK+1)	00059900

66	IKK=IKK+2	00066000
	READ(8,400)(IDD(JJ),JJ=INI,IFI)	000661300
400	FORMAT(I2)	000660400
	GO TO 6	000660500
60	IF(NNUM.EQ.0)GO TO 70	000660600
	INI=IFI+1	000660700
	IFI=IFI+NNUM	000660800
	IKK=1	000660900
	DO 76 IK=1,NNUM	000661000
	B(IK)=A(IKK:IKK+1)	000661100
76	IKK=IKK+2	000661200
	READ(8,400)(IDD(JJ),JJ=INI,IFI)	000661300
70	CONTINUE	000661400
	IF(IFI.EQ.1)GO TO 610	000661810
	I=1	000661900
	J=2	000662000
520	IF(IDD(I).NE.IDD(J))GO TO 600	000662100
	IF(J.EQ.IFI)GO TO 535	000662200
	DO 530 K=J,IFI-1	000662300
530	IDD(K)=IDD(K+1)	000662400
535	IFI=IFI-1	000662700
	IF(J.GT.IFI)GO TO 610	000663000
	GO TO 520	000663100
600	J=J+1	000663200
	IF(J.LE.IFI)GO TO 520	000663300
610	I=I+1	000663400
	J=I+1	000663500
	IF(I.LT.IFI)GO TO 520	000663800
610	CONTINUE	000663810
	RETURN	000664200
630	IF(IMPZ.EQ.1)CALL IMPR	000664210
	GO TO 6	000664220
	END	000664300
*****		
	SUBROUTINE BINARI(N,M,IFI)	000664310
	DIMENSION M(64),M1(4)	000664400
	COMMON /C/ IMPZ,ICH	000664500
	COMMON CONTRL(28),CURSOR(80)	000664510
	CHARACTER CONTRL*1,CURSOR*1	000664520
	CHARACTER A*80,FIN*1,COM*1,SPA*1,M1*6,M*6	000664600
	DATA (M1(I),I=1,40)/40*' ' //	000664700
	DATA SPA,COM,FIN/' ',' ',' ',' ' //	000664800
	WRITE(8,4)CONTRL(8)	000664805
4	FORMAT(4A1)	000664807
10	CALL LIMPZ	000664810
	WRITE(8,299)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	000664900
	CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	000664910
299	FORMAT(1X,4A1,'PROPORCIONA LOS MINTERMINOS SEPARADOS POR CUMA PARA	000665000
	' FIN DE TEXTO :',4A1,' DATOS:')	000665100
	READ(8,300)A	000665200
	IF(IMPZ.EQ.1)CALL IMPR	000665210
300	FORMAT(A00)	000665300
	IFIN=0	000665400
	NC=0	000665500
	NM=1	000665600
	IIN=0	000665700
	ICOM=1	000665800
	DO 50 I=7,80	000665900
	IF(A(I:1).EQ.FIN.AND.NC.NE.0)GO TO 10	000666000
	IF(A(I:1).EQ.FIN)GO TO 50	000666100
	IF(A(I:1).EQ.SPA)GO TO 40	000666200
	IF(A(I:1).EQ.COM)GO TO 30	000666300
	IF(A(I:1).EQ.'0'.OR.A(I:1).EQ.'1')GO TO 15	000666400
	WRITE(8,305)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	000666500



```

305 FORMAT(1X,4A1,'*** ERROR: UNO DE LOS CARACTERES NO ES UN DIGITO BI00066600
  *NARIO') 00066700
  GO TO 85 00066800
15 IF(IIN.EQ.1)GO TO 51 00066900
  IF(NC.EQ.6)GO TO 52 00067000
  ICOM=0 00067100
  NC=NC+1 00067200
  M1(NM)(NC:NC)=A(I:I) 00067300
  GO TO 50 00067400
18 IFIN=1 00067500
19 ICOM=0 00067600
30 IF(ICOM.EQ.1)GO TO 5) 00067700
  IF(NC.NE.N)GO TO 52 00067800
  NC=0 00067900
  NM=NM+1 00068000
  IIN=0 00068100
  ICOM=1 00068200
  IF(IFIN.EQ.1)GO TO 6) 00068300
  GO TO 50 00068400
49 IF(I.EQ.80.AND.NC.NE.2)GO TO 19 00068500
  IF(NC.EQ.0)GO TO 5) 00068600
  IIN=1 00068700
  GO TO 50 00068800
51 WRITE(8,310)CONTRL(23),CJNTRL(28),CURSOR(1),CJRSOR(19) 00068900
310 FORMAT(1X,4A1,'*** ERROR: SE OMITIO UNA COMA') 00069000
  GO TO 85 00069100
52 WRITE(8,320)CONTRL(23),CJNTRL(28),CURSOR(1),CJRSOR(19) 00069200
320 FORMAT(1X,4A1,'*** ERROR: EL NUMERO DE DIGITOS NO ES IGUAL AL NUME00069300
  *RD',/,1X,' DE VARIABLES EN EL MAPA') 00069400
  GO TO 85 00069500
50 CONTINUE 00069600
  IF(NM.EQ.1)GO TO 19 00069700
  INI=IFI+1 00069800
  IFI=IFI+(NM-1) 00069900
  KL=0 00070200
  DO 55 IL=INI,IFI 00070300
  KL=KL+1 00070400
55 M(IL)=M1(KL) 00070500
  GO TO 10 00070600
60 IF(NM.EQ.0)GO TO 70 00070700
  INI=IFI+1 00070800
  IFI=IFI+(NM-1) 00070900
  KL=0 00071000
  DO 63 IL=INI,IFI 00071100
  KL=KL+1 00071200
63 M(IL)=M1(KL) 00071300
70 IF(IFI.EQ.1)GO TO 83 00071400
83 RETURN 00071800
85 IF(IMPZ.EQ.1)CALL IMPR 00071820
  GO TO 10 00071820
  END 00071900
*****00071910
SUBROUTINE INIELE(ID,IFI,IDIR,VD) 00072000
DIMENSION IDIR(64),ID(64),VD(64) 00072100
CHARACTER VD*1 00072200
DO 20 I=1,IFI 00072300
  J1=ID(I)+1 00072400
  J2=IDIR(J1) 00072500
20 VD(J2)=*1* 00072600
RETURN 00072700
END 00072800
*****00072810
SUBROUTINE CONDEC(N,M,IFI,IO) 00072900
DIMENSION M(64),ID(64) 00073000

```

```

CHARACTER M*6
DD 60 K=1,IFI
KF=0
DD 50 I=1,N
IF(M(K)(I:I).EQ.'0')GO TO 50
KF=1
ID(K)=ID(K)+2.**(N-I)
50 CONTINUE
IF(KF.EQ.0)ID(K)=0
60 CONTINUE
RETURN
END

```

```

*****
SUBROUTINE COMBI(N,IFI,ID,M)
DIMENSION ID(64),M(64)
CHARACTER M*6
DD 10 I=1,IFI
K=N
VT=FLOAT(ID(I))
M(I)(1:6)='000000'
IF(VT.EQ.0.)GO TO 1)
3 IF(VT.NE.1.0)GO TO 5
M(I)(K:K)='1'
GO TO 10
5 RR=VT/2.
IR=RR
RF=RR-IR
IF(RF.EQ.0.)GO TO 6
M(I)(K:K)='1'
6 VI=IR
K=K-1
GO TO 3
10 CONTINUE
RETURN
END

```

```

*****
SUBROUTINE DECHAP(VD,NREN,MCOL,BUF,IMAT,KBAN)
DIMENSION IVE(16),VD(64),IMAT(8,8),BUF(640)
CHARACTER BUF*1,VD*1,BF*2
DATA (IVE(I),I=1,16)/17,97,177,257,337,417,497,577,51,131,211
.,291,371,451,531,611/
IF(KBAN.EQ.1)GO TO 8
DD 5 I=1,640
5 BJF(I)=' '
6 IVD=0
DD 15 I=1,NREN
IJ=IVE(I)
IK=IVE(I+8)
DD 15 J=1,MCOL
IF(KBAN.EQ.1)GO TO 1)
WRITE(BF,20)IMAT(I,J)
BUF(IJ)=BF(1:1)
BJF(IJ+1)=BF(2:2)
IJ=IJ+3
10 IVD=IVD+1
BJF(IK)=VD(IVD)
IK=IK+3
15 CONTINUE
20 FORMAT(I2)
RETURN
END

```

```

*****
SUBROUTINE VALIDA(IFI,IFI,IDD,LD,N,LIN,LIN,IVD,VEC1,NIP,IP
.,IIP,MAT)

```

DIMENSION IOD(64), IOC(64), LIN(19), LIM(19), IVD(19,9)	00079010
DIMENSION VEC(64), IP(29), VEC1(64), MAT(19,64)	00079020
COMMON CONTRL(28), CURSOR(8)	00079022
COMMON /C/ IMPZ, ICH	00079023
CHARACTER CONTRL*1, CURSOR*1	00079024
CHARACTER MAT*14	00079027
INTEGER VEC, VH, VEC1	00079031
CALL LIMPIA	00079035
LK=0	00079100
DO 40 I=1, IFII	00079200
DO 30 J=1, IFI	00079300
39 IF(IOD(I).EQ.IOC(J))GO TO 38	00079400
WRITE(8,35)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00079500
35 FORMAT(1X,4A1,'UNO DE LOS ELEMENTOS PROPORCIONADOS NO ES UN MINTERMINUS')	00079600
RETURN	00079800
38 LK=LK+1	00079900
VEC(LK)=J	00080100
40 CONTINUE	00080200
42 ICAH=0	00080300
DO 45 I=1, IFII-1	00080400
IF(VEC(I).LE.VEC(I+1))GO TO 43	00080500
VH=VEC(I)	00080600
VEC(I)=VEC(I+1)	00080700
VEC(I+1)=VH	00080800
ICAH=1	00080900
43 CONTINUE	00081000
IF(ICAH.EQ.1)GO TO 42	00081100
IC=0	00081200
LIMT=2**H	00081300
IF(IFII.GE.LIMT)GO TO 50	00081400
45 IEX=2**IC	00081500
IF(IFII.LE.IEX)GO TO 63	00081600
IF(ICX.GT.IFI)GO TO 57	00081700
IC=IC+1	00081800
GO TO 45	00081900
54 IF(IFII.EQ.LIMT)GO TO 63	00082000
WRITE(8,53)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00082100
53 FORMAT(1X,4A1,'SE SOBREPASA EL VALOR MAXIMO DE MINTERMINUS')	00082300
RETURN	00082400
57 WRITE(8,59)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00082500
59 FORMAT(1X,4A1,'NUMERO DE MINTERMINUS ERRONEO')	00082700
RETURN	00082800
63 IF(IC.EQ.0)GO TO 87	00082900
IBC=IC+1	00083000
LI=IEX	00083100
K=LIN(IBC)	00083200
L=K*LI-1	00083300
KR=LIM(IBC)	00083400
DO 80 I=1,KR	00083500
LK=1	00083600
DO 72 J=K,L	00083700
IF(VEC(LK).NE.IVD(J))GO TO 75	00083800
72 LK=LK+1	00083900
GO TO 90	00084000
75 K=L+1	00084100
L=L*LI	00084200
80 CONTINUE	00084300
WRITE(8,85)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00084400
85 FORMAT(1X,4A1,'TU REDUCCION ES INVALIDA')	00084600
RETURN	00084700
87 K=VEC(LK)	00084710
90 IF(IIP.EQ.1)GO TO 100	00085100
IP(NIP)=K	00085200

IIP=1	00095210
GO TO 115	00085300
100 DO 110 J=1,NIP	00085400
110 IF(IP(J).EQ.K)GO TO 120	00085500
NIP=NIP+1	00085600
IP(NIP)=K	00085700
115 DO 112 IM=1,IFI	00085800
KI=VECI(IM)	00085900
112 VEC1(KI)=1	00086000
WRITE(8,91)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19),	00086010
MAT(TBC,I)(1:N)	00086020
91 FORMAT(1X,4A1,'TU REDUCCION ES VALIDA',5X,A6)	00086030
RETURN	00086100
120 WRITE(8,125)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00086200
125 FORMAT(1X,4A1,'ESTA REPETIDA TU RESPUESTA: NO REGISTRADA')	00086400
RETURN	00086500
END	00086600
*****	00086610
SUBROUTINE DIA G(VECI,NIP,NOP,TAB,II,DIR,HAT2,IFI,IP,NIS,N)	00086700
DIMENSION VEC1(64),TAB(40),DIR(15),MAT2(70),IP(20)	00086710
COMMON CONTRL(28),CURSOR(80)	00086712
COMMON /C/ IMPZ,ICH	00086713
CHARACTER CONTRL*1,CJRSOR*1	00086714
CHARACTER TAB*14,B*3,MAT2*30	00086720
INTEGER VEC1,DIR,VAL	00086730
DATA ICON/1/	00086735
WRITE(8,7)CONTRL(8)	00086760
7 FORMAT(A1)	00086770
ICC=3	00086800
DO 10 I=1,IFI	00086900
IF(VECI(I).EQ.1)GO TO 10	00087000
ICC=ICC+1	00087100
10 CONTINUE	00087200
IF(ICC.EQ.0)GO TO 22	00087300
WRITE(8,20)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00087400
20 FORMAT(1X,4A1,'TU RESPUESTA NO CUBRE TODOS LOS MINTERMINOS',)	00087600
GO TO 26	00087700
22 WRITE(8,24)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00087800
24 FORMAT(1X,4A1,'TU RESPUESTA CUBRE EN SU TOTALIDA LOS MINTERMINOS')	00088000
26 IF(NIP.EQ.NOP)GO TO 28	00088100
IF(NIP.LF.NOP)GO TO 30	00088200
WRITE(8,44)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(20)	00088300
GO TO 50	00088500
28 WRITE(8,40)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(20)	00088600
GO TO 50	00088800
30 WRITE(8,42)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(20)	00088900
40 FORMAT(1X,4A1,'TU NUM DE REDUCCIONES ES IGUAL AL NUM DE IMPLICANTE	00089100
.S PRINOS ESENCIALES')	00089200
42 FORMAT(1X,4A1,'TU NUM DE REDUCCIONES ES MENOR AL NUM DE IMPLICANTE	00089300
.S PRINOS ESENCIALES')	00089400
44 FORMAT(1X,4A1,'TU NUM DE REDUCCIONES ES MAYOR AL NUM DE IMPLICANTE	00089500
.S PRINOS ESENCIALES')	00089600
50 CONTINUE	00089700
DO 200 JR=1,NIP	00089710
K=IP(JR)	00089720
DO 60 I=1,II	00089800
B(1:3)=TAB(I)(11:13)	00089900
READ(8,55)IVAL	00090000
55 FORMAT(I3)	00090100
IF(IVAL.EQ.K)GO TO 70	00090200
60 CONTINUE	00090300
ICON=ICON+1	00090310
WRITE(8,65)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(ICON)	00090400
65 FORMAT(1X,4A1,'NO ES UN IMPLICANTE PRIMO')	00090600

```

GO TO 200
70 IF (TAB(I)(1:1).EQ.'*') GO TO 80
   ICON=ICON+1
   WRITE(8,75)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(ICON)
     ,TAB(I)(3:3+N)
75 FORMAT(1X,4A1,'ES IMPLICANTE PRIMO, PERD NO ESENCIAL',5X,A6)
   GO TO 90
80 ICON=ICON+1
   WRITE(8,85)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(ICON),
     ,TAB(I)(3:3+N)
85 FORMAT(1X,4A1,'ES UN IMPLICANTE PRIMO ESENCIAL',5X,A6)
   GO TO 200
90 DO 100 I=1,NIS
   IDR=DIR(I)
   KN=1
92 IF (MAT2(IDR)(KN:KN+1).EQ.'*') GO TO 100
   UC(1:2)=MAT2(IDR)(KN:KN+1)
   READ(8,94)VA1
94 FORMAT(I2)
   UC(1:3)=TAB(VA1)(11:13)
   READ(8,55)IVA11
   IF (IVA11.EQ.K) GO TO 120
   KN=KN+2
   GO TO 92
100 CONTINUE
   ICON=ICON+1
   WRITE(8,105)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(ICON)
105 FORMAT(1X,4A1,'NO ES IMPLICANTE PRIMO SECUNDARIO ESENCIAL')
   GO TO 200
120 ICON=ICON+1
   WRITE(8,125)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(ICON)
     ,TAB(VA1)(3:3+N)
125 FORMAT(1X,4A1,'ES UN IMPLICANTE PRIMO SECUNDARIO ESENCIAL',5X,A6)
200 CONTINUE
   IF (IMP2.EQ.1) CALL IMPR
   RETURN
   END
*****
SUBROUTINE SOLUCN,IFIN,M,ID,LIN,LIN,DIR,IVD,TAB,II,MAT2,NUP,NIS,
.MAT)
DIMENSION M(64),NS(64),ID(64),NZ(19),ILON(10),IN2(10)
,INI(10),IFI(10),IP(10),I(2),MAT(10,64),
,IVD(10,4),TAB(60),IVC(64),MAT1(75),MAT2(70)
,LIH(1),LIN(1),COS(64),NMI(64),DIR(15)
INTEGER COS,DIR
CHARACTER M*6,TT*6,B*3,K*1,ELE*14,MAT*14,UNO*1,MATT*3
,SPA*14,TAG*14,SPA2*3,CC*1,MAT1*62,MAT2*30
DATA X,UNO,SPA,SPA2,CC/'X','1',' ',' ','C'/'
NJP=0
DO 10 I=1,IFIN
INUM=0
DO 7 J=1,N
IF (M(I)(J:J).NE.JND) GO TO 7
INUM=INUM+1
7 CONTINUE
NS(I)=INUM
10 CONTINUE
DO 500 I=1,IFIN
500 WRITE(6,510)I,NS(I)
510 FORMAT(1X,'NS',I2,'=',I,12)
IS=IFIN-1
15 IBJR=0
DO 20 IK=1,IS
IF (NS(IK).LE.NS(IK+1)) GO TO 20

```

```

00090700
00090800
00090900
00091000
00091100
00091200
00091300
00091400
00091500
00091600
00091700
00091800
00091900
00092000
00092100
00092200
00092300
00092400
00092500
00092600
00092700
00092800
00092900
00093000
00093100
00093200
00093300
00093400
00093500
00093600
00093700
00093800
00093900
00094000
00094100
00094200
00094300
00094400
00094500
00094600
00094700
00094800
00094900
00095000
00095100
00095200
00095300
00095400
00095500
00095600
00095700
00095800
00095900
00096000
00096100
00096200
00096300
00096400

```

IT=NS(IK)	00096500
NS(IK)=NS(IK+1)	00096600
NS(IK+1)=IT	00096700
IT(1:N)=M(IK)(1:N)	00096800
M(IK)(1:N)=M(IK+1)(1:N)	00096900
M(IK+1)(1:N)=IT(1:N)	00097000
I=ID(IK)	00097100
ID(IK)=ID(IK+1)	00097200
ID(IK+1)=I	00097300
I0UR=1	00097400
20 CONTINUE	00097500
IF(I0UR.EQ.1)GO TO 15	00097600
DO 520 I=1,IFIN	00097700
520 WRITE(6,525)M(I)(1:N),ID(I),NS(I)	00097800
525 FORMAT(IX,'M=',A6,' ID=',I2,' NS=',I2)	00097900
IAL=-1	00098000
IPV=0	00098100
DO 30 I=1,IFIN	00098200
IF(IAL.EQ.NS(I))GO TO 26	00098300
IAL=NS(I)	00098400
IF(IPV.EQ.1)GO TO 24	00098500
NZ(I)=NS(I)	00098600
IN2(I)=1	00098700
IPV=1	00098800
IA=1	00098900
ILON(IA)=0	00099000
GO TO 26	00099100
24 IA=IA+1	00099200
ILON(IA)=0	00099300
NZ(IA)=NS(I)	00099400
IN2(IA)=IN2(IA-1)+ILON(IA-1)	00099500
26 ILON(IA)=ILON(IA)+1	00099600
30 CONTINUE	00099700
DO 540 I=1,IA	00099800
540 WRITE(6,545)IN2(I),ILON(I),I	00099900
545 FORMAT(IX,'IN2=',I2,' ILON=',I2,' I=',I2)	00100000
DO 40 I=1,IFIN	00100100
MAT(I,I)(1:14)=SPA	00100200
MAT(I,I)(1:N)=M(I)(1:N)	00100300
WRITE(8,36)I	00100400
36 FORMAT(I3)	00100500
MAT(I,I)(1:13)=8(I:3)	00100600
I0D=I0D+1	00100700
IVD(I0D)=I	00100800
40 CONTINUE	00100900
MAT(I,I)(1:14)=SPA	00101000
DO 550 I=1,IFIN	00101100
550 WRITE(6,555)MAT(I,I)(1:14),IVD(I)	00101200
555 FORMAT(IX,'MAT =',A14,' IVD=',I2)	00101300
50 IN=0	00101400
IMP=IMP+1	00101500
ND=IA-1	00101600
DO 55 II=1,ND+1	00101700
WRITE(6,545)IN2(II),ILON(II),II	00101800
INI(II)=IN2(II)	00101900
LIN(IMP)=LIN(IMP)+ILON(II)	00102000
55 IF(II)=ILON(II)	00102100
IF(IMP.NE.1)GO TO 57	00102200
LIN(IMP)=1	00102300
GO TO 59	00102400
57 LIN(IMP)=LIN(IMP-1)+LIN(IMP-1)*(2**((IMP-2)))	00102500
59 CONTINUE	00102600
DO 1000 IXT=1,IA	00102700
IXL=INI(IXT)	00102800

```

IX2=INI(IX1)+IFI(IX1)-1
DO 1000 IXY=IX1,IX2
1000 WRITE(6,3000)MAT(IMP,IXY)(1:14)
IF(IULT.EQ.1)GO TO 3500
IA=1
IW=0
IV=0
DO 300 LK=1,ND
IH=0
IBAN3=0
J1=INI(LK)
J2=INI(LK)+IFI(LK)-1
C WRITE(6,557)LK,J1,J2,IMP
C 557 FORMAT(1X,'LK=',I2,' J1 =',I2,' J2 =',I2,' IMP=',I2)
DO 250 J=J1,J2
IF(IMP.EQ.1)GO TO 75
ICON=0
DO 60 IB=1,N
IF(MAT(IMP,J)(IB:IB).NE.X)GO TO 60
ICON=ICON+1
IP(ICON)=IB
IF(ICON.EQ.(IMP))GO TO 65
60 CONTINUE
65 CONTINUE
75 CONTINUE
J3=INI(LK+1)
J4=INI(LK+1)+IFI(LK+1)-1
DO 200 MX=J3,J4
IBAN=0
IF(IMP.EQ.1)GO TO 85
DO 80 IB=1,ICON
LI=IP(IB)
IF(MAT(IMP,MX)(LI:LI).EQ.X)GO TO 80
GO TO 200
80 CONTINUE
85 IDIF=0
DO 95 L=1,N
IF(MAT(IMP,J)(L:L).EQ.MAT(IMP,MX)(L:L))GO TO 95
IDIR=L
IDIF=IDIF+1
IF(IDIF.GT.1)GO TO 200
95 CONTINUE
IF(IW.EQ.1)GO TO 115
IF(IV.EQ.1)GO TO 100
ILON(IA)=0
IV=1
IN2(IA)=1
GO TO 110
100 IA=IA+1
ILON(IA)=0
IN2(IA)=IN2(IA-1)+ILON(IA-1)
110 CONTINUE
115 ELE(1:N)=MAT(IMP,J)(1:N)
ELE(IDIR:IDIR)=X
IF(ILON(IA).EQ.0)GO TO 117
IY1=IN2(IA)
IY2=IN2(IA)+ILON(IA)-1
IBAN=0
DO 116 IY=IY1,IY2
IF(ELE(1:N).NE.MAT(IMP,J)(IY)(1:N))GO TO 116
IBAN=1
GO TO 117
116 CONTINUE
117 MAT(IMP,J)(10:10)=*C

```

```

00102900
00103000
00103100
00103110
00103200
00103300
00103400
00103500
00103600
00103610
00103700
00103800
00103900
00104000
00104100
00104200
00104300
00104400
00104500
00104600
00104700
00104800
00104900
00105000
00105100
00105200
00105300
00105400
00105500
00105600
00105700
00105800
00105900
00106000
00106100
00106200
00106300
00106400
00106500
00106600
00106700
00106800
00106900
00107000
00107100
00107200
00107300
00107400
00107500
00107600
00107700
00107800
00107900
00108000
00108100
00108200
00108300
00108400
00108500
00108600
00108700
00108800
00108900
00109000

```

```

MAT(IMP,MX)(10:1J)='C'
IF(IBAN.EQ.1)GO TO 23)
LON(IA)=LON(IA)+1
IN=IN+1
I(1)=J
I(2)=MX
IX=IDD+1
MAT(IMP+1,IN)(1:14)=SPA
MAT(IMP+1,IN)(1:N)=ELE(1:N)
IBAN3=1
WRITE(8,36)IX
MAT(IMP+1,IN)(11:13)=B(1:3)
MAT(IMP+1,IN+1)(1:14)=SPA
MAT(IMP+2,1)=SPA
IJF=2*(IMP-1)
IPRR=IDD+1
DO 125 IT=1,2
MH=I(IJF)
MATT(1:3)=MAT(IMP,MH)(11:13)
READ(MATT,36)IPD
DO 122 IH=1,IJF
IDD=IDD+1
IVD(IDD)=IVD(IPD)
IW=1
IPD=IPD+1
122 CONTINUE
125 CONTINUE
126 ICAM=0
DO 127 IXT=IPRR,IDD-1
IF(IVD(IXT).LE.IVD(IXT+1))GO TO 127
I/A=IVD(IXT)
IVD(IXT)=IVD(IXT+1)
IVD(IXT+1)=I/A
ICAM=1
127 CONTINUE
IF(ICAM.EQ.1)GO TO 126
23) CONTINUE
25) CONTINUE
300 CONTINUE
DO 2000 IXT=1,HD+1
IX1=INI(IXT)
IX2=INI(IXT)+IFI(IXT)-1
DO 2000 IXY=IX1,IX2
2000 WRITE(6,3000)MAT(IMP,IXY)(1:14)
3000 FORMAT(1X,' MAT = ',A14)
IF(IA.NE.1)GO TO 50)
IULT=1
IF(IBAN3.EQ.0)GO TO 3300
GO TO 50)
3300 DO 4000 IXT=1,IDD
4000 WRITE(6,5000)IVD(IXT),IXT
5000 FORMAT(10X,' IVD=',I2,' IXT=',I3)
5010 II=0
I=IMP+1
IHEN=I
6010 J=1
6020 IF(MAT(I,J)(11:13).EQ.SPA2(1:3))GO TO 6040
IF(MAT(I,J)(10:10).EQ.CC)GO TO 6030
II=II+1
WRITE(8,6027)I
6027 FORMAT(1I)
TAB(II)(1:14)=SPA
TAB(II)(2:2)=B(1:1)
TAB(II)(3:3+N-1)=MAT(I,J)(1:N)

```

```

001091 00
001092 00
001093 00
001094 00
001095 00
001096 00
001097 00
001098 00
001099 00
001100 00
001101 00
001102 00
001103 00
001104 00
001105 00
001106 00
001107 00
001108 00
001109 00
001110 00
001111 00
001112 00
001113 00
001114 00
001115 00
001116 00
001117 00
001118 00
001119 00
001120 00
001121 00
001122 00
001123 00
001124 00
001125 00
001126 00
001127 00
001128 00
001129 00
001130 00
001131 00
001132 00
001133 00
001134 00
001135 00
001136 00
001137 00
001138 00
001139 00
001140 00
001141 00
001142 00
001143 00
001144 00

```



```

TAB(I1)(11:13)=MAT(I,J)(11:13)
WRITE(6,6028)I1,TAB(I1)(1:14)
6028 FORMAT(1X,'I1=',I2,' TAB=',A14)
6030 J=J+1
GO TO 6020
6040 I=I-1
IF(I.NE.0)GO TO 6310
DO 6060 I=1,IFIN
IBAN=0
DO 6050 J=1,II
B(1:1)=TAB(J)(2:2)
READ(8,6027)IK
IKL=2*(IK-1)
B(1:3)=TAB(J)(11:13)
READ(8,36)J1
J2=J1+IKL-1
DO 6045 M4=J1,J2
IF(IVD(I).NE.IVD(M4))GO TO 6345
JJ=J
IBAN=IBAN+1
6345 CONTINUE
IF(IBAN.GT.1)GO TO 6350
6050 CONTINUE
TAB(JJ)(1:1)='*'
NOP=NOP+1
6060 CONTINUE
DO 6070 I=1,II
WRITE(6,6028)I,TAB(I)(1:14)
DO 6080 I=1,II
IF(TAB(I)(1:1).EQ.**)GO TO 6380
NOP=NOP+1
B(1:1)=TAB(I)(2:2)
READ(8,6027)IK
IKL=2*(IK-1)
B(1:3)=TAB(I)(11:13)
READ(8,36)J1
J2=J1+IKL-1
DO 6075 LJ=J1,J2
IND1=IVD(LJ)
6075 IVEC(IND1)=1
6080 CONTINUE
DO 6090 I=1,IFIN
WRITE(6,6092)IVEC(I)
6092 FORMAT(1X,' IVEC=',I3)
IXJ=0
DO 7200 I=1,IFIN
IF(IVEC(I).EQ.1)GO TO 7200
IXJ=IXJ+1
IF(IXJ.EQ.1)GO TO 6099
WRITE(6,6095)IXJ-1,MAT1(IXJ-1)(1:10)
6095 FORMAT(3X,I3,3X,A10)
6096 FORMAT(I2)
6099 WRITE(8,6096)I
MAT1(IXJ)(1:2)=B(1:2)
WRITE(6,7002)IXJ,MAT1(IXJ)(1:2)
7002 FORMAT(1X,'IXJ=',I3,'MAT1=',2A1)
IA=1
IU=2
DO 7198 J=1,II
IF(TAB(J)(1:1).EQ.***)GO TO 7198
B(1:3)=TAB(J)(11:13)
READ(8,36)I1I
B(1:1)=TAB(J)(2:2)
READ(8,6027)I1P

```

```

00114500
00114600
00114700
00114800
00114900
00115000
00115100
00115200
00115300
00115400
00115500
00115600
00115700
00115800
00115900
00116000
00116100
00116200
00116300
00116400
00116500
00116600
00116700
00116800
00116900
00117000
00117100
00117200
00117300
00117310
00117400
00117500
00117600
00117700
00117800
00117900
00118000
00118100
00118200
00118300
00118400
00118500
00118600
00118700
00118800
00118900
00119000
00119100
00119200
00119300
00119310
00119400
00119500
00119600
00119700
00119800
00119900
00120000
00120100
00120200
00120300
00120400
00120500

```

ITPP=2*(ITP-1)	00120600
ITPP=IFT+ITPP-1	00120700
DO 7100 IJ=IFT,ITPP	00120800
IF(IVD(IJ).NE.1)GO TO 7100	00120900
IA=IA+2	00121000
IU=IU+2	00121100
WRITE(6,6096)J	00121200
MAT1(IXJ)(IA:IU)=0(1:2)	00121300
MAT1(IXJ)(IU+1:IU+2)=SPA2(1:2)	00121400
WRITE(6,7003)IXJ,MAT1(IXJ)(1:10)	00121500
7003 FORMAT(1X,'IXJ2=',I3,'MAT1',A10)	00121600
GO TO 7198	00121700
7100 CONTINUE	00121800
GO TO 7198	00121810
7190 HOP=HOP+1	00121820
7198 CONTINUE	00121900
7200 CONTINUE	00122000
IF(IXJ.EQ.0)GO TO 980)	00122010
IF(IXJ.EQ.1)GO TO 7550	00122020
MAT1(IXJ+1)(3:6)=''	00122100
DO 7210 I=1,IXJ	00122200
7210 WRITE(6,7003)I,MAT1(I)(1:10)	00122300
IXX=0	00122400
I3=3	00122500
I2=4	00122600
IA=3	00122610
IJ=4	00122620
IXT=2	00122700
IM=4	00122800
7300 IF(MAT1(1)(I3:I2).NE.SPA2(1:2))THEN	00122900
IF(MAT1(2)(IA:IU).NE.SPA2(1:2))THEN	00123000
IXX=IXX+1	00123100
MAT2(IXX)(1:2)=MAT1(1)(I3:I2)	00123200
MAT2(IXX)(3:4)=MAT1(2)(IA:IU)	00123300
MAT2(IXX)(5:6)=SPA2(1:2)	00123310
IA=IA+2	00123400
IU=IU+2	00123500
ELSE	00123600
I3=I3+2	00123700
I2=I2+2	00123800
IA=3	00123900
IU=4	00124000
ENDIF	00124100
ELSE	00124200
GO TO 7590	00124300
ENDIF	00124400
GO TO 7300	00124700
7550 I3=1	00124710
I2=2	00124720
IXX=0	00124730
MAT1(3)(3:4)=''	00124740
7560 I3=I3+2	00124750
I2=I2+2	00124760
IF(MAT1(1)(I3:I2).EQ.SPA2(1:2))GO TO 7590	00124770
IXX=IXX+1	00124775
MAT2(IXX)(1:2)=MAT1(1)(I3:I2)	00124780
MAT2(IXX)(3:10)=''	00124785
GO TO 7560	00124790
7590 DO 7620 I=1,IXX	00124800
7620 WRITE(6,7630)I,MAT2(I)(1:10)	00124900
7630 FORMAT(5X,'I=',I3,5X,'MAT2=',A10)	00125000
IF(IXJ.EQ.1)GO TO 9035	00125010
7640 IXT=IXT+1	00125100
I3=3	00125200

	I2=4	00125300
	IF(MAT1(IXT)(I3:I2).NE.SPA2(1:2))GO TO 7700	00125400
	GO TO 8800	00125500
7700	DO 7750 I=1,IXX	00125600
	MAT2(I)(IM+1:IM+2)=MAT1(IXT)(I3:I2)	00125700
7750	MAT2(I)(IM+3:IM+4)=SPA2(1:2)	00125800
	I=IXX	00125900
7800	I3=I3+2	00126000
	I2=I2+2	00126100
	IF(MAT1(IXT)(I3:I2).NE.SPA2(1:2))GO TO 7850	00126200
	GO TO 6700	00126300
7850	IYY=1	00126400
	DO 7900 I=1,IXX	00126500
	IYY=IYY+1	00126600
7900	MAT2(IYY)(I:IM)=MAT2(I)(I:IM)	00126700
	IM=I	00126800
	DO 7950 I=IM,IYY	00126900
	MAT2(I)(IM+1:IM+2)=MAT1(IXT)(I3:I2)	00127000
7950	MAT2(I)(IM+3:IM+4)=SPA2(1:2)	00127100
	GO TO 7800	00127200
8700	IXX=IYY	00127300
	IM=IM+2	00127400
	GO TO 7640	00127500
8800	DO 8890 IL1=1,IXX	00127600
8890	WRITE(6,5595)IL1,MAT2(IL1)(I:IM)	00127700
8895	FOR MAT(IX,'IL1=',I3,' MAT2=',A30)	00127800
	DO 9030 I=1,IXX	00127900
	IA=1	00128000
	I2=2	00128100
	I3=3	00128200
	I4=4	00128300
9030	IF(MAT2(I)(I3:I4).EQ.SPA2(1:2))GO TO 9030	00128400
	IF(MAT2(I)(IA:I2).EQ.MAT2(I)(I3:I4))THEN	00128500
9010	MAT2(I)(I3:I4)=MAT2(I)(I3+2:I4+2)	00128600
	IF(MAT2(I)(I3:I4).EQ.SPA2(1:2))GO TO 9015	00128700
	I3=I3+2	00128800
	I4=I4+2	00128900
	GO TO 9010	00129000
	ENDIF	00129100
9015	IA=IA+2	00129200
	I2=I2+2	00129300
	I3=IA+2	00129400
	I4=I2+2	00129500
	GO TO 9030	00129600
9030	CONTINUE	00129700
9035	DO 9200 I=1,IXX	00129800
	IA=1	00129900
	I2=2	00130000
9040	IF(MAT2(I)(IA:I2).EQ.SPA2(1:2))GO TO 9200	00130100
	B(1:2)=MAT2(I)(IA:I2)	00130200
	READ(6,6096)J	00130300
	B(1:1)=TAB(J)(2:2)	00130400
	READ(6,6027)JJ	00130500
	COS(I)=COS(I)+JJ	00130600
	NNI(I)=NNI(I)+1	00130700
	IA=IA+2	00130800
	I2=I2+2	00130900
	GO TO 9040	00131000
9200	CONTINUE	00131100
	MM=NNI(I)	00131200
	DO 9300 I=2,IXX	00131300
	IF(MM.LE.NNI(I))GO TO 9300	00131400
	MM=NNI(I)	00131500
9300	CONTINUE	00131600
		00131700
		00131800
		00131900
		00132000
		00132100
		00132200
		00132300
		00132400
		00132500
		00132600

MEN1=5000	00132700
DO 9400 I=1,IXX	00132800
IF(NMI(I).NE.MEN)GO TO 9400	00132900
IF(MEN1.LE.COS(I))GO TO 9400	00133000
MEN1=COS(I)	00133100
9400 CONTINUE	00133200
DO 9500 I=1,IXX	00133300
IF(NMI(I).NE.MEN)GO TO 9500	00133400
IF(COS(I).NE.MEN1)GO TO 9500	00133500
KJ=KJ+1	00133600
DIR(KJ)=I	00133700
9500 CONTINUE	00133800
GO TO 9920	00133900
9920 DO 9600 I=1,KJ	00134000
9600 WRITE(6,9700)I,DIR(I)	00134100
9700 FORMAT(1X,' DIR',I2,'=',I3)	00134200
NIS=KJ	00134300
NOP=NOP+1	00134400
9800 RETURN	00134500
END	00134600
SUBROUTINE LIMPIA	00134700
COMMON CONTRL(28),CURSOR(80)	00134800
CHARACTER CONTRL=1,CURSJR=1	00134900
CHARACTER BLCK(80)=4	00135000
DATA BLCK/80*' '	00135100
WRITE(8,100)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1),	00135200
BLCK(I),I=1,20),CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19),	00135300
BLCK(I),I=1,80)	00135400
100 FORMAT(4A1,20A4,4A1,80A4)	00135500
RETURN	00135600
END	00135700
SUBROUTINE IMPR	00135800
COMMON CONTRL(28),CURSOR(80)	00135900
COMMON /C/ IMPZ,ICH	00136000
DIMENSION A(26),LEE(1920)	00136100
CHARACTER CONTRL=1,CURSJR=1	00136200
CHARACTER A=82,LEE=1	00136300
WRITE(8,30)CONTRL(23),CONTRL(28),CURSOR(77),CURSOR(24),	00136400
CONTRL(14),CONTRL(16),CONTRL(17)	00136500
30 FORMAT(6A1,'I',A1)	00136600
DO 5 I=1,82	00136700
A(I)(1:I)='*'	00136800
5 A(26)(1:I)='*'	00136900
DO 10 I=2,25	00137000
A(I)(1:I)='*'	00137100
10 A(I)(82:82)='*'	00137200
READ(8,30)(LEE(I),I=1,1920)	00137300
30 FORMAT(160(A1,A1,A1,A1,A1,A1,A1,A1,A1,A1,A1,A1))	00137400
J=1	00137500
DO 40 I=2,25	00137600
DO 35 JK=2,81	00137700
A(I)(JK:JK)=LEE(J)(1:1)	00137800
J=J+1	00137900
35 CONTINUE	00138000
40 CONTINUE	00138100
ICH=ICH+1	00138200
IF(ICH.GT.2)THEN	00138300
WRITE(9,44)	00138400
44 FORMAT(1H1)	00138500
ICH=1	00138600
ENDIF	00138700
WRITE(9,45)	00138800
45 FORMAT(////)	00138900
DO 60 I=1,26	00139000

```
6J WRITE(9,70)A(I)(1:62)
70 FORMAT(6X,AB2)
WRITE(8,75)CONTROL(23),CONTROL(28),CURSOR(77),CURSOR(24)
75 FORMAT(4A1,' '*)
RETURN
END
```

```
00135330
00135340
00135350
00135360
00135370
00135380
```





\*DATOS:0,2,3,7,8\*

0 1 3 2  
4 5 7 6  
12 13 15 14  
8 9 11 10

0 0 0 0  
0 0 0 0  
0 0 0 0  
0 0 0 0

\* PROPORCIONA LOS MINTERMINOS EN SU EQUIVALENTE DECIMAL SEPARADOS POR COMAS  
\* Y TERMINA CON EL CARACTER => ;

??!?

1 0 1 1  
0 0 1 0  
0 0 0 0  
1 0 0 0

\* ESTOS SON LOS MINTERMINOS QUE PROPORCIONASTE UBICADOS EN EL MAPA DE KARNAUGH

??!?



\*\*\*\*\*  
\*DATOS:3,7;

\* 0 1 3 2  
\* 4 5 7 6  
\* 12 13 15 14  
\* 8 9 11 10

\* 1 0 1 1  
\* 0 0 1 0  
\* 0 0 0 0  
\* 1 0 0 0

\*PROPORCIONA TU REDUCCION, HACIENDO REFERENCIA A LOS NINTERMINOS POR SU  
\*EQUIVALENTE DECIMAL, SEPARADOS POR COMAS Y TERMINA CON EL CARACTER (?)  
\*-PARA TERMINAR DE PROPORCIONAR REDUCCIONES DIGITA LA PALABRA => FIN

\*\*\*\*\*  
\*??I?\*

\* 0 1 3 2  
\* 4 5 7 6  
\* 12 13 15 14  
\* 8 9 11 10

\* 1 0 1 1  
\* 0 0 1 0  
\* 0 0 0 0  
\* 1 0 0 0

\*TU REDUCCION ES VALIDA

\*\*\*\*\*  
\*??I?\*

\*\*\*\*\*  
\*DATOS:0,8\*

\* 0 1 3 2  
\* 4 5 7 6  
\* 12 13 15 14  
\* 8 9 11 10

\* 1 0 1 1  
\* 0 0 1 0  
\* 0 0 0 0  
\* 1 0 0 0

\*PROPORCIONA TU REDUCCION, HACIENDO REFERENCIA A LOS MINTERINOS POR SU  
\*EQUIVALENTE DECIMAL, SEPARADOS POR COMAS Y TERMINA CON EL CARACTER (;)  
\*-PARA TERMINAR DE PROPORCIONAR REDUCCIONES DIGITA LA PALABRA => FIN

??1?\*

\*\*\*\*\*  
\* 0 1 3 2  
\* 4 5 7 6  
\* 12 13 15 14  
\* 8 9 11 10

\* 1 0 1 1  
\* 0 0 1 0  
\* 0 0 0 0  
\* 1 0 0 0

\*TU REDUCCION ES VALIDA

??1?\*

\*\*\*\*\*  
\*DATOS:2,3\*

0 1 3 2  
4 5 7 6  
12 13 15 14  
8 9 11 10

1 0 1 1  
0 0 1 0  
0 0 0 0  
1 0 0 0

\*PROPORCIONA TU REDUCCION, HACIENDO REFERENCIA A LOS HINTERMINOS POR SU  
\*EQUIVALENTE DECIMAL, SEPARADOS POR COMAS Y TERMINA CON EL CARACTER (;)  
\*PARA TERMINAR DE PROPORCIONAR REDUCCIONES DIGITA LA PALABRA => FIN

??I?\*

0 1 3 2  
4 5 7 6  
12 13 15 14  
8 9 11 10

1 0 1 1  
0 0 1 0  
0 0 0 0  
1 0 0 0

\*TU REDUCCION ES VALIDA

??I?\*

LA FUNCION OBTENIDA ES:

$$F(A,B,C,D) = \overline{A}CD + B\overline{C}D + ABC$$

ES UNA FUNCION OPTIMA

OTRAS FUNCIONES OPTIMAS QUE SE PUEDEN OBTENER :

$$F(A,B,C,D) = \overline{A}CD + B\overline{C}D + ABD$$

### IV.3 MANUAL DE USUARIO

El comando para correr este programa dependera de la maquina que se utilice, por lo que debera investigarse en el lugar correspondiente.

El primer dato que solicita el programa es el numero de variables que tiene la funcion a minimizar. Este debera ser un numero entero no mayor de 6 ni menor de 2.

Entonces se presentara al usuario un mapa tentativo con todos los posibles minterminos para el numero de variables propuestas y le dara la opcion de renombrar las variables, esto es con el objeto de asignarle nombres que use mas comunmente o bien para cambiar la posicion de los minterminos, para trabajar con un mapa ordenado en forma diferente. En el caso de desear renombrarlas, las nuevas deberan ser de 1 solo caracter.

Se presentaran a continuacion las formas en que el usuario puede proporcionar los minterminos, estas formas son:

- Decimal: con esta opcion, se deberan dar los minterminos en forma decimal separados por comas y terminar con el caracter '/' y presionar la tecla <RETURN>.
- Cadenas: con esta opcion, los minterminos se deberan dar en su representacion canonica, es decir, la variable con su valor binario asociado de la siguiente manera:

Si la funcion tiene 3 variables y uno de los minterminos es '5' en su equivalente decimal y el peso de las variables de mayor a menor es CBA, se dara:

C1B0A1

de esta manera se daran todos los minterminos separados por comas y para terminar se indicara con el caracter '/' y presionando la tecla <RETURN>.

- Binaria: la forma de dar los minterminos en esta opcion es semejante a la de Cadenas, solo que no se dara el nombre de la variable asociado, unicamente su representacion binaria, por esto es importante tener presente el orden de las variables. Del mismo ejemplo anterior, la forma en que se debe dar el mintermino es:

tambien deberan darse separados por comas y terminar con ';' y <RETURN>.

Con estos minterminos, se presentaran ahora dos mapas, el de la izquierda sigue representando todos los posibles minterminos para la funcion y el de la derecha, se lleno poniendo '1' en las posiciones de los minterminos de la funcion y '0' en los que no entran en ella.

Siempre que se pida una decision al usuario (SI o NO), se deberan dar los dos caracteres.

A continuacion preguntara si existen minterminos que se combinen, en caso afirmativo, los pedira, estos deberan darse en la forma decimal ya indicada.

Validara la respuesta e indicara si es correcta o no.

En el primer caso, ademas presentara los dos mapas, indicando en el de la derecha, los minterminos que combino con el caracter 'X'.

Preguntara si aun existen minterminos por combinar, en caso afirmativo, repetira el proceso anterior; de otra manera pedira saber si hubo minterminos que no se pudieron combinar, si los hay, se debera dar 1 por vez en forma decimal acompañado al final por ';'.

Al terminar de cubrir todos los minterminos, se presentara la solucion que obtuvo el usuario indicandole si es optima o no. Si no lo es se tiene la opcion de tratar de llegar a ella reinicializando el proceso, o bien, solo de conocer la solucion optima.

La solucion sera mostrada en funcion de las variables del mapa, todas ellas seran acompañadas de "0" si la variable es negada o de "1", si no lo es.

Si no se reiniciliza el proceso, el programa termina en este momento.

## V. QUINE MC.CLUSKEY

### V.1 ACCIONES

En este capítulo se presenta el programa que guía al usuario en la minimización de funciones usando propiamente el método de Quine - Mc.Cluskey.

Dado que con este siempre se llega a la(s) solución(es) óptima(s), no hay necesidad de validar este aspecto como sucede con Mapas de Karnaugh, así mismo, en el capítulo anterior, se indicó que el método de Mapas de Karnaugh utiliza otro método que es el de Quine-Mc.Cluskey en el algoritmo de solución.

El diagrama jerárquico de este programa se muestra en la Fig. V.1.A

El programa principal organiza el uso de las subrutinas que utiliza para pedir al usuario la función a minimizar, validarla y resolverla. Además valida las proposiciones que hace el usuario para combinar los cubos de los grupos que se le presentan. Finalmente, le muestra la solución a la que se llega indicándole paso a paso la forma de llegar a ella.

Varias de las subrutinas usadas por este programa son semejantes a las del programa de Mapas de Karnaugh y se listan a continuación. Para informarse de la función de ellas, puede consultarse el capítulo anterior.

- TEXTO
- EXPONENTE
- RAIZ
- ACONJUNTO
- SECUNDARIOS
- SOLUMAQ

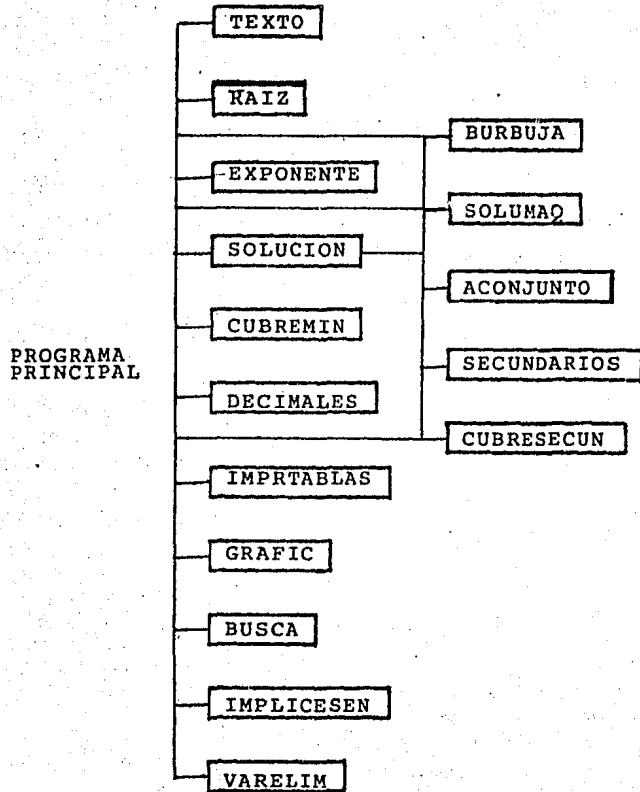


FIG. V.1.A DIAGRAMA JERARQUICO



- BURBUJA
- DECIMALES

La primera presenta siempre una introduccion al metodo, y las demas leen, validan y resuelven la funcion.

A continuacion se presentan las subrutinas que utiliza este programa pero que no necesita el anterior. Sus parametros se describen de la siguiente manera:

NOMBRE DEL PARAMETRO: TIPO <CONTENIDO>

- SOLUCION

En este programa esta subrutina sigue siendo fundamental por ser en ella donde se minimiza la funcion. El algoritmo usado se describe en el Anexo A. Solo se modifico el uso del arreglo COGITOS para mostrar posteriormente al usuario la diferencia de los cubos que se combinan ademas de las marcas de "C" (se combino), "\*" (no se combino). Se auxilia de la subrutina CUBRESECUN para realizar su cometido.

- CURRESECUN

La identificacion de implicantes primos secundarios la realiza en forma independiente en esta subrutina con el objeto de mostrar al usuario la forma de hacerlo. Es por esta razon que quedo fuera de SOLUCION.

- CUBREMIN

Variables de entrada

CUBIERTOS: caracter vectorial <matriz de marcas de la tabla de implicantes primos>  
 NIMPL: entero escalar <direcciona el ultimo renglon de la tabla de implicantes primos>  
 NDIGIS: entero escalar <numero de cubos-0>  
 DIGS: entero escalar <cubos-0>

Valida el ultimo renglon de la tabla de implicantes primos que propone el usuario, comparandolo con el que ya obtuvo de la solucion (CUBIERTOS[NIMPL,\*]), cuyas columnas son los cubos-0 de la funcion.

## - IMPRTABLAS

## Variables de entrada

CUBRE: caracter vectorial <matriz de marcas de la tabla de implicantes primos>  
 DIGS: entero escalar <cubos-0>  
 MINT: caracter vectorial <nombre asociado a cada implicante>  
 NDIGS: entero escalar <numero de cubos-0>  
 NIMPRENG: entero escalar <numero de cubos-0 que contiene cada implicante>  
 NIMPS: entero escalar <numero de implicantes primos>  
 ULCUBRE: caracter vectorial <vector que cubre el usuario, reflejo del ultimo renglon de la tabla de implicantes primos>

Grafica las tablas de implicantes primos

## - GRAFIC

## Variables de entrada

CONT: entero escalar <numero de renglon de la matriz que contiene los cubos que en ese momento analiza el usuario>  
 CONTG: entero escalar <un numero que se asigna al grupo que se analiza>  
 NGRUP: entero escalar <numero de grupos que se van a imprimir>

Grafica los grupos que analiza en ese momento el usuario.

## - IMPLICESEN

## Variables de entrada

CUBO: caracter vectorial <nombre asociado a cada implicante>  
 MARCS: caracter vectorial <marcas de implicantes esenciales>  
 NIMP: entero escalar <numero de implicantes primos>

Lee y valida las proposiciones del usuario para encontrar los implicantes primos esenciales verificando las marcas (MARCS) de todos los implicantes (NIMP). CUBO se emplea para verificar si los implicantes proporcionados por el usuario estan marcados como esenciales o no.

**- VARELIM**

Variables de entrada

**CUBO:** caracter escalar <nombre del implicante del que se van a eliminar las variables que se minimizaron durante el proceso>

**REGLON:** entero escalar <renglon de la matriz de soluciones>

Variable de salida

**CONT1:** entero escalar <contador de caracteres del renglon de la matriz de soluciones>

Esta subrutina sirve para presentar los implicantes que forman la solucion mostrando las variables que se eliminaron y las que no, con su valor binario asociado.

### V.1.B. FORTRAN 77

El segundo desarrollo del sistema que guía al usuario en la minimización de funciones por el método de Quine Mc.Cluskey se presenta a continuación.

En la figura V.1.B.1 se muestra el Diagrama Jerárquico del programa y en la figura V.1.B.2, un diagrama en el que se indica la secuencia de acción de cada subrutina.

En seguida se presentan las subrutinas que forman este programa, algunas de ellas, son similares a las descritas en el capítulo IV.1.B, por lo cual serán omitidas.

- NUMVAR

- DECIM

- BINARI

- SOLUC (Anexo B)

- MINTER

Variables de entrada: IDD, M, IFI, N

Despliega por terminal los minterminos proporcionados por el usuario, adicionando el equivalente binario de cada uno.

Mensajes para comunicarse con el usuario:

- Estos son los minterminos que proporcionaste, con su equivalente binario

- CUBOSO

Variables de entrada: IDD, M, IN22, ILON2, IMP, IIM, IFIN, N

Muestra al usuario, en tres etapas, la forma en que se agrupan los minterminos o cubos-0 de la función.

1o. Muestra los minterminos en su equivalente binario, agrupados por el número de "1" que contienen en esa representación.

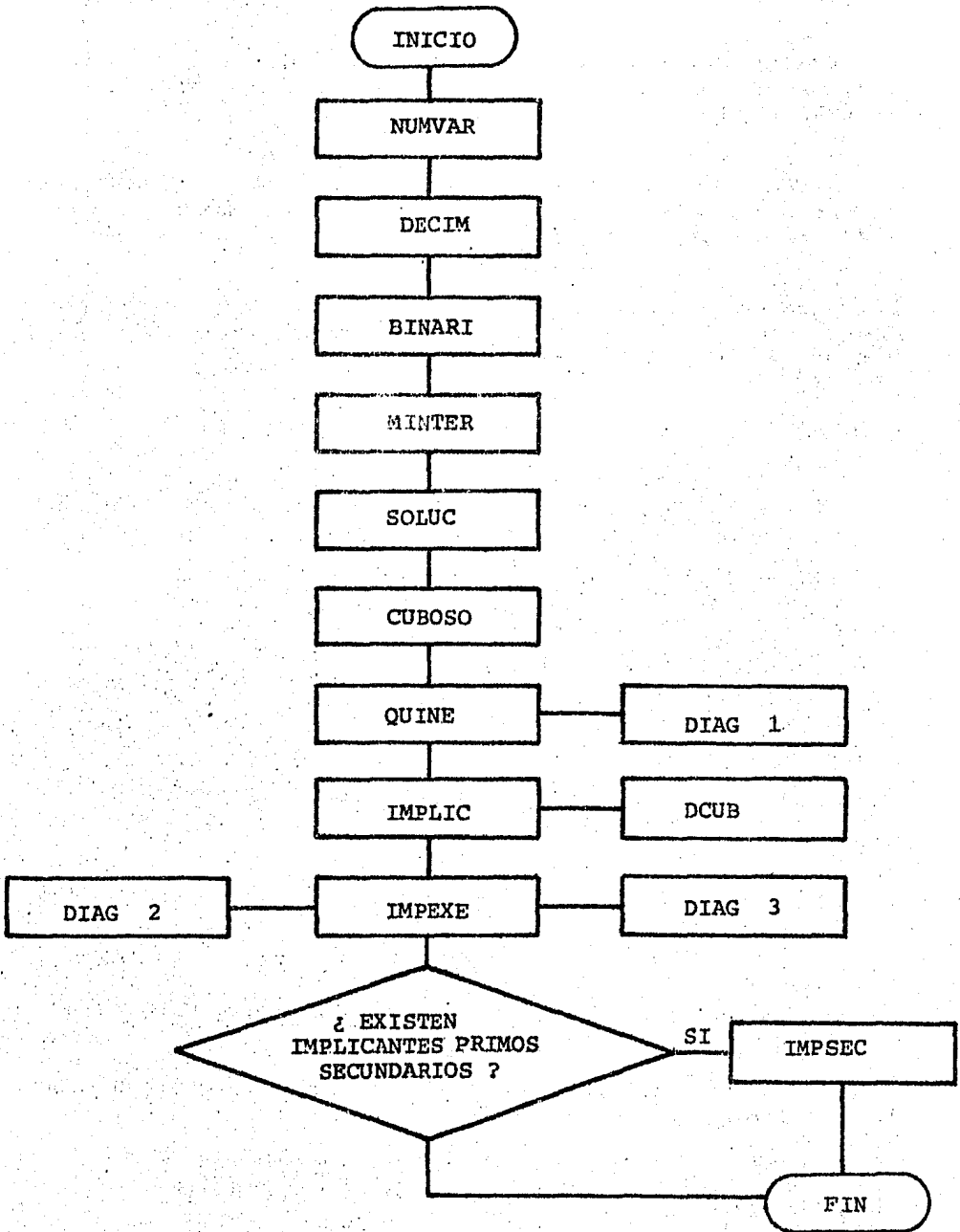


FIG. U.1.B.1

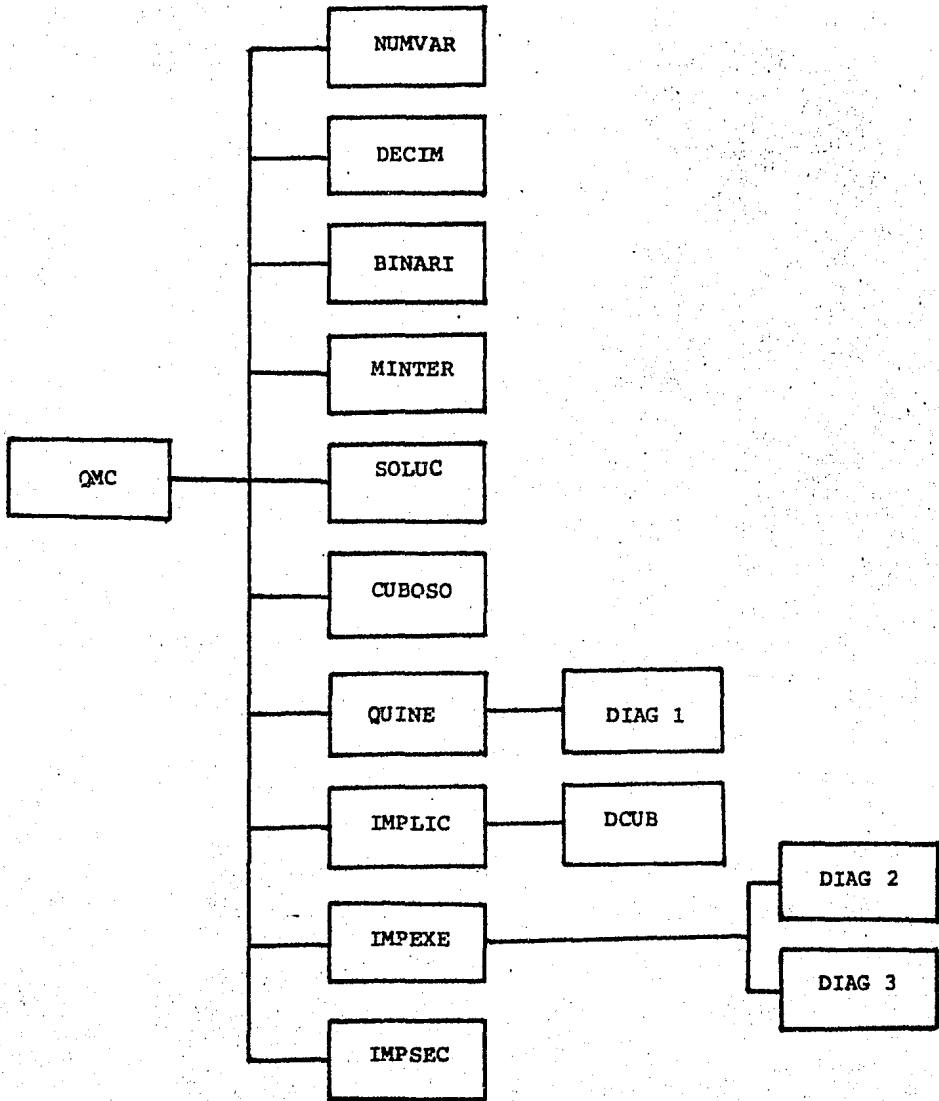


FIG. U.1.B.2

2o. Al recibir una señal para continuar, agrega el equivalente decimal de cada mintermino, inmediatamente despues del correspondiente en la representacion binaria.

3o. Al recibir nuevamente la señal de continuacion, borra el equivalente binario de los minterminos agrupados, y permanece solo el equivalente decimal.

Mensajes para comunicarse con el usuario:

- Estos son los minterminos en su representacion binaria agrupados segun el numero de unos en esa representacion
- Ahora se agrega el equivalente decimal de los minterminos
- De aqui en adelante solo se trabaja con la representacion decimal

- QUINE

Variables de entrada: N, IMP, IDD, IVD, IN22, ILON2, LIN, LIM, IMM, MAT

Guia al usuario en la formacion de cubos de orden superior a cero.

La siguiente serie de pasos es repetitiva hasta que los grupos de cubos de un cierto orden ya no se combinan para formar cubos de orden inmediato superior, o cuando solo existe un grupo de cubos de un determinado orden.

Paso 1.- Se muestran todos los grupos de cubos del orden actual

Paso 2.- Se presentan pares de grupos adyacentes hasta cubrir el ultimo grupo.

En cada par de grupos y hasta que el usuario indica que ya no existen cubos que se combinan:

- El usuario propone un cubo de cada grupo, como cubos que se combinan.
- Se valida la propuesta y se emite un diagnostico.

Si los cubos se combinan:

Ambos cubos se marcan con el caracter "C" y en un grupo adicional se muestra el cubo de orden inmediato superior a que dio lugar la combinacion.

Paso 3.- Al terminar la comparacion de grupos, se muestran nuevamente todos los grupos de cubos de este orden, pero ahora los cubos que se combinaron estaran marcados con el caracter "C".

Mensajes para comunicarse con el usuario:

Los mensajes de diagnostico son emitidos por la subrutina DIAG1.

- DIAG1

Variables de entrada: ICON, JCON, IPASO, IMEN, JMEN, IPO, JPD  
Variable de salida: IND1

Valida la propuesta de combinacion de cubos hecha por el usuario y emite un diagnostico.

Mensajes con los que se comunica con el usuario:

- Mas de dos cubos señalados; marca solo dos
- Correcto, los cubos de estos grupos no se combinan
- Existen cubos que se combinan, marcalos
- Señalaste dos cubos del grupo de la izquierda
- Señalaste dos cubos del grupo de la derecha
- El cubo de la izquierda ya lo analizaste antes
- El cubo de la izquierda no es el que debes comparar
- Existe un par de cubos que debes combinar antes, identificalos
- Bien, estos cubos se combinan
- Los cubos que propones no se combinan, verifica

- IMPLIC

Variables de entrada: TAB, IVD, IDD, II, N

Muestra los implicantes primos que se obtuvieron, indicando a que clase de cubos pertenecen.

Mensajes para comunicarse con el usuario:

- Estos son los implicantes primos que obtuviste (cubos que no se combinaron)

- DCUB

Variables de entrada: ICUB, IDIR, KC, IJ, IVD, IDD, N, TABB, A

Edita los implicantes primos, adicionando parentesis, comas y las potencias de dos que indican la variable eliminada.

- IMPEXE

Variables de entrada: ID, IVD, TAB, II, IFI, IVEC

Guia al usuario en la construccion de la tabla de implicantes primos, a fin de que identifique los implicantes primos esenciales.



### Descripcion del proceso:

- 1) Se presenta el primer renglon de la tabla, conteniendo en cada columna un mintermino.
- 2) Se presenta un nuevo renglon de la tabla, identificandolo con una letra del alfabeto.

Con la misma letra que se identifica el renglon, se identifica el implicante primo que se le muestra al usuario.

Se pide al usuario que anote marcas en las columnas que corresponden a los minterminos que forman al implicante primo mostrado.

Se valida la respuesta y en caso de error, se emite un mensaje.

El proceso descrito en el paso 2), se repite hasta cubrir todos los implicantes primos.

Los implicantes primos presentados se agrupan de acuerdo al costo (numero de cubos-0), indicando la agrupacion por medio de una linea de guiones al cambiar a otro nivel de costo de implicantes.

- 3) A continuacion se muestra una columna del lado izquierdo de la tabla, que se usara para identificar a los implicantes primos esenciales.

El usuario examina las columnas de minterminos para identificar aquellos que contienen una sola marca, esto indica que el implicante primo al que pertenece, es un implicante primo esencial y se marca con un "\*" en la columna de la extrema derecha.

- 4) Al terminar de identificar a los implicantes primos esenciales, se marcan los minterminos contenidos en ellos, en el renglon base.

Si el renglon base tiene marcas en todas las columnas, entonces todos los cubos-0 estan contenidos en los implicantes esenciales y la suma de estos es la realizacion minima de suma de productos.

En caso de no quedar cubiertos todos los minterminos por los implicantes primos esenciales, se presenta una tabla reducida en la que el usuario podra identificar a los implicantes primos secundarios al aplicar el dominio.

Tambien se muestra la aplicacion de la tecnica de Petrick para obtener los implicantes primos que formen soluciones optimas.

Mensajes para comunicarse con el usuario:

Los mensajes que diagnostican la respuesta del usuario, se emiten a traves de las subrutinas DIAG2 y DIAG3.

- DIAG2

Variables de entrada: IFI, IRES, MM, ILON, IRP, INDI

Valida las marcas que hace el usuario en los minterminos que pertenecen a un implicante primo, en las columnas correspondientes.

Mensajes para comunicarse con el usuario:

- Sobrepasa los minterminos marcados a los minterminos del implicante
- Respuesta nula, marca los minterminos
- No señalaste correctamente la tabla

- DIAG3

Variables de entrada: II, IRP, IRES, MM, IST, INDI

Valida las marcas que hace el usuario para identificar los implicantes esenciales.

Mensajes para comunicarse con el usuario:

- Falta marcar algunos de los implicantes primos esenciales
- Estan marcados mas implicantes que los implicantes esenciales
- Algunos de los implicantes primos estan incorrectamente marcados.

```

#
#FILE (VRB3)TESIS/QUINE ON UNAM2
100 PROGRAM QUINE(INPUT,OUTPUT);
200 (* ESTE PROGRAMA RESUELVE FUNCIONES A TRAVES DE QUINE MC.
300 Y GUIA A EL USUARIO A RESOLVERLO, VALIDANDO TODAS SUS SUGERENCIAS.
400 EL PROGRAMA USA PARA SU SOLUCION, PROCEDURES CUYA FUNCION SE EXPLICA
500 A CONTINUACION:
600
700 - TEXTO: CONTIENE UNA PEQUEÑA INTRODUCCION AL METODO.
800 - EXPONENTE: OBTIENE LA POTENCIA DE UN NUMERO EN BASE 2.
900 - RAIZ: OBTIENE LA RAIZ N(ESIMA) DE UN NUMERO.
1000 - ACONJUNTO: SI HAY IMPLICANTES SECUNDARIOS, ESTA SUBROUTINA ES
1100 Y LOS OBTIENE A TRAVES DEL METODO DE PETRICK.
1200 - SECUNDARIOS : DEFINE LA TABLA DE IMPLICANTES PRIMOS SECUNDARIOS
1300 - SOLUMAQ : RESUELVE A TRAVES DE Q.M, A PARTIR DE TOMAR LOS CUBOS
1400 CUYA DIFERENCIA SEA POTENCIA DE 2 Y OBTIENE LA
1500 O SOLUCIONES POSIBLES
1600 - CUBRESECUN: REvisa SI EXISTEN IMPLICANTES PRIMOS SECUNDARIOS
1700 Y OBTIENE LA SOLUCION FINAL ANEXANDOLOs.
1800 - CUBREMIN: VALIDA LAS RESPUESTAS DEL USUARIO PARA LLENAR EL
1900 ULTIMO RENGLON DE LA TABLA DE IMPLICANTES PRIMOS
2000 ESENCIALES.
2100 - BURBUJA : ORDENA LOS MINTERMINOS POR EL METODO DE BURBUJA
2200 - SOLUCION : PROCEDURE FUNDAMENTAL DEL PROGRAMA YA QUE ES DONDE SE
2300 RESUELVE EL PROGRAMA LA FUNCION BAJO LA SIGUIENTE
2400 SECUENCIA:
2500 SEPARA LOS MINTERMINOS POR EL NUMERO DE UNOS DE SU REPRESENTACION
2600 BINARIA ALMACENANDOLOs EN SOLUCI.ARREGLO Y EL NUMERO DE ELLOS EN
2700 SOLUCI.NELEM
2800 HACE LA COMBINACION DE LOS MINTERMINOS CUYA DIFERENCIA SEA POTEN
2900 CIA DE 2, MARCANDO SIMULTANEAMENTE LOS QUE SE COMBINAN Y LOS QUE
3000 NO, EN EL PRIMER CASO LOS MARCA CON UNA "C", EN CASO CONTRARIO,
3100 CON UN "*".
3200 SOLUCI.NELEM ESTA ORGANIZADO DE TAL MANERA QUE LOS RENGLONES IN-
3300 DICAN EL NUMERO DE UNOS QUE TIENEN LOS CUBOS DE LOS RENGLONES
3400 1-10 Y A PARTIR DEL 11 EN ADELANTE, LAS DECENAS INDICAN EL NUMERO
3500 DE ELEMENTOS DEL CUBO Y LAS UNIDADES EL RENGLON DEL CUBO QUE SE
3600 COMBINO CON EL RENGLON SIGUIENTE. EJEMPLO :
3700 EL RENGLON 13 INDICA:
3800 2**1 : QUIERE DECIR QUE TIENE DOS ELEMENTOS POR CUBO
3900 3 : QUIERE DECIR QUE SE COMBINARON UN CUBO DEL RENGLON 3 CON
4000 - DECIMALES : VERIFICA LA ENTRADA DE LOS MINTERMINOS DE LA
4100 FUNCION EN FORMA DECIMAL
4200 - IMPRTABLAS: GRAFICA LAS TABLAS DE IMPLICANTES PRIMOS.
4300 - GRAFIC: GRAFICA LOS CUBOS DE LOS GRUPOS QUE VA ANALIZANDO EL
4400 USUARIO
4500 - BUSCA : USADO EN LA SOLUCION DEL USUARIO PARA BUSCAR LOS CUBOS
4600 PROPUESTOS POR EL EN EL ARREGLO SOLUCI.ARREGLO
4700 - IMPLICESEN: VALIDA LA RESPUESTA DEL USUARIO PARA ENCONTRAR
4800 LOS IMPLICANTES PRIMOS ESENCIALES.
4900 - VARELIM: MARCA CON X LAS VARIABLES ELIMINADAS DE LOS CUBOS
5000 QUE INTEGRAN LA(S) SOLUCION(ES) OPTIMA(S).
5100 ESTAS SON TODOS LOS PROCEDURES USADOS EN EL PROGRAMA
5200 EL PROGRAMA PRINCIPAL COMANDA TODA LA SOLUCION DEL ALUMNO
5300 *)
5400 TYPE CHARACTER = ARRAY[0..32] OF CHAR;
5500 CARAC = ARRAY[0..200] OF CHAR;
5600 MATRICES = ARRAY[0..70,0..65] OF INTEGER;
5700 MATRICESC = ARRAY[0..70,0..64] OF CHAR;
5800 VECTOR = ARRAY[-1..100] OF INTEGER;
5900 MATENT = ARRAY[0..30,0..30] OF INTEGER;
5901

```

```

5902
5903
5904
6000 MATCARAC = ARRAY[0..72,0..150] OF CHAR;
6100 STRING = ARRAY [0..6] OF CHAR;
6200 NUMEROS = SET OF INTEGER;
6300 LETRAS = RECORD
6400     LETRA : STRING;
6500     NOASOC : VECTOR;
6600 END;
6700 SOLUC = RECORD
6800     ARREGLO : MATRICES;
6900     ARRMARCA : MATRICESC;
7000     NOCUBIERTO : MATCARAC;
7100     MATSOL,IMPSEC,TABIMP: MATENT;
7200     SOLTOTAL,CUBIERTO : MATCARAC;
7300     MARC : CHARACTER;
7400     NLEM,NMATSOL,NSQLTOTAL,NTABSEC,NTABIMP : VECTOR;
7500 END;
7600 VAR
7700 I1,K,J1,NI,NJ,I,J,II,CONT,CONT1,CONT2,CONT3,CONT4,CONT5,CONT6,CONTAB,
7800 CONT7,CONT10, (* CONTADORES *)
7900 NVARIA (* NUMERO DE VARIABLES DE LA FUNCION *),
8000 NDIGITOS (* TOTAL DE MINTERMINOS *),
8100 NIMPSEC (* TOTAL DE IMPLICANTES SECUNDARIOS *),
8200 NIMPLIC (* TOTAL DE IMPLICANTES PRIMOS *),
8300 NDI0SEC (* *),
8400 NCASILLAS (* NUMERO DE MINTERMINOS PARA SOLUCIONES PARCIALES *),
8500 TOTSOL (* TOTAL DE SOLUCIONES FINALES *),
8600 POTENCIA (* NUMERO MAXIMO DE MINTERMINOS PARA NVARIA *),
8700 T1 (* NUMERO DE 1S DE LOS MINTERMINOS *),
8800 T (* DIRECCIONAMIENTO DE RENGLONES *),
8900 MAX (* *),
9000 NSECUN,
9100 DECIMAL (* MINTERMINO EN FORMA DECIMAL *),
9200 DIFERENCIA,DIFERENCIAL (* RESULTADOS DE DIFERENCIA ENTRE MINTERMI
9300 NOS PARA REVISAR SI ES POTENCIA DE 2 *)
9400 : INTEGER;
9500 ARRVAR1,ARRVAR2 (* POSICION DE LAS VARIABLES POR PESO *),
9600 CAMBIO,METODO (* LEE DECISIONES DEL USUARIO DE SI O NO *)
9700 : STRING;
9800 SOLUCIONES : MATCARAC; (* TODAS LAS SOLUCIONES DE LA
9900 MAQUINA *)
10000 TABALUM : MATENT;
10100 SIMBOLO : LETRAS;
10200 SOLUCI : SOLUC;
10300 MARCAMIN, (* MARCA IMPLICANTES SECUNDARIOS QUE DAN SOL. OPTIMA*)
10400 IMPS, (* IMPLICANTES PRIMOS QUE PROPONE EL USUARIO *)
10500 CUBALUM, (* ULTIMO RENGLON DE LA TABLA DE IMPLICANTES PRIMOS
10600 QUE LLENA EL USUARIO Y SE VALIDA *)
10700 MARCAS, (* MARCA DE IMPLICANTES PRIMOS PROPUESTOS POR EL
10800 USUARIO *)
10900 CUBO, (* NOMBRE QUE ASIGNA EL PROGRAMA A LOS CUBOS DE LA
11000 TABLA DE IMPLICANTES PRIMOS ORIGINAL *)
11100 CUBOS, (* NOMBRE QUE ASIGNA EL PROGRAMA A LOS CUBOS DE LA
11200 TABLA DE IMPLICANTES SECUNDARIOS *)
11300 CASDIO (* SE MARCAN LOS CUBOS QUE CUBRE EL USUARIO *)
11400 : CHARACTER;
11500 NSOLUCIONES (* NUMERO TOTAL DE SOLUCIONES *),
11600 NCONJEXTEN (* NUMERO DE POSIBLES SOLUCIONES PARA IMPL. SEC *),
11700 NCONJUNTO ,
11800 PBURB (* CUBOS ENVIADOS A ORDENARSE *),
11900 DIGITOS (* MINTERMINOS *),
12000 DIGSEC (* MINTERMINOS DE LOS IMPLICANTES SECUNDARIOS *),
12100 NTABALUM (* NUMERO DE REDUCCIONES DEL ALUMNO *),
12101

```

12102  
12103  
12200  
12300  
12400  
12500  
12600  
12700  
12800  
12900  
13000  
13100  
13200  
13300  
13400  
13500  
13600  
13700  
13800  
13900  
14000  
14100  
14200  
14300  
14400  
14500  
14600  
14700  
14800  
14900  
15000  
15100  
15200  
15300  
15400  
15500  
15600  
15700  
15800  
15900  
16000  
16100  
16200  
16300  
16400  
16500  
16600  
16700  
16800  
16900  
17000  
17100  
17200  
17300  
17400  
17500  
17600  
17700  
17800  
17900  
18000  
18100  
18200  
18300  
18301  
18302

```
CASI (* AUXILIAR, TOMA LOS MISMOS VALORES DE CASILLAS*);
CASILLAS (* CUBOS MANEJADOS POR EL ALUMNO *)
: VECTOR;
BINARIO : ARRAY [0..30,0..30] OF CHAR;
UND (* MINTERMINOS CON 1 UND EN SU REPRESENTACION BINARIA *);
CONJ1,CONJ2 (* CONJUNTOS AUXILIARES PARA VERIFICACION DE INTERSECCIONES O IGUALDAD DE CUBOS *)
: NUMERDS;
CONJUNTO (* ARREGLO DE CONJUNTOS DE CUBOS *)
: ARRAY [0..50,0..150] OF NUMEROS;
CDIGITOS : ARRAY[0..70,0..1,0..72] OF CHAR;
SQLALUM : CARAC;
AUX : ARRAY[0..6] OF CHAR;
OPCION,AUX1 : CHAR;
BANDERA,BANDERA1,BANDERA2,BANDTOTAL : BOOLEAN; (* BANDERAS *)

PROCEDURE TEXTO;
BEGIN
  WRITELN('LOS MINTERMINOS DE UNA FUNCION SON LLAMADOS CUBOS-0.');
```

```
  WRITELN('EL METODO CONSISTE EN UNA BUSQUEDA EXHAUSTIVA DE TODAS',
    ' LAS COMBINACIONES');
```

```
  WRITELN('POSIBLES DE LOS CUBOS-N PARA FORMAR LOS CUBOS-(N+1) Y ',
    ' LUEGO SELECCIONAR');
```

```
  WRITELN('LA COMBINACION MINIMA DE CUBOS PARA REALIZAR LA FUNCION.');
```

```
  WRITELN('SE INICIA OBTENIENDO LA REPRESENTACION BINARIA DE LOS ',
    ' CUBOS-0 Y SE');
```

```
  WRITELN('AGRUPAN EN FUNCION DEL NUMERO DE '1 S.');
```

```
  WRITELN('LOS CUBOS-1 SE OBTIENEN COMPARANDO GRUPOS CONSECUTIVOS',
    ' POR PARES.EJEMPLO:');
```

```
    GRUPO 1          GRUPO 2');
```

```
    001 1          011 3');
```

```
    010 2          110 6');
```

```
  WRITELN('SI COMPARAMOS UN CUBO DEL GRUPO 2 CON OTRO DEL GRUPO ',
    ' 1 Y LA DIFERENCIA');
```

```
  WRITELN('ES POTENCIA DE 2 HEMOS OBTENIDO UN CUBO-1.');
```

```
  WRITELN(' CUBO-0(GRUPO 2) - CUBO-0(GRUPO 1) = 3 - 2 = 1');
```

```
  WRITELN('OBSERVA QUE SE HA HECHO LA DIFERENCIA DEL GRUPO MAYOR',
    ' - GRUPO MENOR, LO');
```

```
  WRITELN('CONTRARIO NO ES CIERTO.');
```

```
  WRITELN('ESTE PROCEDIMIENTO SE CONTINUA HASTA CUBRIR TODOS LOS ',
    ' CUBOS DE TODOS LOS');
```

```
  WRITELN('GRUPOS.');
```

```
  WRITELN('PARA OBTENER LOS CUBOS-2 HASTA CUBOS-N DEBES OBSERVAR ',
    ' LA MISMA CONDICION');
```

```
  WRITELN('DE DIFERENCIA DE POTENCIA DE 2 ENTRE LOS CUBOS DE CADA',
    ' GRUPO CONSECUTIVO');
```

```
  WRITELN('EN SUS ELEMENTOS CORRESPONDIENTES Y ADEMAS QUE ',
    ' LA(S) DIFERENCIA(S)');
```

```
  WRITELN('ANTERIOR(ES), LA CUAL TE PRESENTARE EN ESTE PROGRAMA ',
    ' ENTRE PARENTESIS');
```

```
  WRITELN('SEA LA MISMA.');
```

```
  WRITELN('CONTINUA...'); READLN;
```

```
  WRITELN('SI ALGUN CUBO DE ALGUN GRUPO NO SE COMBINA, LO MARCAR',
    ' AS, ESTE FORMARA');
```

```
  WRITELN('PARTE DE LOS IMPLICANTES PRIMOS.');
```

```
  WRITELN('LOS IMPLICANTES PRIMOS SON TODOS AQUELLOS CUBOS QUE ',
    ' DURANTE EL PROCESO');
```

```
  WRITELN('DE COMPARACION NO LOGRARON COMBINARSE, DE ENTRE ELLOS',
    ' SE SELECCIONARAN');
```

```
  WRITELN('AQUELLOS QUE NOS DARAN LA SOLUCION MINIMA.');
```

```
  WRITELN('ESTOS IMPLICANTES SE ORDENAN EN UNA TABLA EN FUNCION ',
    ' DE LA CANTIDAD DE');
```

```

18303
18304
18400 WRITELN('CUBOS-O QUE CONTENGAN.');
```

18500 WRITELN('ESTA TABLA TENDRA EN SUS RENGLONES LOS IMPLICANTES OR',  
18600 'DENADOS EN FUNCION');

18700 WRITELN('DE LA CANTIDAD DE CUBOS-O Y EN SUS COLUMNAS TODOS LOS',  
18800 'CUBOS-O.');

18900 WRITELN('DE CADA IMPLICANTE SE BUSCARAN EN LAS COLUMNAS,LOS ',  
19000 'CUBOS-O QUE LO FORMAN');

19100 WRITELN('Y SE MARCARA LA COLUMNA CORRESPONDIENTE.');

19200 WRITELN('QUE LE PERTENEZCAN.');

19300 WRITELN('EL PASO SIGUIENTE ES VERIFICAR CADA COLUMNA QUE CONTENGA ',  
19400 'SOLO UNA MARCA,');

19500 WRITELN('CUANDO SE ENCUENTREN, SE MARCARAN EN UN RENGLON ADICIONAL',  
19600 'AL IGUAL QUE');

19700 WRITELN('TODAS AQUELLAS QUE ESTEN MARCADAS EN EL RENGLON QUE SE EN',  
19800 'CONTRO SOLO UNA');

19900 WRITELN('MARCA, ESTE IMPLICANTE SE CONVIERTE EN ESCENCIAL Y FORMA',  
20000 'RA PARTE DE LA');

20100 WRITELN('SOLUCION.');

20200 WRITELN('SI AL TERMINAR DE REVISAR TODAS LAS COLUMNAS Y MAR',  
20300 'CADO LOS IMPLICANTES');

20400 WRITELN('ESCENCIALES HAY COLUMNAS QUE NO SE MARCARON, PASAREMOS ',  
20500 'ESTAS COLUMNAS A');

20600 WRITELN('OTRA TABLA ASI COMO LOS IMPLICANTES QUE LAS CONTENGAN.');

20700 WRITELN('CONTINUA...'); READLN;

20800 WRITELN('EXISTEN DIVERSOS METODOS PARA RESOLVER ESTA ULTIMA TABLA',  
20900 ', EN ESTE PROGRA');

21000 WRITELN('MA LO HAREMOS A TRAVES DE UN METODO LLAMADO DE PETRICK, EL ',  
21100 'CUAL CONSISTE');

21200 WRITELN('EN TOMAR LOS IMPLICANTES QUE FORHAN ESTA TABLA E INTER',  
21300 'PRETARLAS COMO');

21400 WRITELN('VARIABLES BOOLEANAS, ESTO ES TOMARAN VALORES CIERTO O ',  
21500 'FALSO, DEPENDIENDO');

21600 WRITELN('DE SI SE ESCOGE O NO PARA EL ANALISIS.');

21700 WRITELN('A PARTIR DE LA PRIMERA COLUMNA SE DEBEN SELECCIONAR ',  
21800 'CUALQUIERA DE LOS');

21900 WRITELN('IMPLICANTES QUE TENGAN MARCA EN ELLA PARA CUBRIRLO ',  
22000 'DE MANERA QUE SI');

22100 WRITELN('REPRESENTAMOS CADA IMPLICANTE POR UNA LETRA Y a ES EL ',  
22200 'PRIMER IMPLICANTE');

22300 WRITELN('Y d EL TERCERO DE LA TABLA SECUNDARIA, Y AMBOS TIENEN ',  
22400 'MARCADOS EN ALGUNA');

22500 WRITELN('DE LAS COLUMNAS, SE EXPRESARAN POR:');

22600 WRITELN(' (a+d)=1');

22700 WRITELN('YA QUE CUALQUIERA DE LOS DOS CUBRE ESE CUBO. DEBIDO A ',  
22800 'TODOS ELLOS DEBEN');

22900 WRITELN('CUBRIRSE,SE UNIRAN POR EL CONECTOR 'Y', ESTO ES,',  
23000 'SI OTRA COLUMNA CONTIENE');

23100 WRITELN('POR EJEMPLO LOS CUBOS c y f MARCADOS, SE TENDRA LA ',  
23200 'EXPRESION:');

23300 WRITELN(' (a+b)(c+f)=1');

23400 WRITELN('ESTE PROCESO SE SIGUE HASTA INVOLUCRAR TODAS LAS COLUM',  
23500 'NAS.');

23600 WRITELN('ENSEGUIDA SE REALIZARAN LOS PRODUCTOS INDICADOS Y SE ',  
23700 'SIMPLIFICARA LA');

23800 WRITELN('EXPRESION RESULTANTE POR MEDIO DE LOS TEOREMAS FUNDA',  
23900 'MENTALES DEL ALGEBRA');

24000 WRITELN('DE BOOLE, POR LO MENOS ALGUNO DE LOS PRODUCTOS DEBE ',  
24100 'SER CIERTO (1), DE');

24200 WRITELN('ESTOS SE SELECCIONA ALGUNO CUYO COSTO SEA MENOR, ESTO ES,',  
24300 'EL NUMERO DE');

24400 WRITELN('ENTRADAS ES MAS REDUCIDO.');

24500 WRITELN('LOS CUBOS QUE CONTENGA ESTE PRODUCTO MAS LOS IMPLICANTES ',  
24501  
24502

```

24503
24504
24600      'DE LA PRIMERA');
24700  WRITELN('TABLA, DAN UNA SOLUCION MINIMA. ');
24800  WRITELN('CONTINUA... '); READLN;
24900  WRITE('AHORA BIEN ');
25000  END; (* TEXTO *)
25100
25200  FUNCTION EXPONENTE(N:INTEGER) : INTEGER;
25300  BEGIN
25400      EXPONENTE:=ROUND(EXP(N* LN(2)));
25500  END;
25600
25700  FUNCTION RAIZ(NUMERO:INTEGER) : INTEGER;
25800  VAR I,COCIENTE : INTEGER;
25900  BEGIN
26000      COCIENTE:=0;
26100      I:=NUMERO;
26200      WHILE I > 1 DO
26300          BEGIN
26400              I:=I DIV 2;
26500              COCIENTE:=COCIENTE+1;
26600          END;
26700      RAIZ:=COCIENTE;
26800  END; (* RAIZ *)
26900
27000  PROCEDURE ACONJUNTO (NDIGS,NIMP:INTEGER;NN:VECTOR;CUBRE:MATCARAC;
27100  VAR CONT:INTEGER);
27200  CONST ELIMINA=1000;
27300  VAR
27400      BANDERA:BOOLEAN;
27500      TOPE:VECTOR;
27600      J1,I,J,K,JJ,K1,K2,POSICION,RELATIVO,MINIMO : INTEGER;
27700  BEGIN
27800      K:=-1;
27900      FOR I:=0 TO NDIGS DO
28000          BEGIN
28100              JJ:=0;
28200              FOR J:=1 TO NIMP DO
28300                  IF CUBRE[CJ,I]='0' THEN
28400                      BEGIN
28500                          IF JJ = 0 THEN
28600                              WRITE('(',CUBOS[CJ:1]);
28700                          ELSE
28800                              WRITE('+',CUBOS[CJ:1]);
28900                          CONJUNTO[CJ,JJ]:=CJ;
29000                          JJ:=JJ+1;
29100                      END;
29200                      IF JJ > 0 THEN
29300                          WRITE(')');
29400                      NCONJUNTO[CJ]:=JJ;
29500                  END;
29600              WRITELN('=1');
29700              FOR I:=0 TO NDIGS DO
29800                  NCONJEXTENC[I]:=NCONJUNTO[CJ];
29900          (* PREPARA LOS CONJUNTOS PARA UNIRSE *)
30000          FOR I:=0 TO (NDIGS-1) DO
30100              FOR J:=(I+1) TO NDIGS DO
30200                  BEGIN
30300                      JJ:=0;
30400                      FOR K:=NCONJEXTENC[J] TO (NCONJEXTENC[J]+NCONJEXTENC[I]-1) DO
30500                          BEGIN
30600                              CONJUNTO[CJ,KJ]:=CONJUNTO[CJ,JJ];
30700                              JJ:=JJ+1;
30701
30702

```

```

30703
30704
30800     END;
30900     NCONJEXTENCJJ:=K;
31000     CONJUNTOCJ,KJ:=CELIMINA;
31100     END;
31200     CONJUNTOC0,NCONJUNTOC0J:=CELIMINA;
31300     FOR I:=0 TO NDIGS-1 DO
31400     BEGIN
31500         J:=0;
31600         WHILE CONJUNTOCI,J <> CELIMINA DO
31700         BEGIN
31800             FOR K:=J*NCONJUNTOCI+1 TO (NCONJUNTOCI+1)*(1+J) - 1 DO
31900                 CONJUNTOCI+1,KJ:=CONJUNTOCI+1,KJ+CONJUNTOCI,J;
32000                 J:=J+1;
32100         END;
32200     END;
32300     I:=0;
32400     IF K > -1 THEN
32500     BEGIN
32600         WRITELN('EFECTUANDO LAS OPERACIONES INDICADAS LLEGAMOS A:');
32700         FOR J:=0 TO K-1 DO
32800         BEGIN
32900             IF J > 0 THEN
33000             BEGIN
33100                 I:=I+1;
33200                 WRITE('+');
33300             END;
33400             FOR JJ:=1 TO NIMP DO
33500                 IF JJ IN CONJUNTOCNDIGS,JJ THEN
33600                 BEGIN
33700                     I:=I+1;
33800                     WRITE(CUBOS[JJ:1]);
33900                 END;
34000                 IF I >= 72 THEN
34100                 BEGIN
34200                     WRITELN;
34300                     I:=0;
34400                 END;
34500             END;
34600             WRITELN('=1');
34700         END;
34800         JJ:=0;
34900         FOR I:=0 TO (NCONJEXTENCNDIGS) DO
35000         BEGIN
35100             BANDERA:=FALSE;
35200             K1:=I;
35300             IF CONJUNTOCNDIGS,I <> CELIMINA THEN
35400             BEGIN
35500                 FOR J:=0 TO (NCONJEXTENCNDIGS-1) DO
35600                     IF (J <> I) AND (CONJUNTOCNDIGS,J <> CELIMINA) AND
35700                         (CONJUNTOCNDIGS,I)*CONJUNTOCNDIGS,J = CONJUNTOCNDIGS,I) THEN
35800                     BEGIN
35900                         BANDERA:=TRUE;
36000                         CONJUNTOCNDIGS,JJ:=CELIMINA;
36100                     END;
36200                 FOR J:=0 TO NCONJEXTENCNDIGS-1 DO
36300                     IF (J <> I) AND (CONJUNTOCNDIGS,J <> CELIMINA) AND
36400                         (CONJUNTOCNDIGS,J)*CONJUNTOCNDIGS,I = CONJUNTOCNDIGS,J) THEN
36500                     BEGIN
36600                         BANDERA:=TRUE;
36700                         CONJUNTOCNDIGS,I:=CELIMINA;
36800                         K1:=J;
36900                         FOR J1:=0 TO NCONJEXTENCNDIGS-1 DO
36901
36902

```



```

36903
36904
37000      IF (J1 <> K1) AND (CONJUNTOENDIGS,J1) <> (ELIMINAJ) AND
37100      (CONJUNTOENDIGS,K1)*CONJUNTOENDIGS,J1) =CONJUNTOENDIGS,K1) THEN
37200          CONJUNTOENDIGS,J1)=ELIMINAJ;
37300      END; BANDERA:=TRUE; WHILE BANDERA DO BEGIN BANDERA:=FALSE;
37400      FOR J:=0 TO (JJ-1) DO
37500          IF CONJUNTOENDIGS+1,JJ)*CONJUNTOENDIGS,K1) = CONJUNTOENDIGS,K1)
37600          THEN BEGIN
37700              FOR J1:=J TO JJ-2 DO
37800                  CONJUNTOENDIGS+1,J1)=CONJUNTOENDIGS+1,J1+1);
37900                  JJ:=JJ-1; BANDERA:=TRUE; J:=JJ;
38000          END;
38100      END;
38200      CONJUNTOENDIGS+1,JJ)=CONJUNTOENDIGS,K1);
38300      JJ:=JJ+1;
38400      IF K1 <> I THEN
38500          CONJUNTOENDIGS,K1)=ELIMINAJ;
38600      END;
38700      END;
38800      K2:=JJ-1;
38900      WRITELN('FINALMENTE, SIMPLIFICANDO, SE LLEGA A:');
39000      FOR J:=0 TO K2 DO
39100          BEGIN
39200              IF J>0 THEN
39300                  WRITE('+');
39400              FOR JJ:=1 TO NIMP DO
39500                  IF JJ IN CONJUNTOENDIGS+1,JJ THEN
39600                      WRITE('CUBROSEJ:1);
39700          END;
39800          WRITELN('=1');
39900          FOR I:=0 TO K2 DO
40000              BEGIN
40100                  J:=1;
40200                  K1:=0;
40300                  TOPEEIJ:=0;
40400                  CONJ2:=CONJUNTOENDIGS+1,I);
40500                  WHILE J<=NIMP DO
40600                      BEGIN
40700                          IF (J IN CONJ2) THEN
40800                              BEGIN
40900                                  TOPEEIJ:=TOPEEIJ+SOLUCI.NTABSECCJ);
41000                                  SOLUCI.MATSOLCI,K1)=J;
41100                                  K1:=K1+1;
41200                              END;
41300                              J:=J+1;
41400                          END;
41500                          SOLUCI.NMATSOLCI)=K1;
41600                      END;
41700      (* SE BUSCA EL CONJUNTO CON COSTO MINIMO *)
41800      FOR I:=0 TO K2 DO
41900          MARCAMINIJ:=' ';
42000          MINIMO:=TOPECOJ;
42100          POSICION:=0;
42200          FOR I:=1 TO K2 DO
42300              IF TOPEEIJ<MINIMO THEN
42400                  BEGIN
42500                      MINIMO:=TOPEEIJ;
42600                      POSICION:=I;
42700                  END;
42800          MARCAMINEPOSICIONJ:='*';
42900      (* BUSCA TODOS LOS CONJUNTOS CON EL MISMO COSTO MINIMO *)
43000      FOR I:=0 TO K2 DO
43100          IF I<>POSICION THEN
43101
43102

```

```

43103
43104
43200     IF TOPECII=MINIMO THEN
43300         MARCAMIN CII:='*';
43400         CONT:=K2;
43500 END ; (* ACONJUNTO *)
43600
43700 PROCEDURE SECUNDARIOS(J:INTEGER;VAR II:INTEGER);
43800 VAR
43900     JJ,CONT,K ; INTEGER;
44000 BEGIN
44100 (* SE OBTIENE LA TABLA DE IMPLICANTES SECUNDARIOS *)
44200     FOR K:=0 TO (SOLUCI.NTABIMPCJJ-1) DO
44300         SOLUCI.IMPSECCII,KJ:=SOLUCI.TABIMPCJ,KJ;
44400     FOR I:=0 TO NDIGITOS DO
44500         IF (SOLUCI.CUBIERTO CJ,IJ='0') AND (SOLUCI.CUBIERTO CNIMPLIC+1,IJ
44600             = ' ') THEN
44700             BEGIN
44800                 JJ:=0;
44900                 WHILE DIGITOSCII<>DIGSECCJJ DO
45000                     JJ:=JJ+1;
45100                 SOLUCI.NOCUBIERTO CII,JJJ!='0';
45200                 SOLUCI.NTABSECCIIJJ:=SOLUCI.NTABIMPCJJ;
45300                 CUBOS CIIJJ:=CUBO CJJ;
45400             END;
45500             II:=II+1;
45600 END ; (* SECUNDARIOS *)
45700
45800 PROCEDURE SOLUMAR(VAR CONT:INTEGER;NELEME N,I:INTEGER;MATRIZ;MATENT);
45900 VAR
46000     BINARYO:ARRAY[0..30,0..30] OF CHAR;
46100     LIM2,K,CONT1,CONT2,CONT3,DIR,DIR1,DECIMAL,DELTA,LIM:INTEGER;
46200     AUXILIAR ; STRING;
46300     AUX:CHAR;
46400 BEGIN
46500     DIR:=0;
46600     FOR CONT1:=0 TO (NELEME N-1) DO
46700         BEGIN
46800             DECIMAL:=MATRIZ C I,CONT1;
46900             FOR CONT2:=0 TO (NVARIA-1) DO
47000                 BEGIN
47100                     IF (DECIMAL MOD 2) = 1 THEN
47200                         BINARYO C DIR,CONT2:='1'
47300                     ELSE
47400                         BINARYO C DIR,CONT2:='0';
47500                     DECIMAL:=DECIMAL DIV 2;
47600                 END;
47700                 DIR:=DIR+1;
47800             END;
47900             DELTA:=2;
48000             LIM2:=NELEME N DIV 2;
48100             LIM:=LIM2;
48200             IF NELEME N > 1 THEN
48300                 FOR CONT3:=1 TO LIM2 DO
48400                     BEGIN
48500                         WHILE NOT(CONT3 IN UNO) DO
48600                             CONT3:=CONT3+1;
48700                         FOR CONT1:=0 TO (LIM-1) DO
48800                             BEGIN
48900                                 DIR1:=CONT1*DELTA;
49000                                 DIR:=DIR1+CONT3;
49100                                 FOR CONT2:=0 TO (NVARIA-1) DO
49200                                     IF (BINARYO C DIR,CONT2) <> (BINARYO C DIR1,CONT2) THEN
49300                                         BINARYO C DIR1,CONT2:='X';
49301
49302

```

```

49303
49304
49400     END;
49500     LIM:=LIM DIV 2;
49600     DELTA:=DELTA*2;
49700 END;
49800 FOR CONT1:=1 TO NVARIA DO
49900     AUXILIARCONT1:=SIMBOLO.LETRACONT1;
50000 FOR CONT1:=1 TO (NVARIA-1) DO
50100 FOR CONT2:=(CONT1+1) TO NVARIA DO
50200     IF AUXILIARCONT2 < AUXILIARCONT1 THEN
50300         BEGIN
50400             AUX:=AUXILIARCONT1;
50500             AUXILIARCONT1:=AUXILIARCONT2;
50600             AUXILIARCONT2:=AUX;
50700         END;
50800     CONT1:=0;
50900 FOR K:=(NVARIA-1) DOWNT0 0 DO
51000     IF BINARYO[K]<>'X' THEN
51100         BEGIN
51200             SOLUCI.SOLTOTAL[CONT,CONT1]:=AUXILIAK+1;
51300             SOLUCI.SOLTOTAL[CONT,CONT1+1]:=BINARYO[K];
51400             CONT1:=CONT1+2;
51500             SOLUCI.NSOLTOTAL[CONT]:=CONT1;
51600         END;
51700     CONT:=CONT+1;
51800 END; (* SOLUMAO *)
51900
52000 PROCEDURE BURBUJA(VAR DIGITOS:VECTOR;NDIGITOS:INTEGER);
52100 VAR
52200     CONT1,AUX,CONT2:INTEGER;
52300 BEGIN
52400     FOR CONT2:=0 TO (NDIGITOS-1) DO
52500     FOR CONT1:=(CONT2+1) TO NDIGITOS DO
52600     IF DIGITOSE[CONT1] < DIGITOSE[CONT2] THEN
52700         BEGIN
52800             AUX:=DIGITOSE[CONT2];
52900             DIGITOSE[CONT2]:=DIGITOSE[CONT1];
53000             DIGITOSE[CONT1]:=AUX;
53100         END;
53200 END; (* DE BURBUJA *)
53300
53400 PROCEDURE CUDRESECUN;
53500 VAR
53600     CONTCUBRE,I,J,II,J1,K,CONT:INTEGER;
53700 BEGIN
53800     CONTCUBRE:=0;
53900 FOR I:=0 TO NDIGITOS DO
54000     IF SOLUCI.CUBIERTOENIMPLIC+1,I] = ' ' THEN
54100         CONTCUBRE:=CONTCUBRE+1;
54200     IF CONTCUBRE > 0 THEN BANDERA2:=TRUE ELSE BANDERA2:=FALSE;
54300     II:=1;
54400     NSECUN:=0;
54500     IF CONTCUBRE > 0 THEN
54600     BEGIN
54700         I:=0;
54800         FOR J:=0 TO NDIGITOS DO
54900             IF SOLUCI.CUBIERTOENIMPLIC+1,J] = ' ' THEN
55000                 BEGIN
55100                     DIGSECCJ:=DIGITOSE[J];
55200                     I:=I+1;
55300                 END;
55400             NDIGSEC:=I-1;
55500             FOR J:=1 TO NIMPLIC DO
55600

```

```

55503
55504
55600 IF SOLUCI.MARCCJJ <> '*' THEN
55700     SECUNDARIOS(J,II);
55800     NIMPSEC:=II-1;
55900     IF BANDTOTAL THEN
56000     ACONJUNTO(NDIGSEC,NIMPSEC,SOLUCI.NTABSEC,SOLUCI.NOCUBIERTO,NSECUN);
56100 END; (* IF *)
56200     CONT:=1;
56300     FOR I:=1 TO (NIMPLIC) DO
56400         IF SOLUCI.MARCCIJ = '*' THEN
56500             SOLUMAQ(CONT,SOLUCI.NTABIMPCIJ,I,SOLUCI.TABIMP);
56600     FOR I:=0 TO NSECUN DO
56700         BEGIN
56800             K:=0;
56900             FOR J:=1 TO CONT-1 DO
57000                 BEGIN
57100                     J1:=0;
57200                     WHILE J1<SOLUCI.NSOLTOTALECJJ DO
57300                         BEGIN
57400                             SOLUCIONESCI,KJ:=SOLUCI.SOLTOTALECJ,J1;
57500                             K:=K+1;
57600                             J1:=J1+1;
57700                         END;
57800                             SOLUCIONESCI,KJ:='+';
57900                             K:=K+1;
58000                         END;
58100                             IF K < 2 THEN
58200                                 NSOLUCIONESCIJ:=0
58300                             ELSE
58400                                 NSOLUCIONESCIJ:=K-1;
58500                             END;
58600 IF (CONTCUBRE > 0) AND BANDTOTAL THEN
58700 BEGIN
58800     FOR I:=0 TO NSECUN DO
58900     FOR J:=0 TO SOLUCI.NMATSOLCIJ-1 DO
59000     BEGIN
59100         SOLUMAQ(CONT,SOLUCI.NTABSECC(SOLUCI.MATSOLCI,JJ),SOLUCI.MATSOLCI,JJ,
59200             SOLUCI.IMPSEC);
59300     IF NSOLUCIONESCIJ > 0 THEN
59400     BEGIN
59500         SOLUCIONESCI,NSOLUCIONESCIJ:=+'';
59600         J1:=NSOLUCIONESCIJ+2;
59700     END
59800     ELSE
59900         J1:=0;
60000     K:=0;
60100     WHILE K<SOLUCI.NSOLTOTALECONT-1 DO
60200     BEGIN
60300         SOLUCIONESCI,J1:=SOLUCI.SOLTOTALECONT-1,K;
60400         K:=K+1;
60500         J1:=J1+1;
60600     END;
60700     NSOLUCIONESCIJ:=J1;
60800     END;
60900 END;
61000     TOTSOL:=CONT;
61100 END; (* CURRESECUN *)
61200
61300 PROCEDURE CUDREMIN(CUBIERTOS:HATCARAC;NIMPL,NDIGIS:INTEGER;DIGIS:VECTOR)
61400 ; BEGIN
61500     READLN;
61600     I:=0;
61700     BANDERA:=TRUE;
61701
61702

```

```

61703
61704
61800 WHILE BANDERA DO
61900 BEGIN
62000 IF I <= NDIGIS THEN
62100 READ(CUBALUMCIJ);
62200 WHILE (CUBALUMCIJ = ' ') AND (NOT EOLN) DO
62300 READ(CUBALUMCIJ);
62400 IF (NOT EOLN) OR (CUBALUMCIJ = '@') OR (CUBALUMCIJ = 'B') THEN
62500 BEGIN
62600 CASE CUBALUMCIJ OF
62700 '0' : CUBALUMCIJ:='0';
62800 'B','b': CUBALUMCIJ:=' ';
62900 ELSE: BEGIN
63000 WRITELN('CARACTER NO RECONOCIDO, REPITE LA LINEA',
63100 ' CORRECTAMENTE');
63200 READLN;
63300 I:=-1;
63400 END;
63500 END; (* CASE *)
63600 I:=I-1;
63700 END
63800 ELSE
63900 IF I <> (NDIGIS) THEN
64000 BEGIN
64100 WRITELN('NO CUBRISTE CORRECTAMENTE LOS MINTERMINOS; INTEN',
64200 ' TALO NUEVAMENTE');
64300 READLN;
64400 I:=0;
64500 BANDERA:=TRUE;
64600 END
64700 ELSE
64800 BEGIN
64900 BANDERA:=FALSE;
65000 FOR CONT:=0 TO NDIGIS DO
65100 FOR CONT:=0 TO NDIGIS DO
65200 IF CUBALUMCIJ <> CUBIERTOSINIMPL,CONTJ THEN
65300 BEGIN
65400 WRITE('REVISAR EL MINTERMINO ',DIGIS[CONTJ:2]);
65500 IF CUBIERTOSINIMPL,CONTJ = ' ' THEN
65600 WRITELN(' NO SE PUEDE CUBRIR');
65700 ELSE
65800 WRITELN(' SI SE CUBRE');
65900 BANDERA:=TRUE;
66000 END;
66100 IF BANDERA THEN
66200 BEGIN
66300 FOR I:=0 TO 30 DO
66400 CUBALUMCIJ:=' ';
66500 READLN;
66600 END;
66700 I:=0;
66800 END;
66900 END;
67000 END; (*CUBREMIN *)
67100
67200 PROCEDURE SOLUCION;
67300 VAR
67400 MODUL,AUX,MED,DIFERENCIA,J1,K2,L,T,I,J,K,I1 : INTEGER;
67500 CONTCUBRE,ESTADO,I3,I2,K1,CONT,CONT1,DIR,
67600 T1:INTEGER;
67700 BEGIN
67800 BANDERA:=TRUE;
67900 SOLUCI.NELEM(-1):=0;
67901
67902

```

```

67903
67904
68000 FOR I:=0 TO 70 DO
68100 BEGIN
68200   FOR J:=0 TO 50 DO
68300     BEGIN
68400       SOLUCI.ARREGLOC1,JJ:=0;
68500       SOLUCI.ARRMARCAC1,JJ:='0';
68600     END;
68700     SOLUCI.NELEM1J:=0;
68800     SOLUCI.NSOLTOTAL1J:=0;
68900   END;
69000   FOR I:=1 TO 30 DO
69100     BEGIN
69200       FOR J:=0 TO 50 DO
69300         BEGIN
69400           SOLUCI.NOCUBIERTO1,JJ:=' ';
69500           SOLUCI.CUBIERTO1,JJ:=' ';
69600         END;
69700         SOLUCI.MARCC1J:=' ';
69800       END;
69900       FOR I:=0 TO NDIGITOS DO
70000         CASE DIGITOS1J OF
70100           0: SOLUCI.NELEM0J:=1;
70200           1,2,4,8,16,32,64: BEGIN
70300             SOLUCI.ARREGLOC1,SOLUCI.NELEM1J:=DIGITOS1J;
70400             SOLUCI.NELEM1J:=SOLUCI.NELEM1J+1;
70500           END;
70600           3,5,6,9,10,12,17,18,20,24,33,34,36,40,48: BEGIN
70700             SOLUCI.ARREGLOC2,SOLUCI.NELEM2J:=DIGITOS1J;
70800             SOLUCI.NELEM2J:=SOLUCI.NELEM2J+1;
70900           END;
71000           7,11,13,14,19,21,22,25,26,28,35,37,38,41,42,44,49,50,52,56: BEGIN
71100             SOLUCI.ARREGLOC3,SOLUCI.NELEM3J:=DIGITOS1J;
71200             SOLUCI.NELEM3J:=SOLUCI.NELEM3J+1;
71300           END;
71400           15,23,27,29,30,39,43,45,46,51,53,54,57,58,60: BEGIN
71500             SOLUCI.ARREGLOC4,SOLUCI.NELEM4J:=DIGITOS1J;
71600             SOLUCI.NELEM4J:=SOLUCI.NELEM4J+1;
71700           END;
71800           31,47,55,59,61,62: BEGIN
71900             SOLUCI.ARREGLOC5,SOLUCI.NELEM5J:=DIGITOS1J;
72000             SOLUCI.NELEM5J:=SOLUCI.NELEM5J+1;
72100           END;
72200         63: BEGIN
72300           SOLUCI.ARREGLOC6,SOLUCI.NELEM6J:=DIGITOS1J;
72400           SOLUCI.NELEM6J:=SOLUCI.NELEM6J+1;
72500         END;
72600       END; (* CASE *)
72700     IF SOLUCI.NELEM0J=1 THEN
72800     IF SOLUCI.NELEM1J>0 THEN
72900     BEGIN
73000       SOLUCI.ARRMARCAC0,0J:='C';
73100       FOR I:=0 TO (SOLUCI.NELEM1J*2-1) DO
73200         BEGIN
73300           SOLUCI.ARREGLOC11,IJ:=0;
73400           I:=I+1;
73500           MED:=I DIV 2;
73600           CASE SOLUCI.ARREGLOC1,MEDJ OF
73700             1: CDIGITOSC11,1,I-1J:='1';
73800             2: CDIGITOSC11,1,I-1J:='2';
73900             4: CDIGITOSC11,1,I-1J:='4';
74000             8: CDIGITOSC11,1,I-1J:='8';
74100             ELSE: CDIGITOSC11,1,I-1J:='X';
74101
74102

```



```

80303
80304
80400 BANDERA:=TRUE;
80500 T1:=EXPONENTE(K DIV 10);
80600 FOR I:=0 TO (SOLUCI.NELEMCTJ-1) DO
80700 BEGIN
80800   FOR J:=0 TO (SOLUCI.NELEMCT+1J-1) DO
80900     BEGIN
81000       DIFERENCIA:=SOLUCI.ARREGLOCT+1,JJ-SOLUCI.ARREGLOCT,IJ;
81100       FOR J1:=1 TO K2 DO
81200         BEGIN
81300           J:=J+1;
81400           I:=I+1;
81500           AUX:=SOLUCI.ARREGLOCT+1,JJ-SOLUCI.ARREGLOCT,IJ;
81600           IF (AUX=DIFERENCIA) AND (AUX>0) AND (AUX IN UNO) THEN
81700             BANDERA:=BANDERA AND TRUE
81800           ELSE
81900             BANDERA:=FALSE;
82000         END;
82100         I:=I-K2;
82200         IF BANDERA THEN
82300           BEGIN
82400             J:=J-K2;
82500             FOR I1:=1 TO I+K2 DO
82600               SOLUCI.ARRMARCACT,I1J:='C';
82700             FOR I1:=J TO J+K2 DO
82800               SOLUCI.ARRMARCACT+1,I1J:='C';
82900             I1:=I;
83000             I2:=J;
83100             FOR I3:=K1 TO (K1+K2) DO
83200               BEGIN
83300                 PBURBCI3-K1J:=SOLUCI.ARREGLOCT,I1J;
83400                 PBURBCI3-K1+K2+1J:=SOLUCI.ARREGLOCT+1,I2J;
83500                 I1:=I1+1;
83600                 I2:=I2+1;
83700               END;
83800             BURBUJA(PBURB,T1-1);
83900             IF K1>0 THEN
84000               FOR I3:=0 TO ((K1 DIV T1) -1) DO
84100                 BEGIN
84200                   BANDERA:=TRUE;
84300                   CONT:=0;
84400                   FOR I1:=T1*I3 TO T1*(I3+1)-1 DO
84500                     BEGIN
84600                       IF PBURBCCONTJ = SOLUCI.ARREGLOCK,I1J THEN
84700                         BANDERA:=BANDERA AND TRUE
84800                       ELSE
84900                         BANDERA:=FALSE;
85000                       CONT:=CONT+1;
85100                     END;
85200                   IF BANDERA THEN
85300                     I3:=K1;
85400                 END
85500               ELSE
85600                 BANDERA:=FALSE;
85700               IF NOT BANDERA THEN
85800                 BEGIN
85900                   CONT:=K1;
86000                   FOR I3:=0 TO (T DIV 10) DO
86100                     BEGIN
86200                       CASE I3 OF
86300                         0: I1:=1;
86400                         1: I1:=2;
86500                         2: I1:=4;

```



```

86503
86504
86600          3: I1:=8;
86700          4: I1:=16;
86800          END; (* CASE *)
86900          DIFERENCIA:=PBURBCI1J - PBURBCOJ;
87000          CASE DIFERENCIA OF
87100            1: CDIGITOSEK,1,CONTJ:='1';
87200            2: CDIGITOSEK,1,CONTJ:='2';
87300            4: CDIGITOSEK,1,CONTJ:='4';
87400            8: CDIGITOSEK,1,CONTJ:='8';
87500            ELSE: CDIGITOSEK,1,CONTJ:='X';
87600          END; (* CASE *)
87700          CONT:=CONT+1;
87800          END;
87900          FOR I3:=K1 TO (K1+T1-1) DO
88000            SOLUCI.ARREGLOCK,I3J:=PBURBCI3-K1J;
88100            SOLUCI.NELEMCKJ:=SOLUCI.NELEMCKJ+T1;
88200            K1:=SOLUCI.NELEMCKJ;
88300          END;
88400          J:=J+K2;
88500          END
88600          ELSE
88700          FOR I1:=I TO I+K2 DO
88800            IF SOLUCI.ARRMARCACT,I1J <> 'C' THEN
88900              SOLUCI.ARRMARCACT,I1J:='*';
89000            BANDERA:=TRUE;
89100          END;
89200          I:=I+K2;
89300          END;
89400          T:=T+1;
89500          K:=K+1;
89600          END
89700          ELSE
89800          BEGIN
89900            FOR I:=0 TO SOLUCI.NELEMCTJ-1 DO
90000              IF SOLUCI.ARRMARCACT,IJ <> 'C' THEN
90100                SOLUCI.ARRMARCACT,IJ:='*';
90200              T:=T+2; K:=K+2;
90300            END;
90400            K2:=T1-1;
90500            T:=((T+10) DIV 10)*10+1;
90600            K:=((K+10) DIV 10)*10+1;
90700            WHILE (SOLUCI.NELEMCTJ=0) AND (T<70) DO
90800              BEGIN
90900                T:=T+1;
91000                K:=K+1;
91100              END;
91200            END;
91300            BURBUJA(DIGITOS,NDIGITOS);
91400            NIMPLIC:=1;
91500            T:=70;
91600            WHILE (SOLUCI.NELEMCTJ = 0) AND (T>0) DO
91700              T:=T-1;
91800            MAX:=T DIV 10;
91900            WHILE (T >= 0) AND (SOLUCI.NELEMCTJ <> 0) DO
92000              BEGIN
92100                MODUL:=(T DIV 10)* 10;
92200                WHILE (T > MODUL) OR (T = 0) DO
92300                  BEGIN
92400                    J:=0;
92500                    T1:=EXPONENTE(T DIV 10);
92600                    WHILE J < SOLUCI.NELEMCTJ DO
92700                      BEGIN
92701
92702

```

```

92703
92704
92800 WHILE (SOLUCI.ARRMARCACT,JJ <> '*') AND (J<SOLUCI.NELEMCTJ) DO
92900   J:=J+1;
93000 IF SOLUCI.ARRMARCACT,JJ = '*' THEN
93100   BEGIN
93200     FOR I:=0 TO (T1-1) DO
93300       BEGIN
93400         SOLUCI.TABIMPCNIMPLIC,IJ:=SOLUCI.ARREGLOCT,JJ;
93500         J:=J+1;
93600       END;
93700       SOLUCI.NTABIMPCNIMPLICJ:=T1;
93800       NIMPLIC:=NIMPLIC+1;
93900     END;
94000   END;
94100   IF T>=0 THEN
94200     T:=T-1;
94300   WHILE (SOLUCI.NELEMCTJ=0) AND (T>MODUL) DO
94400     T:=T-1;
94500   END;
94600   IF T >= 0 THEN
94700     T:=T-1;
94800   WHILE (SOLUCI.NELEMCTJ=0) AND (T>0) DO
94900     T:=T-1;
95000   END;
95100   NIMPLIC:=NIMPLIC-1;
95200   FOR I:=1 TO NIMPLIC DO
95300     FOR J:=0 TO NDIGITOS DO
95400       FOR K:=0 TO (SOLUCI.NTABIMPCJ-1) DO
95500         IF SOLUCI.TABIMPCI,KJ = DIGITOSCJ THEN
95600           SOLUCI.CUBIERTOCTI,JJ:='0';
95700
95800   FOR I:=0 TO NDIGITOS DO
95900     BEGIN
96000       CONTCUBRE:=0;
96100       FOR J:=1 TO NIMPLIC DO
96200         IF SOLUCI.CUBIERTOCTJ,IJ = '0' THEN
96300           BEGIN
96400             CONTCUBRE:=CONTCUBRE+1;
96500             ESTADO:=J;
96600           END;
96700           IF CONTCUBRE = 1 THEN
96800             BEGIN
96900               SOLUCI.MARCEESTADOJ:='*';
97000               FOR K:=0 TO NDIGITOS DO
97100                 IF SOLUCI.CUBIERTOCESTADO,KJ = '0' THEN
97200                   SOLUCI.CUBIERTOENIMPLIC+1,KJ:='0';
97300             END; (* IF *)
97400           END; (* FOR *)
97500           CUBRESECUN;
97600   END; (* DE SOLUCION *)
97700
97800 PROCEDURE DECIMALES(VAR DIGS:VECTOR;VAR NDIGS:INTEGER);
97900 LABEL 60;
98000 VAR
98100   AUX1,CNUMERO,CCERO,I,I1,T1,J,I2:INTEGER;
98200   MINTER:CHAR;
98300 BEGIN
98400   60:
98500   READLN;
98600   READ(MINTER);
98700   J:=0;
98800   CCERO:=ORD('0');
98900   WHILE MINTER <> ';' DO
99001
89907.

```

```

98903
98904
99000 BEGIN
99100 WHILE MINTER = ' ' DO
99200 READ (MINTER);
99300 I:=1;
99400 AUX1:=0;
99500 IF((MINTER <='9') AND (MINTER >='0')) THEN
99600 BEGIN
99700 WHILE ((MINTER<='9') AND (MINTER >='0')) DO
99800 BEGIN
99900 CNUMERO:=ORD(MINTER);
100000 AUX1:=AUX1*I+(CNUMERO - CCERO);
100100 I:=10*I;
100200 READ(MINTER);
100300 END;
100400 IF (AUX1>=POTENCIA) OR (AUX1<0) THEN
100500 BEGIN
100600 WRITE('UN NUMERO DE LOS PROPORCIONADOS, NO ESTA EN UN');
100700 WRITELN(' RANGO PERMISIBLE, VERIFICA');
100800 GOTO 60;
100900 END
101000 ELSE
101100 IF J=0 THEN
101200 FOR I2:=J-1 DOWNTD 0 DO
101300 IF DIGSEI2J = AUX1 THEN
101400 BEGIN
101500 WRITELN('HAY UN NINTERMINO REPETIDO,VERIFICA ');
101600 GOTO 60;
101700 END;
101800 DIGSEI2J:=AUX1;
101900 J:=J+1;
102000 WHILE MINTER = ' ' DO
102100 READ(MINTER);
102200 IF MINTER = ' ' THEN
102300 READ (MINTER);
102400 END
102500 ELSE
102600 BEGIN
102700 WRITELN('HAY UN CARACTER NO VALIDO, VERIFICA');
102800 GOTO 60;
102900 END;
103000 END;
103100 NDIGS:=J-1;
103200 END; (* DECIMALES *)
103300
103400 PROCEDURE IMPRTABLAS(NIMPRENG,DIGS: VECTOR; CUBRE:MATCARAC; NDIGS,
103500 NIMPS:INTEGER; MARCAS,ULCUBRE,MINT:CARACTER);
103600 VAR
103700 CONT,CONT1: INTEGER;
103800 BEGIN
103900 WRITELN;
104000 WRITE(' ');
104100 FOR CONT:=0 TO NDIGS DO
104200 WRITE(' ',DIGSECONTJ:2);
104300 WRITELN(' ');
104400 FOR CONT:=1 TO NIMPS DO
104500 BEGIN
104600 IF CONT > 1 THEN
104700 IF NIMPRENG[CONT] <> NIMPRENG[CONT-1] THEN
104800 BEGIN
104900 WRITE('-----');
105000 FOR CONT1:=0 TO NDIGS DO
105100 WRITE('-----');
105101
105102

```

```

105103
105104
105200      WRITELN;
105300      END;
105400      WRITE(MARCAS[CONT1:1,' ',MINT[CONT1:2]);
105500      FOR CONT1:=0 TO NDIGS DO
105600          WRITE(' ',CUBREC[CONT1:1]);
105700          WRITELN(' ');
105800      END;
105900      WRITE(' ');
106000      FOR CONT1:=0 TO NDIGS DO
106100          WRITE(' ',ULCUBREC[CONT1:1]);
106200      WRITELN(' ');
106300      END; (* IMPRTABLAS *)
106400
106500      PROCEDURE GRAFIC(CONT,HGRUP,CONT6:INTEGER);
106600      VAR
106700          CONT1,CONT3,CONT4,CONT6,CONT7 :INTEGER;
106800      BEGIN
106900          CONT7:=CONT6;
107000          IF CONT<10 THEN
107100              T1:=1
107200          ELSE
107300              T1:=EXPONENTE(CONT DIV 10);
107400          CASE CONT OF
107500              0,1,2,3,4: BEGIN
107600                  FOR CONT1:=1 TO NGRUP DO
107700                      BEGIN
107800                          WRITE(' ':4,'GRUPO ',CONT7:2,' ');
107900                          CONT7:=CONT7+1;
108000                      END;
108100                      CONT6:=-1;
108200                  END;
108300              11,12,13,14: BEGIN
108400                  FOR CONT1:=1 TO NGRUP DO
108500                      BEGIN
108600                          WRITE(' ':7,'GRUPO ',CONT7:2,' ':5);
108700                          CONT7:=CONT7+1;
108800                      END;
108900                      CONT6:=0;
109000                  END;
109100              21,22,23,24: BEGIN
109200                  FOR CONT1:=1 TO NGRUP DO
109300                      BEGIN
109400                          WRITE(' ':12,'GRUPO ',CONT7:2,' ':5);
109500                          CONT7:=CONT7+1;
109600                      END;
109700                      CONT6:=1;
109800                  END;
109900              31,32,33,34: BEGIN
110000                  FOR CONT1:=1 TO NGRUP DO
110100                      BEGIN
110200                          WRITE(' ':18,'GRUPO ',CONT7:2,' ':5);
110300                          CONT7:=CONT7+1;
110400                      END;
110500                      CONT6:=2;
110600                  END;
110700              41,42,43,44: BEGIN
110800                  FOR CONT1:=1 TO NGRUP DO
110900                      BEGIN
111000                          WRITE(' ':24,'GRUPO ',CONT7:2);
111100                          CONT7:=CONT7+1;
111200                      END;
111300                      CONT6:=3;
111301
111302

```

```

111303
111304
111400
111500
111600
111700
111800
111900
112000
112100
112200
112300
112400
112500
112600
112700
112800
112900
113000
113100
113200
113300
113400
113500
113600
113700
113800
113900
114000
114100
114200
114300
114400
114500
114600
114700
114800
114900
115000
115100
115200
115300
115400
115500
115600
115700
115800
115900
116000
116100
116200
116300
116400
116500
116600
116700
116800
116900
117000
117100
117200
117300
117400
117500
117501
117502

```

```

END;
51,52,53,54; BEGIN
FOR CONT1:=1 TO NGRUP DO
BEGIN
WRITE(' :24,'GRUPO ',CONT7:2);
CONT7:=CONT7+1;
END;
END;
CONT6:=4;
END;
END; (*CASE *)
WRITELN;
IF SOLUCI.NELEMCCONTJ > SOLUCI.NELEMCCONTJ+1 THEN
BEGIN
FOR CONT4:=0 TO SOLUCI.NELEMCCONTJ-1 DO
BEGIN
WRITE(' :6);
FOR CONT1:=CONT4 TO (CONT4+T1-2) DO
WRITE(SOLUCI.ARREGLOCCONT,CONT1:2,',');
WRITE(SOLUCI.ARREGLOCCONT,CONT4+T1-1:2);
IF CONT > 10 THEN
BEGIN
WRITE(' (');
FOR CONT1:=0 TO (CONT6-1) DO
WRITE(CDIGITOSECONT,1,CONT4+CONT1:1,',');
WRITE(CDIGITOSECONT,1,CONT6+CONT4:1,')');
END;
WRITE(CDIGITOSECONT,0,CONT4:1, ' :10);
FOR CONT1:=CONT4 TO (CONT4+T1-2) DO
WRITE(SOLUCI.ARREGLOCCONT+1,CONT1:2,',');
WRITE(SOLUCI.ARREGLOCCONT+1,CONT4+T1-1:2);
IF CONT > 10 THEN
BEGIN
WRITE(' (');
FOR CONT1:=0 TO (CONT6-1) DO
WRITE(CDIGITOSECONT+1,1,CONT4+CONT1:1,',');
WRITE(CDIGITOSECONT+1,1,CONT6+CONT4:1,')');
END;
WRITELN(CDIGITOSECONT+1,0,CONT4:1);
CONT4:=CONT4+T1-1;
END;
IF SOLUCI.NELEMCCONTJ < SOLUCI.NELEMCCONTJ+1 THEN
FOR CONT3:=CONT4 TO SOLUCI.NELEMCCONTJ-1 DO
BEGIN
WRITE(' :6);
FOR CONT1:=CONT3 TO (CONT3+T1-2) DO
WRITE(SOLUCI.ARREGLOCCONT,CONT1:2,',');
WRITE(SOLUCI.ARREGLOCCONT,CONT3+T1-1:2);
IF CONT > 10 THEN
BEGIN
WRITE(' (');
FOR CONT1:=0 TO (CONT6-1) DO
WRITE(CDIGITOSECONT,1,CONT3+CONT1:1,',');
WRITE(CDIGITOSECONT,1,CONT6+CONT3:1,')');
END;
WRITELN(CDIGITOSECONT,0,CONT3:1);
CONT3:=CONT3+T1-1;
END;
END;
ELSE
BEGIN
FOR CONT4:=0 TO SOLUCI.NELEMCCONTJ-1 DO
BEGIN

```

```

117503
117504
117600 WRITE(' :6);
117700 FOR CONT1:=CONT4 TO (CONT4+T1-2) DO
117800 WRITE(SOLUCI.ARREGLOCCONT,CONT1:2,',');
117900 WRITE(SOLUCI.ARREGLOCCONT,CONT4+T1-1:2);
118000 IF CONT > 10 THEN
118100 BEGIN
118200 WRITE(' ');
118300 FOR CONT1:=0 TO (CONT6-1) DO
118400 WRITE(CDIGITOSECONT,1,CONT4+CONT1:1,',');
118500 WRITE(CDIGITOSECONT,1,CONT6+CONT4:1,',');
118600 END;
118700 WRITE(CDIGITOSECONT,0,CONT4:1,' :10);
118800 FOR CONT1:=CONT4 TO (CONT4+T1-2) DO
118900 WRITE(SOLUCI.ARREGLOCCONT+1,CONT1:2,',');
119000 WRITE(SOLUCI.ARREGLOCCONT+1,CONT4+T1-1:2);
119100 IF CONT > 10 THEN
119200 BEGIN
119300 WRITE(' ');
119400 FOR CONT1:=0 TO (CONT6-1) DO
119500 WRITE(CDIGITOSECONT+1,1,CONT4+CONT1:1,',');
119600 WRITE(CDIGITOSECONT+1,1,CONT6+CONT4:1,',');
119700 END;
119800 WRITELN(CDIGITOSECONT+1,0,CONT4:1);
119900 CONT4:=CONT4+T1-1;
120000 END;
120100 IF SOLUCI.NELEMCONT3 <> SOLUCI.NELEMCONT+1 THEN
120200 FOR CONT3:=CONT4 TO SOLUCI.NELEMCONT+1:1 DO
120300 BEGIN
120400 WRITE(' :19);
120500 FOR CONT1:=0 TO CONT6 DO
120600 WRITE(' :7);
120700 FOR CONT1:=CONT3 TO (CONT3+T1-2) DO
120800 WRITE(SOLUCI.ARREGLOCCONT+1,CONT1:2,',');
120900 WRITE(SOLUCI.ARREGLOCCONT+1,CONT3+T1-1:2);
121000 IF CONT > 10 THEN
121100 BEGIN
121200 WRITE(' ');
121300 FOR CONT1:=0 TO (CONT6-1) DO
121400 WRITE(CDIGITOSECONT+1,1,CONT3+CONT1:1,',');
121500 WRITE(CDIGITOSECONT+1,1,CONT6+CONT3:1,',');
121600 END;
121700 WRITELN(CDIGITOSECONT+1,0,CONT3:1);
121800 CONT3:=CONT3+T1-1;
121900 END;
122000 END;
122100 END; (* GRAFIC *)
122200
122300 PROCEDURE BUSCA(VAR T,UBICA:INTEGER; INICIO,FIN:INTEGER);
122400 VAR
122500 CONT,CONT1; INTEGER;
122600 BEGIN
122700 WHILE (SOLUCI.NELEMCTJ = 0) AND (T > (T1*10)) DO
122800 T:=T-1;
122900 BANDERA:=FALSE;
123000 IF (T > T1*10) OR (T=0) THEN
123100 WHILE (NOT BANDERA) AND ((T>T1*10) OR (T=0)) DO
123200 BEGIN
123300 FOR CONT:=0 TO SOLUCI.NELEMCTJ-1 DO
123400 BEGIN
123500 IF SOLUCI.ARREGLOCT,CONTJ = CASILLASCINICIOJ THEN
123600 BEGIN
123700 BANDERA:=TRUE;
123701
123702

```

```

123703
123704
123800
123900
124000
124100
124200
124300
124400
124500
124600
124700
124800
124900
125000
125100
125200
125300
125400
125500
125600
125700
125800
125900
126000
126100
126200
126300
126400
126500
126600
126700
126800
126900
127000
127100
127200
127300
127400
127500
127600
127700
127800
127900
128000
128100
128200
128300
128400
128500
128600
128700
128800
128900
129000
129100
129200
129300
129400
129500
129600
129700
129800
129900
129901
129902

```

```

FOR CONT1:=INICIO TO FIN DO
BEGIN
  IF SOLUCI.ARREGLOCT,CONTJ=CASILLAS[CONT1] THEN
    BANDERA:=BANDERA AND TRUE
  ELSE
    BANDERA:=FALSE;
    CONT:=CONT+1;
  END;
  CONT:=CONT-1;
  END
ELSE
  BEGIN
    BANDERA:=FALSE;
    CONT:=CONT+(FIN-INICIO);
  END;
  IF BANDERA THEN
    BEGIN
      UBICA:=CONT-(FIN-INICIO);
      CONT:=SOLUCI.NELEMCTJ;
    END;
  END;
  T:=T-1;
END;
END; (* BUSCA *)

PROCEDURE IMPLICESEN(CUBO,MARCS:CHARACTER; NIMP:INTEGER);
BEGIN
  BANDERA:=TRUE;
  CONT:=1;
  READLN;
  WHILE BANDERA DO
    BEGIN
      FOR CONT1:=1 TO NIMP DO
        BEGIN
          MARCS[CONT1]:=' ';
          IMPSCONT1:=' ';
        END;
        WHILE NOT EOLN DO
          BEGIN
            READ(IMPSCONTJ);
            WHILE (IMPSCONTJ = ' ') DO
              READ(IMPSCONTJ);
            IF NOT EOLN THEN
              BEGIN
                IF (IMPSCONTJ > CUBO[NIMP]) OR (IMPSCONTJ <
                  CUBO[1]) THEN
                  BEGIN
                    WRITELN(IMPSCONTJ:1,' NO ES UN IMPLICANTE, VERIFICA');
                    READLN;
                    CONT:=0;
                  END
                ELSE
                  FOR CONT1:=1 TO NIMP DO
                    IF CUBO[CONT1] = IMPSCONTJ THEN
                      IF MARCS[CONT1] <> '*' THEN
                        BEGIN
                          WRITELN(IMPSCONTJ:1,' NO ES IMPLICANTE PRIMO ',
                            'ESSENCIAL, VERIFICA');
                          CONT:=0;
                          READLN;
                        END;
                      CONT:=CONT+1;
                    END;
                  END;
                END;
              END;
            END;
          END;
        END;
      END;
    END;
  END;

```

```

129903
129904
130000      END;
130100      END;
130200      IF EOLN THEN
130300      BEGIN
130400          FOR CONT3:=1 TO CONT DO
130500          FOR CONT4:=1 TO NIMP DO
130600              IF IMPSCCONT3J = CUBOCCONT4J THEN
130700                  MARCASCCONT4J:='*';
130800          BANDERA1:=TRUE;
130900          FOR CONT3:=1 TO NIMP DO
131000              IF MARCASCCONT3J <> MARCS[CONT3] THEN
131100                  BANDERA1:=FALSE;
131200          IF BANDERA1 THEN
131300              BANDERA1:=FALSE
131400          ELSE
131500              BEGIN
131600                  WRITELN('NO MARCASTE ALGUN IMPLICANTE ESCENCIAL');
131700                  CONT:=1;
131800                  READLN;
131900              END;
132000          END;
132100      END;
132200      END; (* IMPLICESEN *)
132300
132400      PROCEDURE VARELIN(CUBO:CHAR; VAR CONT1:INTEGER;RENGLON:INTEGER);
132500      VAR CONT2,CONT4: INTEGER;
132600      BEGIN
132700          SOLUCIONES[CONT10,0J]:=CUBO; SOLUCIONES[CONT10,1J]:=' ';
132800          SOLUCIONES[CONT10,2J]:=' '; SOLUCIONES[CONT10,3J]:=' ';
132900          CONT4:=4;
133000          FOR CONT2:=NVARIA DOWNT0 1 DO
133100              BEGIN
133200                  IF SOLUCI.SOLTOTAL[RENGLON,CONT1J] = SIMBOLO.LETRA[CONT2] THEN
133300                      BEGIN
133400                          SOLUCIONES[CONT10,CONT4J]:=SOLUCI.SOLTOTAL[RENGLON,CONT1+1J];
133500                          IF SOLUCI.SOLTOTAL[RENGLON,CONT1+2J] = '+' THEN
133600                              CONT1:=CONT1+3
133700                          ELSE
133800                              CONT1:=CONT1+2;
133900                      END
134000                      ELSE
134100                          SOLUCIONES[CONT10,CONT4J]:='X';
134200                          CONT4:=CONT4+1;
134300                      END;
134400                      CONT10:=CONT10+1;
134500                  END; (* VARELIN *)
134600
134700      (*      PROGRAMA      PRINCIPAL      *)
134800      BEGIN
134900          DIGITOS[0J]:=0;
135000          NDIGITOS:=0;
135100          UND:=[1,2,4,8,16,32,64];
135200          WRITELN('ESTE PROGRAMA TE GUIARA EN LA MINIMIZACION DE FUNCIO',
135300              'NES POR MEDIO DEL');
135400          WRITELN('METODO DE QUINE-McCLUSKEY');
135500          WRITELN('DESEAS CONOCER UNA PEQUERA INTRODUCCION A ESTE METODO?');
135600          READ(CAMBIOC1J,CAMBIOC2J);
135700          WHILE NOT (((CAMBIOC1J='S') AND (CAMBIOC2J='I')) OR
135800              ((CAMBIOC1J='N') AND (CAMBIOC2J='O')))) DO
135900              BEGIN
136000                  WRITELN('REPITE SI DESEAS O NO CONOCERLA');
136100                  READLN; READ(CAMBIOC1J,CAMBIOC2J);
136101
136102

```



```

136103
136104
136200 END;
136300 IF CAMBIO[1] = 'S' THEN
136400     TEXTO;
136500     CUBO[1]:='a';
136600     FOR CONT:=2 TO 25 DO
136700         CUBO[CONT]:=SUCC(CUBO[CONT-1]);
136800     WRITELN('CUANTAS VARIABLES TIENE TU FUNCION?');
136900     READLN; READ(NVARIA);
137000     WHILE ((NVARIA < 1) OR (NVARIA > 4)) DO
137100         BEGIN
137200             IF NVARIA < 1 THEN
137300                 WRITELN('CON ESTE PROGRAMA PUEDES RESOLVER FUNCIONES ',
137400                     '"REALES", REPITE POR FAVOR')
137500             ELSE
137600                 BEGIN
137700                     WRITELN('ESTE PROGRAMA TIENE FINES DIDACTICOS Y ESTA ',
137800                         'SUJETO AL ESPACIO DE LA ');
137900                     WRITELN('PANTALLA, POR LO CUAL RESUELVE FUNCIONES CON EN MAXI',
138000                         'MO DE 4 VARIABLES');
138100                     WRITELN('DESEAS PROPONER ALGUNA FUNCION QUE PUEDES ',
138200                         'RESOLVER A TRAVES DE ESTE ');
138300                     WRITELN('PROGRAMA?');
138400                     READLN; READ(CAMBIO[1],CAMBIO[2]);
138500                     WHILE NOT (((CAMBIO[1]='S')AND(CAMBIO[2]='I')) OR ((CAMBIO[1]
138600                         ='N') AND (CAMBIO[2]='O')))) DO
138700                         BEGIN
138800                             WRITELN('REPITE SI DESEAS PROPONER OTRA FUNCION');
138900                             READLN; READ(CAMBIO[1],CAMBIO[2]);
139000                         END;
139100                         IF ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))THEN
139200                             ABORT
139300                         END;
139400                     WRITELN('CUANTAS VARIABLES ENTONCES DESEAS USAR?');
139500                     READLN; READ(NVARIA);
139600                     WRITELN;
139700                 END;
139800             SIMBOLO.LETRAC[1]:='A';
139900             FOR I:=2 TO NVARIA DO
140000                 SIMBOLO.LETRAC[I]:=SUCC(SIMBOLO.LETRAC[I-1]);
140100             POTENCIA:=EXPONENTE(NVARIA);
140200             WRITELN('LAS VARIABLES DEFINIDAS EN PESO DE MAYOR A MENOR SON:');
140300             FOR I:=NVARIA DOWNTO 1 DO
140400                 WRITE(SIMBOLO.LETRAC[I];1);
140500             WRITELN;
140600             WRITELN;
140700             WRITELN('PROPORCIONA LOS MINTERMINOS EN FORMA DECIMAL ',
140800                 'SEPARADOS POR COMAS Y');
140900             WRITELN('TERMINA CON EL CARACTER (,);');
141000             DECIMALES(DIGITOS,NDIGITOS);
141100             BURBUJA(DIGITOS,NDIGITOS);
141200             BANDTOTAL:=FALSE;
141300             SOLUCION;
141400             CONTAB:=0;
141500             WRITELN;
141600             WRITELN('TUS MINTERMINOS (CUBOS=0) ORDENADOS EN FUNCION DEL ',
141700                 'NUMERO DE UNOS CON SU');
141800             WRITELN('REPRESENTACION BINARIA Y DECIMAL SON:');
141900             CONT6:=NVARIA;
142000             WHILE SOLUCI.NELEM[CONT6] = 0 DO
142100                 CONT6:=CONT6-1;
142200             CONT1:=0;
142300             WHILE SOLUCI.NELEM[CONT1] = 0 DO
142301
142302

```

```

142303
142304
142400     CONT1:=CONT1+1;
142500     CONT2:=0;
142600     FOR CONT1:=CONT1 TO CONT6 DO
142700     IF SOLUCI.NELEMCONT1 > CONT2 THEN
142800         CONT2:=SOLUCI.NELEMCONT1;
142900     CONT4:=1;
143000     FOR CONT1:=CONT1 TO CONT6 DO
143100     BEGIN
143200         WRITE(' GRUPO ',CONT:1);
143300         CONT4:=CONT4+1;
143400     END;
143500     WRITELN;
143600     FOR CONT:=0 TO CONT2 DO
143700     BEGIN
143800         WRITE(' ');
143900         FOR CONT4:=CONT1 TO CONT6 DO
144000         IF CONT < SOLUCI.NELEMCONT4 THEN
144100         BEGIN
144200             DECIMAL:=SOLUCI.ARREGLOCONT4,CONT;
144300             FOR CONT3:=1 TO NVARIA DO
144400             BEGIN
144500                 IF (DECIMAL MOD 2) = 1 THEN
144600                     BINARYOCONT3:='1'
144700                 ELSE
144800                     BINARYOCONT3:='0';
144900                 DECIMAL:=DECIMAL DIV 2;
145000             END;
145100             FOR CONT3:=NVARIA DOWNTO 1 DO
145200                 WRITE(BINARYOCONT3);
145300             CONT3:=NVARIA;
145400             WHILE CONT3 < 4 DO
145500             BEGIN
145600                 WRITE(' ');
145700                 CONT3:=CONT3+1;
145800             END;
145900             WRITE(' ',SOLUCI.ARREGLOCONT4,CONT:2,' ':5);
146000         END
146100         ELSE
146200             WRITE(' ':12);
146300         WRITELN;
146400     END;
146500     WRITELN('DE AQUI EN ADELANTE, USAREMOS LOS MINTERMINOS EN',
146600             'SU REPRESENTACION');
146700     WRITELN('DECIMAL');
146800     FOR CONT1:=0 TO 20 DO
146900     FOR CONT3:=0 TO 72 DO
147000         CDIGITOSECONT1,0,CONT3:=' ';
147100     CONT6:=0;
147200     WHILE (CONT6 < (MAX*10 +10)) DO
147300     BEGIN WRITELN;
147400     WRITELN('ANALISIS DE LOS CUBOS ',CONT6 DIV 10:1,' PARA OBTENER ',
147500             'LOS CUBOS ',CONT6 DIV 10 +1:1);
147600     IF (CONT6 DIV 10) > 0 THEN
147700     BEGIN WRITELN;
147800         WRITELN('PARA QUE SE PUEBAN COMBINAR CUBOS DE LOS SI',
147900                 'QUIENTES GRUPOS, SE DEBE');
148000         WRITELN('CUMPLIR CON:');
148100         WRITELN('A) LA(S) DIFERENCIA(S) INDICADA(S) EN EL PARENTESIS',
148200                 'DEBE(N) SER LA(S)');
148300         WRITELN('MISMA(S), YA QUE CORRESPONDE A LA MISMA VARIABLE ',
148400                 'ELIMINADA');
148500         WRITELN('B) LA(S) DIFERENCIA(S) ENTRE LOS ELEMENTOS CO',
148501
148502

```

```

148503
148504
148600          'RRESPONDIENTES ENTRE LOS '));
148700 WRITELN('CUDOS DE DIFERENTES GRUPOS DEBE SER IGUAL Y POTENCIA ',
148800          'DE 2');
148900 END;
149000 FOR CONT:=CONT6 TO (CONT6+INVARIA-1) DO
149100 BEGIN
149200   IF (SOLUCI.NELEMCCONTJ > 0) AND (SOLUCI.NELEMCCONTJ+1J > 0) THEN
149300     BEGIN
149400       CONT7:=CONT MOD 10;
149500       GRAFIC(CONT,2,CONT7);
149600       IF CONT < 10 THEN
149700         BEGIN
149800           WRITELN('EXISTEN EN ESTOS GRUPOS UN CUDO DE CADA',
149900                   ' UNO, CUYA DIFERENCIA SEA UNA');
150000           WRITELN('POTENCIA DE 2?');
150100           END
150200         ELSE
150300           WRITELN('EXISTEN EN ESTOS GRUPOS CUDOS QUE SE COMBINENT');
150400           READLN; READ(CAMBIOC1J,CAMBIOC2J);
150500           WHILE NOT ((CAMBIOC1J = 'S') AND (CAMBIOC2J = 'I')) OR
150600             ((CAMBIOC1J = 'N') AND (CAMBIOC2J = 'O')) DO
150700             BEGIN
150800               WRITELN('REPITE SI HAY CUDOS POR COMBINAR O NO');
150900               READLN; READ(CAMBIOC1J,CAMBIOC2J);
151000             END;
151100           REPEAT
151200             IF CAMBIOC1J = 'S' THEN
151300               BEGIN
151400                 BANDERA1:=TRUE;
151500                 FOR CONT1:=0 TO SOLUCI.NELEMCCONTJ-1 DO
151600                   IF SOLUCI.ARRMARCAECONT,CONT1J <> '*' THEN
151700                     BANDERA1:=FALSE;
151800                   IF NOT BANDERA1 THEN
151900                     BEGIN
152000                       BANDERA1:=TRUE;
152100                       FOR CONT1:=0 TO SOLUCI.NELEMCCONTJ+1J-1 DO
152200                         IF SOLUCI.ARRMARCAECONT+1,CONT1J <> '*' THEN
152300                           BANDERA1:=FALSE;
152400                       END;
152500                       IF BANDERA1 THEN
152600                         BEGIN
152700                           WRITELN('OBSERVA BIEN LOS GRUPOS! NO SE PUEDEN COMBINAR');
152800                           CAMBIOC1J:='N';
152900                           END;
153000                         END
153100                       ELSE
153200                         BANDERA1:=FALSE;
153300                       IF CAMBIOC1J = 'S' THEN
153400                         BEGIN
153500                           WRITELN;
153600                           WRITELN('DANE UNA PAREJA DE ELLOS');
153700                           DECIMALES(CASILLAS,NCASILLAS);
153800                           FOR CONT2:=0 TO NCASILLAS DO
153900                             CASIECONT2J:=CASILLASECONT2J;
154000                             BURBUJA(CASILLAS,NCASILLAS);
154100                             WHILE NOT ((NCASILLAS+1) IN UNO) DO
154200                               BEGIN
154300                                 IF NCASILLAS = 0 THEN
154400                                   WRITELN('TE FEDI UNA PAREJA, VERIFICA');
154500                                 ELSE
154600                                   WRITELN('EL NUMERO DE ELEMENTOS QUE PROPONES NO ES',
154700                                           ' POTENCIA DE DOS, VERIFICA');
154701
154702

```

```

154703
154704
154800          DECIMALES(CASILLAS,NCASILLAS);
154900          END;
155000          T1:=RAIZ(NCASILLAS+1);
155100          T:=T1*10 + 10;
155200          BUSCA(T,CONT2,0,NCASILLAS);
155300          T:=T+1;
155400          IF BANDERA THEN
155500          BEGIN
155600              BANDERA1:=TRUE;
155700              CONTAB:=T;
155800              CONT5:=CONT2;
155900              FOR CONT2:=0 TO NCASILLAS DO
156000                  CASILLAS[CONT2]:=CASI[CONT2];
156100                  T:=CONT+1;
156200                  T1:=T1 DIV 2;
156300                  BUSCA(T,CONT2,(NCASILLAS DIV 2)+1,NCASILLAS);
156400                  WRITELN;
156500                  IF BANDERA AND ((T+1) = (CONT+1)) THEN
156600                  BEGIN
156700                      FOR CONT1:=CONT2 TO (CONT2+NCASILLAS DIV 2) DO
156800                          IF CDIGITOS[CONT+1,0,CONT1] <> 'C' THEN
156900                              BANDERA1:=FALSE;
157000                      CONT4:=CONT2;
157100                      BUSCA(T,CONT2,0,NCASILLAS DIV 2);
157200                      IF BANDERA AND ((T+1) = CONT) THEN
157300                      BEGIN
157400                          FOR CONT1:=CONT2 TO (CONT2+NCASILLAS DIV 2) DO
157500                              IF CDIGITOS[CONT,0,CONT1] <> 'C' THEN
157600                                  BANDERA1:=FALSE;
157700                      END
157800                      ELSE
157900                      BEGIN
158000                          WRITELN('ALGUNO(S) DE LOS CUBOS, NO PERTENECE(N) A LOS ',
158100                              'GRUPOS QUE TE PRESENTE,');
158200                          WRITELN('VERIFICA');
158300                          BANDERA:=FALSE;
158400                      END
158500                      END
158600                      ELSE
158700                      BEGIN
158800                          WRITELN('ALGUNO(S) DE LOS CUBOS, NO PERTENECE(N) A LOS ',
158900                              'GRUPOS QUE TE PRESENTE,');
159000                          WRITELN('VERIFICA');
159100                          BANDERA:=FALSE;
159200                      END;
159300                      IF BANDERA1 AND BANDERA THEN
159400                      BEGIN
159500                          WRITELN('TODOS LOS CUBOS PROPUESTOS YA ESTAN CUBIERTOS ',
159600                              'ES REDUNDANTE VOLVER A ');
159700                          WRITELN('TOMARLOS');
159800                      END
159900                      ELSE
160000                      IF BANDERA AND NOT BANDERA1 THEN
160100                      BEGIN
160200                          FOR CONT1:=CONT4 TO (NCASILLAS DIV 2 + CONT4) DO
160300                              CDIGITOS[CONT+1,0,CONT1]:='C';
160400                          FOR CONT1:=CONT2 TO (NCASILLAS DIV 2 + CONT2) DO
160500                              CDIGITOS[CONT,0,CONT1]:='C';
160600                          WRITELN('BIEN, ESTOS CUBOS SI SE COMBINAN');
160700                          GRAFIC(CONT,2,CONT7);
160800                      END;
160900                      BANDERA:=TRUE;
160901
160902

```

```

160702
160903
160904
161000 WRITELN('AUN HAY CUBOS POR COMBINAR?');
161100 READLN; READ(CAMBIOC1J,CAMBIOC2J);
161200 WHILE NOT (((CAMBIOC1J = 'S') AND (CAMBIOC2J = 'I')) OR
161300 ((CAMBIOC1J = 'N') AND (CAMBIOC2J = 'O')))) DO
161400 BEGIN
161500 WRITELN('REPITE SI HAY CUBOS POR COMBINAR O NO');
161600 READLN; READ(CAMBIOC1J,CAMBIOC2J);
161700 END;
161800 BANDERA1:=TRUE;
161900 FOR CONT1:=0 TO SOLUCI.NELEMCONTJ-1 DO
162000 IF NOT ((CDIGITOSECONT,0,CONT1J='C') OR (CDIGITOSECONT,0,CONT1J
162100 = '*')) THEN
162200 BANDERA1:=FALSE;
162300 FOR CONT1:=0 TO SOLUCI.NELEMCONTJ+1J-1 DO
162400 IF NOT ((CDIGITOSECONT+1,0,CONT1J='C') OR (CDIGITOSECONT+1,0,
162500 CONT1J = '*')) THEN
162600 BANDERA1:=FALSE;
162700 IF BANDERA1 THEN
162800 IF CAMBIOC1J = 'S' THEN
162900 BEGIN
163000 WRITELN('YA CUBRISTE TODOS LOS CUBOS DE ESTOS GRUPOS, SOLO',
163100 ' GENERARIAS COMBINACIONES');
163200 WRITELN('DE LOS QUE YA TIENES, ASI QUE SEGUIREMOS CON LOS',
163300 ' SIGUIENTES GRUPOS');
163400 CAMBIOC1J:='F';
163500 END
163600 ELSE
163700 BEGIN
163800 WRITELN('BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS',
163900 ' DE ESTOS GRUPOS');
164000 IF SOLUCI.NELEMCONT+2J <> 0 THEN
164100 WRITELN('TE PRESENTARE LOS SIGUIENTES');
164200 WRITELN;
164300 CAMBIOC1J:='F';
164400 END;
164500 END
164600 ELSE
164700 BEGIN
164800 BANDERA1:=TRUE;
164900 WRITELN('LOS CUBOS QUE PROPONES NO SE COMBINAN, VERIFICA');
165000 CAMBIOC1J:='S';
165100 END;
165200 END
165300 ELSE
165400 BEGIN
165500 IF NOT BANDERA1 THEN
165600 BEGIN
165700 WRITELN('HUBO CUBOS QUE NO SE PUDIERON COMBINAR?');
165800 READLN; READ(CAMBIOC1J,CAMBIOC2J);
165900 WHILE NOT (((CAMBIOC1J='S') AND (CAMBIOC2J='I')) OR ((CAMBIOC1J='N')
166000 AND (CAMBIOC2J='O')))) DO
166100 BEGIN
166200 WRITELN('REPITE SI HUBO CUBOS QUE NO SE PUDIERON COMBINAR?');
166300 READLN; READ(CAMBIOC1J,CAMBIOC2J);
166400 END;
166500 END
166600 ELSE
166700 CAMBIOC1J:='S';
166800 IF CAMBIOC1J = 'S' THEN
166900 BEGIN
167000 CAMBIOC1J:='N';
167100 WHILE CAMBIOC1J = 'N' DO
167101
167102

```

```

167103
167104
167200 BEGIN
167300 BANDERA1:=TRUE;
167400 FOR CONT1:=0 TO SOLUCI.NELEM[CONTJ]-1 DO
167500 IF NOT ((CDIGITOSC[CONT,0,CONT1J]='C') OR (CDIGITOSC[CONT,0,CONT1J]
167600 = '*')) THEN
167700 BANDERA1:=FALSE;
167800 FOR CONT1:=0 TO SOLUCI.NELEM[CONT+1J]-1 DO
167900 IF NOT ((CDIGITOSC[CONT+1,0,CONT1J]='C') OR (CDIGITOSC[CONT+1,0,
168000 CONT1J = '*')) THEN
168100 BANDERA1:=FALSE;
168200 IF BANDERA1 THEN
168300 BEGIN
168400 WRITELN('YA CUBRISTE TODOS LOS CUBOS DE ESTOS GRUPOS, SOLO',
168500 ' GENERARIAS COMBINACIONES');
168600 WRITELN('DE LOS QUE YA TIENES, ASI QUE SEGUIREMOS CON LOS',
168700 ' SIGUIENTES GRUPOS');
168800 CAMBIO[CONTJ]='F';
168900 END;
169000 IF CAMBIO[CONTJ] <> 'F' THEN
169100 BEGIN
169200 WRITELN;
169300 WRITELN('DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA');
169400 DECIMALES(CASILLAS,NCASILLAS);
169500 BURBUJA(CASILLAS,NCASILLAS);
169600 IF NCASILLAS = 0 THEN
169700 T1:=0
169800 ELSE
169900 IF NCASILLAS = 1 THEN
170000 T1:=1
170100 ELSE
170200 T1:=RAIZ(NCASILLAS+1);
170300 T:=T1*10+10;
170400 BUSCA(T,CONT2,0,NCASILLAS);
170500 T:=T+1;
170600 WRITELN;
170700 IF BANDERA THEN
170800 IF SOLUCI.ARRNARCACT[CONT2J] <> '*' THEN
170900 BEGIN
171000 IF T = CONT THEN
171100 BEGIN
171200 WRITELN('ESTE CUBO SI SE PUEDE COMBINAR, INTENTALO. ');
171300 BANDERA:=FALSE;
171400 END
171500 ELSE
171600 IF T <> (CONT + 1) THEN
171700 BEGIN
171800 WRITELN('ESTE CUBO NO PERTENECE A LOS GRUPOS PRE',
171900 'SENTES, VERIFICA');
172000 BANDERA:=FALSE;
172100 END;
172200 IF BANDERA THEN
172300 BEGIN
172400 IF CONT > 0 THEN
172500 FOR CONT1:=0 TO SOLUCI.NELEM[CONTJ] - 1 DO
172600 BEGIN
172700 BANDERA:=TRUE;
172800 DIFERENCIA1:=CASILLAS[COJ] - SOLUCI.ARRELOC[CONT,CONT1J];
172900 FOR CONT3:=0 TO NCASILLAS DO
173000 BEGIN
173100 DIFERENCIA:=CASILLAS[CONT3J]-SOLUCI.ARRELOC[CONT,CONT1J];
173200 IF NOT ((DIFERENCIA = DIFERENCIA1) AND (DIFERENCIA IN
173300 UNO) AND (DIFERENCIA > 0)) THEN
173301
173302

```

```

173303
173304
173400          BEGIN
173500              BANDERA:=FALSE;
173600              CONT1:=CONT1+1;
173700          END;
173800      END;
173900      IF BANDERA THEN
174000          CONT1:=SOLUCI.NELEMCCONTJ
174100      ELSE
174200          CONT1:=CONT1-1;
174300      END
174400      ELSE
174500          IF SOLUCI.NELEMCIJ = 0 THEN
174600              BANDERA:=FALSE;
174700          IF BANDERA THEN
174800              BEGIN
174900                  WRITELN('ESTE CUBO SI SE PUEDE COMBINAR, INTENTALO');
175000                  BANDERA:=FALSE;
175100              END
175200          ELSE
175300              BEGIN
175400                  IF CONT = T THEN
175500                      WRITELN('BIEN, ESTE CUBO NO SE PUEDE COMBINAR')
175600                  ELSE
175700                      WRITELN('BIEN, ESTE CUBO NO SE PUEDE COMBINAR (HASTA',
175800                          ' EL MOMENTO)');
175900                      FOR CONT1:=CONT2 TO (CONT2+NCASILLAS) DO
176000                          CDIGITOSECONT+1,0,CONT1J:='*';
176100                      GRAFIC(CONT,2,CONT7);
176200                  END
176300              END
176400          END
176500          ELSE
176600              IF (T = CONT) OR (T = CONT+1) THEN
176700                  IF CDIGITOSECT,0,CONT2J = '*' THEN
176800                      WRITELN('OBSERVA LOS GRUPOS, YA SE HABIA VISTO QUE ',
176900                          'ESTE CUBO NO SE COMBINA')
177000                  ELSE
177100                      BEGIN
177200                          FOR CONT1:=CONT2 TO (CONT2+NCASILLAS) DO
177300                              CDIGITOSECT,0,CONT1J:='*';
177400                          IF T = CONT THEN
177500                              WRITELN('BIEN, ESTE CUBO NO SE PUEDE COMBINAR')
177600                          ELSE
177700                              WRITELN('BIEN, ESTE CUBO NO SE PUEDE COMBINAR (HASTA',
177800                                  ' EL MOMENTO)');
177900                              GRAFIC(CONT,2,CONT7);
178000                          END
178100                      ELSE
178200                          WRITELN('ESTE CUBO NO PERTENECE A LOS GRUPOS QUE ESTAMOS',
178300                              ' ANALIZANDO')
178400                      END
178500                  ELSE
178600                      WRITELN('NO EXISTE ESTE CUBO, VERIFICA');
178700                      BANDERA1:=TRUE;
178800                      FOR CONT1:=0 TO SOLUCI.NELEMCCONTJ-1 DO
178900                          .IF NOT ((CDIGITOSECONT,0,CONT1J='C') OR (CDIGITOSECONT,0,CONT1J
179000                              = '*')) THEN
179100                              BANDERA1:=FALSE;
179200                              FOR CONT1:=0 TO SOLUCI.NELEMCCONT+1J-1 DO
179300                                  IF NOT ((CDIGITOSECONT+1,0,CONT1J='C') OR (CDIGITOSECONT+1,0,
179400                                      CONT1J = '*')) THEN
179500                                      BANDERA1:=FALSE;
179500                                  IF CAMBIOCIJ = 'N' THEN
179501
129502

```

```

179503
179504
179600 BEGIN
179700   WRITELN('AUN HAY CUBOS QUE NO SE COMBINAN?');
179800   READLN; READ(CAMBIOC1J,CAMBIOC2J);
179900   WHILE NOT (((CAMBIOC1J = 'S') AND (CAMBIOC2J = 'I')) OR
180000     ((CAMBIOC1J = 'N') AND (CAMBIOC2J = 'O'))) DO
180100     BEGIN
180200       WRITELN('REPITE SI HAY CUBOS POR CUBRIR O NO?');
180300       READLN; READ(CAMBIOC1J,CAMBIOC2J);
180400     END;
180500     IF (CAMBIOC1J='S') AND (NOT BANDERA1) THEN
180600       CAMBIOC1J:='N'
180700     ELSE
180800       IF (NOT BANDERA1) AND (CAMBIOC1J = 'N') THEN
180900         BEGIN
181000           WRITELN('AUN EXISTEN CUBOS POR CUBRIR, OBSERVA LOS GRUPOS');
181100           WRITELN('SE PUEDEN COMBINAR LOS QUE NO TIENEN MARCAS?');
181200           READLN; READ(CAMBIOC1J,CAMBIOC2J);
181300           WHILE NOT (((CAMBIOC1J = 'S') AND (CAMBIOC2J = 'I')) OR
181400             ((CAMBIOC1J = 'N') AND (CAMBIOC2J = 'O'))) DO
181500             BEGIN
181600               WRITELN('REPITE SI HAY CUBOS QUE SE COMBINEN O NO?');
181700               READLN; READ(CAMBIOC1J,CAMBIOC2J);
181800             END;
181900           END
182000         ELSE
182100           BEGIN
182200             IF (CAMBIOC1J = 'S') AND BANDERA1 THEN
182300               WRITELN('YA CUBRISTE TODOS LOS CUBOS, SOLO GENERARIAS',
182400                 'COMBINACIONES ');
182500             ELSE
182600               WRITELN('BIEN, COMPLETASTE CORRECTAMENTE EL ANALISIS',
182700                 'DE ESTOS GRUPOS');
182800             IF SOLUCI.NELEM[CONTJ] <> 0 THEN
182900               WRITELN('TE PRESENTARE LOS SIGUIENTES');
183000             CAMBIOC1J:='F';
183100           END;
183200         END;
183300       END;
183400     END;
183500   END
183600   ELSE
183700     BEGIN
183800       BANDERA1:=TRUE;
183900       FOR CONT1:=0 TO SOLUCI.NELEM[CONTJ]-1 DO
184000         IF NOT ((CDIGITOSECONT,0,CONT1]='C') OR (CDIGITOSECONT,0,CONT1
184100           = '*')) THEN
184200           BANDERA1:=FALSE;
184300         FOR CONT1:=0 TO SOLUCI.NELEM[CONTJ]-1 DO
184400           IF NOT ((CDIGITOSECONT+1,0,CONT1]='C') OR (CDIGITOSECONT+1,0,
184500             CONT1 = '*')) THEN
184600             BANDERA1:=FALSE;
184700           IF NOT BANDERA1 THEN
184800             BEGIN
184900               WRITELN('AUN TIENES CUBOS QUE CUBRIR, INTENTALO');
185000               WRITELN('SE PUEDEN COMBINAR LOS QUE NO TIENEN MARCAS?');
185100               READLN; READ(CAMBIOC1J,CAMBIOC2J);
185200               WHILE NOT (((CAMBIOC1J = 'S') AND (CAMBIOC2J = 'I')) OR
185300                 ((CAMBIOC1J = 'N') AND (CAMBIOC2J = 'O'))) DO
185400                 BEGIN
185500                   WRITELN('REPITE SI HAY CUBOS POR COMBINAR O NO?');
185600                   READLN; READ(CAMBIOC1J,CAMBIOC2J);
185700                 END;
185701
1857A2

```



```

185703
185704
185800          END
185900          ELSE
186000          IF BANDERA1 THEN
186100              CAMBIOCIJ:='F';
186200          END;
186300          END;
186400          UNTIL CAMBIOCIJ = 'F';
186500          END
186600          ELSE
186700              IF ((SOLUCI.NELEMCONTJ> 0) AND (SOLUCI.NELEMCONT-1J = 0)) THEN
186800                  BEGIN
186900                      CONT7:=CONT MOD 10;
187000                      GRAFIC(CONT,1,CONT7);
187100                      WRITELN('POR NO HABER UN GRUPO ANTERIOR NI UNO POSTERIOR ',
187200                          'CON EL CUAL ESTE SE PUEDA');
187300                      WRITELN('COMBINAR, SE MARCARAN TODOS SUS CUBOS COMO IMPLI',
187400                          'CANTES');
187500                      FOR CONT1:=0 TO SOLUCI.NELEMCONTJ-1 DO
187600                          CDIGITOSECONT,0,CONT1:='*';
187700                          GRAFIC(CONT,1,CONT7);
187800                          CONT:=CONT+1;
187900                      END;
188000          END; (*FOR*)
188100          IF CONT6=0 THEN
188200              CONT6:=CONT6+11
188300          ELSE
188400              CONT6:=CONT6+10;
188500          END; (* WHILE *)
188600          FOR CONT:=0 TO 30 DO
188700              BEGIN
188800                  CUBALUMCONTJ:=' ';
188900                  MARCASECONTJ:=' ';
189000              END;
189100          WRITELN;
189200          WRITELN('CONTINUACION TE PRESENTO LA TABLA DE IMPLICANTES ',
189300              'PRIMOS, LLENA EL');
189400          WRITELN('ULTIMO RENGLON, SI EXISTE SOLO UNA MARCA EN UNA COLU',
189500              'MNA, MARCA ESTA Y');
189600          WRITELN('TODAS AQUELLAS QUE CONTENGAN A LOS MINTERMINOS DE ESE ',
189700              'IMPLICANTE CON *0*');
189800          WRITELN('SI NO SE CUMPLE LO ANTERIOR MARCALA CON EL CARACTER *B*');
189900          IMPRTABLAS(SOLUCI.NTABIMP,DIGITOS,SOLUCI.CUBIERTO,NDIGITOS,
190000              NIMPLIC,MARCAS,CUBALUM,CUBO);
190100          WRITELN;
190200          CUBREMIN(SOLUCI.CUBIERTO,NIMPLIC+1,NDIGITOS,DIGITOS);
190300          WRITELN('BIEN, TU TABLA CON EL ULTIMO RENGLON ES:');
190400          IMPRTABLAS(SOLUCI.NTABIMP,DIGITOS,SOLUCI.CUBIERTO,NDIGITOS,
190500              NIMPLIC,MARCAS,CUBALUM,CUBO);
190600          BANDERA:=TRUE;
190700          FOR CONT:=0 TO NDIGITOS DO
190800              IF SOLUCI.CUBIERTOENIMPLIC+1,CONTJ-<>' ' THEN
190900                  BANDERA:=FALSE;
191000          IF NOT BANDERA THEN
191100              BEGIN
191200                  WRITELN;
191300                  WRITELN('AHORA DAME LOS IMPLICANTES PRIMOS ESCENCIALES, SEPARADOS ',
191400                      'POR UN ESPACIO');
191500                  IMPLICESEN(CUBO,SOLUCI.MARC,NIMPLIC);
191600                  WRITELN('BIEN, MARCASTE CORRECTAMENTE LOS IMPLICANTES PRIMOS',
191700                      'ESCENCIALES');
191800                  WRITELN;
191900                  WRITELN('LA TABLA COMPLETA ES:');
191901
191902

```

```

191903
191904
192000 IMPRTABLAS(SOLUCI,NTABIMP,DIGITOS,SOLUCI,CUBIERTO,NDIGITOS,
192100 NIMPLIC,MARCAS,CUBALUM,CUBO);
192200 END;
192300 IF BANDERA2 THEN
192400 BEGIN
192500 WRITELN;
192600 WRITELN('COMO NO SE CUBRIERON TODOS LOS MINTERMINOS, HAY ',
192700 'IMPLICANTES PRIMOS SECUN');
192800 WRITELN('DARIOS, A CONTINUACION TE PRESENTO LA TABLA');
192900 WRITELN;
193000 FOR CONT:=0 TO 30 DO
193100 BEGIN
193200 CUBALUM[CONT]:=' ';
193300 MARCAS[CONT]:=' ';
193400 END;
193500 IMPRTABLAS(SOLUCI,NTABSEC,DIGSEC,SOLUCI,NOCUBIERTO,NDIGSEC,
193600 NIMPSEC,MARCAS,CUBALUM,CUBOS);
193700 WRITELN;
193800 WRITELN('USAREMOS UN METODO LLAMADO DE PETRICK PARA RESOLVER ',
193900 'LA TABLA DE IMPLICAN');
194000 WRITELN('TES PRIMOS SECUNDARIOS, A CONTINUACION TE MUESTRO ',
194100 'PASO A PASO EL PROCEDI');
194200 WRITELN('MIENTO PARA HACERLO');
194300 BANDTOTAL:=TRUE;
194400 CUBRESECUN;
194500 END;
194600 IF NSECUN = 0 THEN
194700 IF SOLUCI.NSOLTOTAL[1] = 0 THEN
194800 WRITELN('LA SOLUCION ES UNA TAUTOLOGIA, ES DECIR, "1"')
194900 ELSE
195000 BEGIN
195100 WRITELN('LA SOLUCION EN FUNCION DE LOS CUBOS ES:');
195200 CONT1:=0;
195300 FOR CONT1:=0 TO NIMPLIC DO
195400 IF SOLUCI.MARCC[CONT1] = '*' THEN
195500 BEGIN
195600 IF CONT1 > 0 THEN
195700 WRITE('+');
195800 WRITE(CUBO[CONT1:1]);
195900 CONT1:=CONT1+1;
196000 END;
196100 WRITELN;
196200 WRITELN('LA SOLUCION OPTIMA EN FUNCION DE LAS VARIABLES ES:');
196300 FOR CONT:=0 TO (NSOLUCIONES[0]-1) DO
196400 WRITE(SOLUCIONES[CONT:1]);
196500 WRITELN;
196600 WRITELN('PARA TODAS LAS FUNCIONES');
196700 WRITELN('DEBIDO A QUE SE ELIMINARON LAS VARIABLES QUE A ',
196800 'CONTINUACION TE PRESENTO');
196900 WRITELN('MARCADAS CON "X" DE CADA CUBO');
197000 CONT2:=1; CONT10:=0;
197100 FOR CONT:=1 TO NIMPLIC DO
197200 IF SOLUCI.MARCC[CONT] = '*' THEN
197300 BEGIN
197400 CONT1:=0; VARELIM(CUBO[CONT],CONT1,CONT2);
197500 CONT2:=CONT2+1;
197600 END;
197700 FOR CONT:=0 TO CONT10-1 DO BEGIN
197800 FOR CONT1:=0 TO (NVARIA+3) DO
197900 WRITE(SOLUCIONES[CONT,CONT1:1]);
198000 WRITELN;
198100 END;
198101
198102

```

```

198103
198104
198200     END
198300 ELSE
198400 BEGIN
198500     FOR CONT:=0 TO NSECUN DO
198600     BEGIN
198700         FOR CONT1:=0 TO NSOLUCIONES[CONT] DO
198800             SOLUCI.SOLTOTAL[CONT,CONT1]:=SOLUCIONES[CONT,CONT1];
198900             SOLUCI.NSOLTOTAL[CONT]:=NSOLUCIONES[CONT];
199000     END;
199100     WRITELN('CADA UNO DE ESTOS CUBOS QUE FORMAN LAS SOLUCIONES',
199200             ' OPTIMAS, SE PRESENTAN');
199300     WRITELN('A CONTINUACION, EN ELLOS TE MARCO CON UNA X LAS',
199400             ' VARIABLES QUE FORMAN');
199500     WRITELN('LA SOLUCION, LAS QUE SE MANTIENEN, TIENEN SU VALOR',
199600             ' ASOCIADO');
199700     CONT3:=1;
199800     FOR CONT:=0 TO NSECUN DO
199900     IF MARCAMIN[CONT] = '*' THEN
200000     BEGIN
200100         WRITELN('PARA LA SOLUCION ',CONT3:1);
200200         CONT2:=0; CONT10:=1;
200300         FOR CONT1:=1 TO NIMPLIC DO
200400             IF SOLUCI.MARCCONT1:= '*' THEN
200500                 VARELIM(CUBO[CONT1],CONT2,CONT);
200600         FOR CONT1:=0 TO SOLUCI.NMATSOL[CONT]-1 DO
200700             VARELIM(CUBOS[SOLUCI.MATSOL[CONT,CONT1]],CONT2,CONT);
200800         FOR CONT1:=1 TO CONT10-1 DO
200900         BEGIN
201000             FOR CONT2:=0 TO (NVARIA+3) DO
201100                 WRITE(SOLUCIONES[CONT1,CONT2]:1);
201200             WRITELN;
201300         END;
201400         CONT3:=CONT3+1;
201500     END;
201600     CONT1:=0;
201700     FOR CONT:=0 TO NSECUN DO
201800     IF MARCAMIN[CONT] = '*' THEN
201900         CONT1:=CONT1+1;
202000     IF CONT1 > 1 THEN
202100         WRITELN('LA SOLUCION OPTIMA PUEDE SER CUALQUIERA DE LAS',
202200             ' SIGUIENTES')
202300     ELSE
202400         WRITELN('LA SOLUCION OPTIMA ES:');
202500     FOR CONT:=0 TO NSECUN DO
202600     IF MARCAMIN[CONT] = '*' THEN
202700     BEGIN
202800         FOR CONT1:=0 TO SOLUCI.NSOLTOTAL[CONT] DO
202900         BEGIN
203000             IF (CONT1 > 72) AND (SOLUCI.SOLTOTAL[CONT,CONT1] = '+') THEN
203100                 WRITELN;
203200                 WRITE(SOLUCI.SOLTOTAL[CONT,CONT1]:1);
203300             END;
203400             WRITELN;
203500         END;
203600     END;
203700 END.
#

```

R TESIS/QUINE

#RUNNING 4760

ESTE PROGRAMA TE GUIARA EN LA MINIMIZACION DE FUNCIONES POR MEDIO DEL METODO DE QUINE-McCLUSKEY

DESEAS CONOCER UNA PEQUEÑA INTRODUCCION A ESTE METODO?

#?

SI

LOS MINTERMINOS DE UNA FUNCION SON LLAMADOS CUBOS-0.

EL METODO CONSISTE EN UNA BUSQUEDA EXHAUSTIVA DE TODAS LAS COMBINACIONES POSIBLES DE LOS CUBOS-N PARA FORMAR LOS CUBOS-(N+1) Y LUEGO SELECCIONAR LA COMBINACION MINIMA DE CUBOS PARA REALIZAR LA FUNCION.

SE INICIA OBTENIENDO LA REPRESENTACION BINARIA DE LOS CUBOS-0 Y SE AGRUPAN EN FUNCION DEL NUMERO DE "1 S".

LOS CUBOS-1 SE OBTIENEN COMPARANDO GRUPOS CONSECUTIVOS POR PARES. EJEMPLO:

GRUPO 1	GRUPO 2
001 1	011 3
010 2	110 6

SI COMPARAMOS UN CUBO DEL GRUPO 2 CON OTRO DEL GRUPO 1 Y LA DIFERENCIA ES POTENCIA DE 2 HEMOS OBTENIDO UN CUBO-1.

$$\text{CUBO-0}(\text{GRUPO 2}) - \text{CUBO-0}(\text{GRUPO 1}) = 3 - 2 = 1$$

OBSERVA QUE SE HA HECHO LA DIFERENCIA DEL GRUPO MAYOR - GRUPO MENOR, LO CONTRARIO NO ES CIERTO.

ESTE PROCEDIMIENTO SE CONTINUA HASTA CUBRIR TODOS LOS CUBOS DE TODOS LOS GRUPOS.

PARA OBTENER LOS CUBOS-2 HASTA CUBOS-N DEBES OBSERVAR LA MISMA CONDICION DE DIFERENCIA DE POTENCIA DE 2 ENTRE LOS CUBOS DE CADA GRUPO CONSECUTIVO EN SUS ELEMENTOS CORRESPONDIENTES Y ADEMAS QUE LA(S) DIFERENCIA(S) ANTERIOR(ES), LA CUAL TE PRESENTARE EN ESTE PROGRAMA ENTRE PARENTESIS SEA LA MISMA.

CONTINUA...

SI ALGUN CUBO DE ALGUN GRUPO NO SE COMBINA, LO MARCARAS, ESTE FORMARA PARTE DE LOS IMPLICANTES PRIMOS.

LOS IMPLICANTES PRIMOS SON TODOS AQUELLOS CUBOS QUE DURANTE EL PROCESO DE COMPARACION NO LOGRAN COMBINARSE, DE ENTRE ELLOS SE SELECCIONARAN AQUELLOS QUE NOS DARAN LA SOLUCION MINIMA.

ESTOS IMPLICANTES SE ORDENAN EN UNA TABLA EN FUNCION DE LA CANTIDAD DE CUBOS-0 QUE CONTENGAN.

ESTA TABLA TENDRA EN SUS RENGLONES LOS IMPLICANTES ORDENADOS EN FUNCION DE LA CANTIDAD DE CUBOS-0 Y EN SUS COLUMNAS TODOS LOS CUBOS-0.

DE CADA IMPLICANTE SE BUSCARAN EN LAS COLUMNAS, LOS CUBOS-0 QUE LO FORMAN Y SE MARCARA LA COLUMNA CORRESPONDIENTE.

QUE LE PERTENEZCAN.

EL PASO SIGUIENTE ES VERIFICAR CADA COLUMNA QUE CONTENGA SOLO UNA MARCA, CUANDO SE ENCUENTREN, SE MARCARAN EN UN RENGLON ADICIONAL AL IGUAL QUE TODAS AQUELLAS QUE ESTEN MARCADAS EN EL RENGLON QUE SE ENCONTRO SOLO UNA MARCA, ESTE IMPLICANTE SE CONVIERTE EN ESCENCIAL Y FORMARA PARTE DE LA SOLUCION.

SI AL TERMINAR DE REVISAR TODAS LAS COLUMNAS Y MARCADO LOS IMPLICANTES ESCENCIALES HAY COLUMNAS QUE NO SE MARCARON, PASAREMOS ESTAS COLUMNAS A OTRA TABLA ASI COMO LOS IMPLICANTES QUE LAS CONTENGAN.

CONTINUA...

EXISTEN DIVERSOS METODOS PARA RESOLVER ESTA ULTIMA TABLA, EN ESTE PROGRAMA LO HAREMOS A TRAVES DE UN METODO LLAMADO DE PETRICK, EL CUAL CONSISTE EN TOMAR LOS IMPLICANTES QUE FORMAN ESTA TABLA E INTERPRETARLAS COMO VARIABLES BOOLEANAS, ESTO ES TOMAR VALORES CIERTO O FALSO, DEPENDIENDO DE SI SE ESCOGE O NO PARA EL ANALISIS.

A PARTIR DE LA PRIMERA COLUMNA SE DEBEN SELECCIONAR CUALQUIERA DE LOS IMPLICANTES QUE TENGAN MARCA EN ELLA PARA CUBRIRLO DE MANERA QUE SI REPRESENTAMOS CADA IMPLICANTE POR UNA LETRA Y *a* ES EL PRIMER IMPLICANTE Y *d* EL TERCERO DE LA TABLA SECUNDARIA, Y AMBOS TIENEN MARCADOS EN ALGUNA DE LAS COLUMNAS, SE EXPRESARAN POR:

$$(a+d)=1$$

YA QUE CUALQUIERA DE LOS DOS CUBRE ESE CUBO, DEBIDO A TODOS ELLOS DEBEN CUBRIRSE, SE UNIRAN POR EL CONECTOR "Y", ESTO ES, SI OTRA COLUMNA CONTIENE POR EJEMPLO LOS CUBOS *c* y *f* MARCADOS, SE TENDRA LA EXPRESION:

$$(a+b)(c+f)=1$$

ESTE PROCESO SE SIGUE HASTA INVOLUCRAR TODAS LAS COLUMNAS.

EN SEGUIDA SE REALIZARAN LOS PRODUCTOS INDICADOS Y SE SIMPLIFICARA LA EXPRESION RESULTANTE POR MEDIO DE LOS TEOREMAS FUNDAMENTALES DEL ALGEBRA DE BOOLE, POR LO MENOS ALGUNO DE LOS PRODUCTOS DEBE SER CIERTO (1), DE ESTOS SE SELECCIONA ALGUNO CUYO COSTO SEA MENOR, ESTO ES, EL NUMERO DE ENTRADAS ES MAS REDUCIDO.

LOS CUBOS QUE CONTENGA ESTE PRODUCTO MAS LOS IMPLICANTES DE LA PRIMERA TABLA, DAN UNA SOLUCION MINIMA.  
CONTINUA...

AHORA BIEN CUANTAS VARIABLES TIENE TU FUNCION?

4

LAS VARIABLES DEFINIDAS EN PESO DE MAYOR A MENOR SON:  
DCBA

PROPORCIONA LOS MINTERMINOS EN FORMA DECIMAL SEPARADOS POR COMAS Y TERMINA CON EL CARACTER (!)  
0,2,8,12,13!

TUS MINTERMINOS (CUBOS-0) ORDENADOS EN FUNCION DEL NUMERO DE UNOS CON SU REPRESENTACION BINARIA Y DECIMAL SON:

GRUPO 0	GRUPO 1	GRUPO 2	GRUPO 3
0000 0	0010 2	1100 12	1101 13
	1000 8		

DE AQUI EN ADELANTE, USAREMOS LOS MINTERMINOS EN SU REPRESENTACION DECIMAL

ANALISIS DE LOS CUBOS 0 PARA OBTENER LOS CUBOS 1

GRUPO 0	GRUPO 1
0	2
	8

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA POTENCIA DE 2?

SI

DAME UNA PAREJA DE ELLOS

0,2)

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 0      GRUPO 1  
0C            2C  
                 8

AUN HAY CUBOS POR COMBINAR?  
SI

DAME UNA PAREJA DE ELLOS  
0,8;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 0      GRUPO 1  
0C            2C  
                 8C

AUN HAY CUBOS POR COMBINAR?  
NO

BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS  
TE PRESENTARE LOS SIGUIENTES

GRUPO 1      GRUPO 2  
2C            12  
                 8C

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA  
POTENCIA DE 2?  
SI

DAME UNA PAREJA DE ELLOS  
8,12;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 1      GRUPO 2  
2C            12C  
                 8C

AUN HAY CUBOS POR COMBINAR?  
NO

BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS  
TE PRESENTARE LOS SIGUIENTES

GRUPO 2      GRUPO 3  
12C           13

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA  
POTENCIA DE 2?  
SI

DAME UNA PAREJA DE ELLOS  
12,13;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 2      GRUPO 3  
12C           13C

AUN HAY CUBOS POR COMBINAR?  
NO

BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS

ANALISIS DE LOS CUBOS 1 PARA OBTENER LOS CUBOS 2

PARA QUE SE PUEDAN COMBINAR CUBOS DE LOS SIGUIENTES GRUPOS, SE DEBE CUMPLIR CON:

A) LA(S) DIFERENCIA(S) INDICADA(S) EN EL PARENTESIS DEBE(N) SER LA(S) MISMA(S), YA QUE CORRESPONDE A LA MISMA VARIABLE ELIMINADA

B) LA(S) DIFERENCIA(S) ENTRE LOS ELEMENTOS CORRESPONDIENTES ENTRE LOS CUBOS DE DIFERENTES GRUPOS DEBE SER IGUAL Y POTENCIA DE 2

GRUPO 1	GRUPO 2
0, 2 (2)	8, 12 (4)
0, 8 (8)	

EXISTEN EN ESTOS GRUPOS CUBOS QUE SE COMBINEN?

SI

OBSERVA BIEN LOS GRUPOS! NO SE PUEDEN COMBINAR

DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA

0, 2;

BIEN, ESTE CUBO NO SE PUEDE COMBINAR

GRUPO 1	GRUPO 2
0, 2 (2)*	8, 12 (4)
0, 8 (8)	

AUN HAY CUBOS QUE NO SE COMBINAN?

SI

DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA

0, 8;

BIEN, ESTE CUBO NO SE PUEDE COMBINAR

GRUPO 1	GRUPO 2
0, 2 (2)*	8, 12 (4)
0, 8 (8)*	

AUN HAY CUBOS QUE NO SE COMBINAN?

SI

DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA

8, 12;

BIEN, ESTE CUBO NO SE PUEDE COMBINAR (HASTA EL MOMENTO)

GRUPO 1	GRUPO 2
0, 2 (2)*	8, 12 (4)*
0, 8 (8)*	

AUN HAY CUBOS QUE NO SE COMBINAN?

NO

BIEN, COMPLETASTE CORRECTAMENTE EL ANALISIS DE ESTOS GRUPOS  
 TE PRESENTARE LOS SIGUIENTES

GRUPO 2	GRUPO 3
8,12 (4)*	12,13 (1)

EXISTEN EN ESTOS GRUPOS CUBOS QUE SE COMBINEN?

NO

HUBO CUBOS QUE NO SE PUDIERON COMBINAR?

SI

DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA

12,13†

BIEN, ESTE CUBO NO SE PUEDE COMBINAR (HASTA EL MOMENTO)

GRUPO 2	GRUPO 3
8,12 (4)*	12,13 (1)*

AUN HAY CUBOS QUE NO SE COMBINAN?

NO

BIEN, COMPLETASTE CORRECTAMENTE EL ANALISIS DE ESTOS GRUPOS

A CONTINUACION TE PRESENTO LA TABLA DE IMPLICANTES PRIMOS, LLENA EL  
 ULTIMO RENGLON, SI EXISTE SOLO UNA MARCA EN UNA COLUMNA, MARCA ESTA Y  
 TODAS AQUELLAS QUE CONTENGAN A LOS MINTERMINOS DE ESE IMPLICANTE CON "0"  
 SI NO SE CUMPLE LO ANTERIOR MARCALA CON EL CARACTER "B"

	0	2	8	12	13
a				0	0
b			0	0	
c	0	0			
d	0		0		
	0	0	B	0	0

BIEN, TU TABLA CON EL ULTIMO RENGLON ES:

	0	2	8	12	13
a				0	0
b			0	0	
c	0	0			
d	0		0		
	0	0		0	0

AHORA DAME LOS IMPLICANTES PRIMOS ESCENCIALES, SEPARADOS POR UN ESPACIO

a b

NO MARCASTE ALGUN IMPLICANTE ESCENCIAL

a c



BIEN, MARCASTE CORRECTAMENTE LOS IMPLICANTES PRIMOS ESCENCIALES  
 LA TABLA COMPLETA ES:

	0	2	8	12	13
* a				e	e
b			e	e	
* c	e	e			
d	e	e	e		e

COMO NO SE CUBRIERON TODOS LOS MINTERMINOS, HAY IMPLICANTES PRIMOS SECUN  
 DARIOS, A CONTINUACION TE PRESENTO LA TABLA

	8
b	e
d	e

USAREMOS UN METODO LLAMADO DE PETRICK PARA RESOLVER LA TABLA DE IMPLICAN  
 TES PRIMOS SECUNDARIOS, A CONTINUACION TE MUESTRO PASO A PASO EL PROCEDI  
 MIENTO PARA HACERLO

$$(b+d)=1$$

FINALMENTE, SIMPLIFICANDO, SE LLEGA A:

$$b+d=1$$

CADA UNO DE ESTOS CUBOS QUE FORMAN LAS SOLUCIONES OPTIMAS, SE PRESENTAN  
 A CONTINUACION, EN ELLOS TE MARCO CON UNA X LAS VARIABLES QUE FORMAN  
 LA SOLUCION, LAS QUE SE MANTIENEN, TIENEN SU VALOR ASOCIADO

PARA LA SOLUCION 1

$$a = 110X$$

$$c = 00X0$$

$$b = 1X00$$

PARA LA SOLUCION 2

$$a = 110X$$

$$c = 00X0$$

$$d = X000$$

LA SOLUCION OPTIMA PUEDE SER CUALQUIERA DE LAS SIGUIENTES

$$D1C1B0+D0C0A0+D1B0A0$$

$$D1C1B0+D0C0A0+C0B0A0$$

$$\#ET=10:34.7 \text{ PT}=2.5 \text{ IO}=0.2$$

```

FILE 8(Access='SEQUENTIAL',BLOCKS SIZE=1920,MYUSE='IO',
.FORM='FORMATTED',RECL=1920,KIND='REMOTE',STATUS='OLD')
FILE 9(Access='SEQUENTIAL',BLOCKS SIZE=83,MYUSE='OUT',
.FORM='FORMATTED',RECL=83,KIND='PRINTER',STATUS='NEW')
DIMENSION IX(960),M(64),VD(64),IDIR(64),ID(64),LIN(10),LIN(10),
.DIR(15),IVD(1000),TAB(40),MAT2(70),VEC1(64),IP(20),IN2(10),
.ILON(10),MAT(10,64),IN22(4,10),ILON2(4,10),IIM(4),IVEC(64)
COMMON CONTRL(28),CURSOR(80)
COMMON /C/ IMPZ,ICH
COMMON /B/ IC(4,5),ICT(4,5),IG(4,3),IGT(4,3)
CHARACTER CONTRL*1,CURSOR*1
CHARACTER IX=1,B=30,M=6,VD=1,BUF=1,TAB=14,MAT2=30,CONTIN=8,MAT=14
INTEGER DIR,VEC1
DATA CONTRL/Z00,Z03,Z11,Z05,Z08,Z16,Z3C,Z0C,Z0D,Z25,Z13,
.Z0E,Z0F,Z18,Z19,Z3F,Z1E,Z1C,Z1D,Z1F,Z1E,Z12,
.Z27,Z05,Z06,ZE6,ZE7,Z7F/
DATA CURSOR/Z40,Z4F,Z7F,Z78,Z58,Z6C,Z50,Z7D,Z4D,Z5D,Z5C,
.Z4E,Z68,Z60,Z48,Z61,ZF0,ZF1,ZF2,ZF3,ZF4,ZF5,
.ZF6,ZF7,ZF8,ZF9,Z7A,Z5E,Z4C,Z7E,Z6E,Z6F,Z7C,
.ZC1,ZC2,ZC3,ZC4,ZC5,ZC6,ZC7,ZC8,ZC9,ZD1,ZD2,
.ZD3,ZD4,ZD5,ZD6,ZD7,ZD8,ZD9,ZE2,ZE3,ZE4,ZE5,
.ZE6,ZE7,ZE8,ZE9,Z4A,ZF0,Z5A,Z5F,Z6D,Z79,Z81,
.Z82,Z83,Z84,Z85,Z86,Z87,Z88,Z89,Z91,Z92,Z93,
.Z94,Z95,Z96/
DATA B(1:6)='ABCDEF'/
DATA ((IC(I,J),J=1,5),I=1,4)/6,20,34,48,62,9,27,45,63,0,7,31,55
..2*0,6,44,3*0/
DATA ((ICT(I,J),J=1,5),I=1,4)/10,24,38,52,66,10,28,46,64,0,12,36
..6*0,2*0,18,56,3*0/
DATA ((IG(I,J),J=1,3),I=1,4)/1,25,49,1,25,49,1,25,49,1,43,40/
DATA ((IGT(I,J),J=1,3),I=1,4)/3,27,51,5,29,54,9,33,61,16,58,00/
WRITE(8,5)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
5 FORMAT(4A1,'IMPRESION ? SI=1')
READ(8,7)IMPZ
7 FORMAT(11)
CALL NUMVAR(N)
CALL DECIM(N, ID, IFI)
CALL BINARI(N, IFI, ID, M)
CALL MINTER(ID, M, IFI, N)
WRITE(8,48)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
48 FORMAT(5X,4A1,' EN PROCESO ')
CALL SOLUC(N,IFI,M, ID, LIN, LIN, DIR, IVD, TAB, III, MAT2, NOP, NIS,
IN22, ILON2, IMP, IIM, MAT, IVEC)
WRITE(8,49)CONTRL(23),CONTRL(20),CURSOR(1),CURSOR(19)
49 FORMAT(5X,4A1,' FIN DE PROCESO ')
CALL CUBOSO(ID, M, IN22, ILON2, IMP, IIM, IFI, N)
CALL QUINE(N, IMP, ID, IVD, IN22, ILON2, LIN, LIM, IIM, MAT)
CALL IMPLIC(TAB, IVD, ID, III, N)
CALL IMPEXE(ID, IVD, TAB, III, IFI, IVEC)
END
*****
SUBROUTINE NUMVAR(N)
COMMON CONTRL(28),CURSOR(80)
COMMON /B/ IC(4,5),ICT(4,5),IG(4,2),IGT(4,2)
COMMON /C/ IMPZ,ICH
CHARACTER CONTRL*1,CURSOR*1
WRITE(8,5)CONTRL(8)
5 FORMAT(A1)
6 WRITE(8,8)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)
..CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1),CONTRL(19),CONTRL(21),
CONTRL(23),CONTRL(26)
8 FORMAT(4A1,'CUANTAS VARIABLES TIENE LA FUNCION (2 <= N <= 4)')
1,4A1,'N=: ',A1,1X,3A1)
READ(8,9,ERR=10)N
00001100
00000200
00000300
00000400
00000500
00000600
00000800
00000900
00000910
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003110
00003115
00003120
00003130
00003200
00003600
00003700
00003800
00003900
00004000
00004100
00004200
00004300
00004400
00004500
00004600
00004620
00004630
00004700
00004710
00004800
00004900
00005000
00005010
00005100
00005200
00005300
00005400
00005500
00005600
00005700
00005800
00005900

```



	GO TO 24	00015100
22	CONTINUE	00015200
	IF(K.NE.1)GO TO 23	00015300
	A(J:J)=A(J-1:J-1)	00015400
	A(J-1:J-1)=0	00015500
	J=J+1	00015600
23	BF(1:2)=A(J-2:J-1)	00015700
	READ(BF,320)IS	00015800
320	FORMAT(I2)	00015900
	IF(IS.GE.NS)GO TO 26	00016000
24	ICOM=1	00016100
	IIN=0	00016200
	K=0	00016300
	INUM=0	00016400
	NNUM=NNUM+1	00016500
	IF(IFIN.EQ.1)GO TO 60	00016600
	GO TO 50	00016700
25	IF(IIN.EQ.1)GO TO 51	00016800
	IF(A(I:I).EQ.0*.AND.INUM.EQ.0)GO TO 48	00016900
	K=K+1	00017000
	IF(K.GT.2)GO TO 27	00017100
	A(J:J)=A(I:I)	00017200
	J=J+1	00017300
	ICOM=0	00017400
	IZ=0	00017500
	INUM=1	00017600
	GO TO 50	00017700
26	WRITE(8,338)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00017800
	IF(IMPZ.EQ.1)CALL IMPR	00017900
338	FORMAT(1X,4A1,'*** ERROR: UN VALOR DE MINTERMINO SOBREPASA MAXIMO	00018000
	*VALOR')	00018100
	GO TO 6	00018200
27	WRITE(8,340)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00018300
	IF(IMPZ.EQ.1)CALL IMPR	00018400
340	FORMAT(1X,4A1,'*** ERROR: MAS DE DOS DIGITO PARA REPRESENTAR UN NIG	00018500
	*TERMINO')	00018600
	GO TO 6	00018700
48	IZ=1	00018800
	ICOM=0	00018900
	GO TO 50	00019000
49	IF(I.EQ.80.AND.(INUM.EQ.1.OR.IZ.EQ.1))GO TO 19	00019100
	IF(INUM.EQ.0.AND.IZ.EQ.0)GO TO 50	00019200
	IIN=1	00019300
	GO TO 50	00019400
51	WRITE(8,345)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00019500
	IF(IMPZ.EQ.1)CALL IMPR	00019600
345	FORMAT(1X,4A1,'*** ERROR: SE OMITIO UNA COMA')	00019700
	GO TO 6	00019800
50	CONTINUE	00019900
	IF(NNUM.EQ.0)GO TO 6	00020000
	INI=IFI+1	00020100
	IFI=IFI+NNUM	00020200
	IKK=1	00020300
	DO 66 IK=1,NNUM	00020400
	B(IK)=A(IKK:IKK+1)	00020500
66	IKK=IKK+2	00020600
	READ(8,400)(IDD(JJ),JJ=INI,IFI)	00020700
400	FORMAT(I2)	00020800
	GO TO 6	00020900
60	IF(NNUM.EQ.0)GO TO 70	00021000
	INI=IFI+1	00021100
	IFI=IFI+NNUM	00021200
	IKK=1	00021300
	DO 76 IK=1,NNUM	00021400
		00021500
		00021600
		00021700
		00021800
		00021900
		00022000

	B(K)=A(IKK:IKK+1)	00022100
76	IKK=IKK+2	00022200
	READ(B,400)(IDC(JJ),JJ=INI,IFI)	00022300
70	CONTINUE	00022400
500	FORMAT(1X,'ID=',I2)	00022800
	IF(IFI.EQ.1)GO TO 618	00022900
	I=1	00023000
	J=2	00023100
520	IF(IDD(I).NE.IDD(J))GO TO 600	00023200
	IF(J.EQ.IFI)GO TO 535	00023300
	DO 530 K=J,IFI-1	00023400
530	IDD(K)=IDD(K+1)	00023500
535	IFI=IFI-1	00023800
	IF(J.GT.IFI)GO TO 610	00024100
	GO TO 520	00024200
600	J=J+1	00024300
	IF(J.LE.IFI)GO TO 520	00024400
610	I=I+1	00024500
	J=I+1	00024600
	IF(I.LT.IFI)GO TO 520	00024900
618	CONTINUE	00025000
	INNT=IFI-1	00025005
619	IOUR=0	00025010
	DO 620 IL=1,INNT	00025100
	IF(IDD(IL).LE.IDD(IL+1))GO TO 620	00025200
	LTH=IDD(IL)	00025300
	IDD(IL)=IDD(IL+1)	00025310
	IDD(IL+1)=LTH	00025320
	IBUR=1	00025330
620	CONTINUE	00025340
	IF(IBUR.EQ.1)GO TO 619	00025350
	RETURN	00025400
	END	00025500
*****		
	SUBROUTINE BINARI(N,IFI,IO,M)	00025510
	DIMENSION ID(64),M(64)	00025600
	CHARACTER M*6	00025700
	DO 10 I=1,IFI	00025800
	K=N	00026000
	VT=FLOAT(ID(I))	00026100
	M(I)(1:6)='000000'	00026200
	IF(VT.EQ.0.0)GO TO 10	00026300
3	IF(VT.NE.1.0)THEN	00026400
	RR=VT/2.	00026700
	IR=RR	00026800
	RF=RR-IR	00026900
	IF(RF.NE.0.0)M(I)(K:K)='1'	00027000
	VT=IR	00027200
	K=K+1	00027300
	GO TO 3	00027310
	ELSE	00027402
	M(I)(K:K)='0'	00027404
	ENDIF	00027410
10	CONTINUE	00027500
	RETURN	00027600
	END	00027700
*****		
	SUBROUTINE SOLUC(N,IFIN,M,IO,LIN,LIN,DIR,I,VO,TAG,II,MAT2,NOP,NIS,	00027710
	IN22,ILON2,IMP,IIN,MAT,IVEC)	00027800
	DIMENSION M(64),NS(64),ID(64),NZ(19),ILOK(10),IN2(10),INI(10),	00028000
	IFI(10),IP(10),I1(2),MAT(10,64),IVD(1000),TAB(40),	00028100
	IVEC(64),MAT1(70),MAT2(70),LH(10),LIN(10),COS(64),	00028200
	NHI(64),DIR(15),IN22(4,15),ILON2(4,15),IIN(4)	00028400
	INTEGER COS,DIR	00028500

CHARACTER M=6,TT=6,B=3,X=1,ELE=14,MAT=14,UNO=1,MATT=3	00028600
SPA=14,TAO=14,SPA2=3,CC=1,MAT1=62,MAT2=30	00028700
DATA X,UNO,SPA,SPA2,CC/'X','1','	00028800
NOP=0	00028900
DO 10 I=1,IFIN	00029000
INUM=0	00029100
DO 7 J=1,N	00029200
IF(M(I)(J),NE,UNO)GO TO 7	00029300
INUM=INUM+1	00029400
7 CONTINUE	00029500
NS(I)=INUM	00029600
10 CONTINUE	00029700
DO 500 I=1,IFIN	00029800
500 WRITE(6,510)I,NS(I).	00029900
510 FORMAT(1X,'NS',I2,' ',I2)	00030000
IS=IFIN-1	00030100
15 IBUR=0	00030200
DO 20 IK=1,IS	00030300
IF(NS(IK).LE,NS(IK+1))GO TO 20	00030400
IT=NS(IK)	00030500
NS(IK)=NS(IK+1)	00030600
NS(IK+1)=IT	00030700
TT(1:N)=M(IK)(1:N)	00030800
M(IK)(1:N)=M(IK+1)(1:N)	00030900
M(IK+1)(1:N)=TT(1:N)	00031000
T=ID(IK)	00031100
ID(IK)=ID(IK+1)	00031200
ID(IK+1)=T	00031300
IBUR=1	00031400
20 CONTINUE	00031500
IF(IBUR.EQ,1)GO TO 15	00031600
DO 520 I=1,IFIN	00031700
520 WRITE(6,525)M(I)(1:N),ID(I),NS(I)	00031800
525 FORMAT(1X,'M',A6,' ID=',I2,' -NS=',I2)	00031900
IAL=-1	00032000
IPV=0	00032100
DO 30 I=1,IFIN	00032200
IF(IAL.EQ,NS(I))GO TO 26	00032300
IAL=NS(I)	00032400
IF(IPV.EQ,0)GO TO 24	00032500
NZ(1)=NS(1)	00032600
IN2(1)=1	00032700
IPV=1	00032800
IA=1	00032900
ILON(IA)=0	00033000
GO TO 26	00033100
24 IA=IA+1	00033200
ILON(IA)=0	00033300
NZ(IA)=NS(I)	00033400
IN2(IA)=IN2(IA-1)+ILON(IA-1)	00033500
26 ILON(IA)=ILON(IA)+1	00033600
30 CONTINUE	00033700
IAM=IA	00033800
DO 40 I=1,IFIN	00033900
MAT(1,I)(1:14)=SPA	00034000
MAT(1,I)(1:N)=M(I)(1:N)	00034100
WRITE(8,36)I	00034200
36 FORMAT(I3)	00034300
MAT(1,I)(11:13)=B(1:3)	00034400
IDD=IDD+1	00034500
IVD(IDD)=I	00034600
40 CONTINUE	00034700
MAT(1,I)(1:14)=SPA	00034800
DO 550 I=1,IFIN	00034900
	00035000
	00035100
	00035200
	00035300
	00035400

550	WRITE(6,555)MAT(1,I)(1:14),IVD(I)	00035500
555	FORMAT(1X,'MAT =',A14,' IVD=',I2)	00035600
50	IN=0	00035700
	IMP=IMP+1	00035800
	IIM(IMP)=IA	00035900
	ND=IA-1	00036000
	DO 55 II=1,ND+1	00036100
	IN22(IMP,II)=IN2(II)	00036200
	IION2(IMP,II)=IION(II)	00036300
	WRITE(6,545)IN2(II),IION(II),II	00036400
	INI(II)=IN2(II)	00036500
	LIN(IMP)=LIN(IMP)+IION(II)	00036600
55	IFI(II)=IION(II)	00036700
	IF(IMP.EQ.1)THEN	00036800
	LIN(IMP)=1	00036900
	ELSE	00037000
	LIN(IMP)=LIN(IMP-1)+LIN(IMP-1)*(2***(IMP-2))	00037100
	ENDIF	00037200
	DO 1000 IXT=1,IA	00037300
	IX1=INI(IXT)	00037400
	IX2=INI(IXT)+IFI(IXT)-1	00037500
	DO 1000 IXY=IX1,IX2	00037600
1000	WRITE(6,3000)MAT(IMP,IXY)(1:14)	00037700
	IF(IULT.EQ.1)GO TO 3300	00037800
	IA=1	00037900
	IW=0	00038000
	IV=0	00038100
	IBAN3=0	00038200
	DO 300 LK=1,ND	00038300
	IW=0	00038500
	J1=INI(LK)	00038600
	J2=INI(LK)+IFI(LK)-1	00038900
	DO 250 J=J1,J2	00039000
	IF(IMP.EQ.1)GO TO 75	00039100
	ICON=0	00039200
	DO 60 IB=1,N	00039300
	IF(MAT(IMP,J)(IB:IB).NE.X)GO TO 60	00039400
	ICON=ICON+1	00039500
	IP(ICON)=IB	00039600
	IF(ICON.EQ.(IMP))GO TO 65	00039700
60	CONTINUE	00039800
65	CONTINUE	00039900
75	CONTINUE	00040000
	J3=INI(LK+1)	00040100
	J4=INI(LK+1)+IFI(LK+1)-1	00040200
	DO 200 MX=J3,J4	00040300
	IBAN=0	00040400
	IF(IMP.EQ.1)GO TO 85	00040500
	DO 80 I8=1,ICON	00040600
	L1=IP(I8)	00040700
	IF(MAT(IMP,MX)(L1:L1).EQ.X)GO TO 80	00040800
	GO TO 200	00040900
80	CONTINUE	00041000
85	IDIF=0	00041100
	DO 95 L=1,N	00041200
	IF(MAT(IMP,J)(L:L).EQ.MAT(IMP,MX)(L:L))GO TO 95	00041300
	IDIR=L	00041400
	IDIF=IDIF+1	00041500
	IF(IDIF.GT.1)GO TO 200	00041600
95	CONTINUE	00041700
	IF(IW.EQ.1)GO TO 115	00041800
	IF(IV.EQ.1)GO TO 100	00041900
	IION(IA)=0	00042000
	IV=1	

	IN2(IA)=1	00042100
	GO TO 110	00042200
100	IA=IA+1	00042300
	ILOM(IA)=0	00042400
	IN2(IA)=IN2(IA-1)+ILOM(IA-1)	00042500
110	CONTINUE	00042600
115	ELE(1:N)=MAT(IMP,J)(1:N)	00042700
	ELE(IDIR:IDIR)=X	00042800
	IF(ILOM(IA).EQ.0)GO TO 117	00042900
	IV1=IN2(IA)	00043000
	IV2=IN2(IA)+ILOM(IA)-1	00043100
	IBAN=0	00043200
	DO 116 IV=IV1,IV2	00043300
	IF(ELE(1:N).NE.MAT(IMP+1,IV)(1:N))GO TO 116	00043400
	IBAN=1	00043500
	GO TO 117	00043600
116	CONTINUE	00043700
117	MAT(IMP,J)(10:10)='C'	00043800
	MAT(IMP,MX)(10:10)='C'	00043900
	IF(IBAN.EQ.1)GO TO 200	00044000
	ILOM(IA)=ILOM(IA)+1	00044100
	IN=IN+1	00044200
	II(1)=J	00044300
	II(2)=MX	00044400
	IX=IDD+1	00044500
	MAT(IMP+1,IN)(1:14)=SPA	00044600
	MAT(IMP+1,IN)(1:N)=ELE(1:N)	00044700
	IBAN3=1	00044800
	WRITE(6,36)IX	00044900
	MAT(IMP+1,IN)(11:13)=B(1:3)	00045000
	MAT(IMP+1,IN+1)(1:14)=SPA	00045100
	MAT(IMP+2,1)=SPA	00045200
	IJF=2+(IMP-1)	00045300
	IPRR=IDD+1	00045400
	DO 125 IT=1,2	00045500
	MM=II(IT)	00045600
	MAT(1:3)=MAT(IMP,MM)(11:13)	00045700
	READ(MAT,36)IPD	00045800
	DO 122 IM=1,IJF	00045900
	IDD=IDD+1	00046000
	IVD(IDD)=IVD(IPD)	00046100
	IM=1	00046200
	IPD=IPD+1	00046300
122	CONTINUE	00046400
125	CONTINUE	00046500
126	ICAM=0	00046600
	DO 127 IXT=IPRR,IDD-1	00046700
	IF(IVD(IXT).LE.IVD(IXT+1))GO TO 127	00046800
	IVW=IVD(IXT)	00046900
	IVD(IXT)=IVD(IXT+1)	00047000
	IVD(IXT+1)=IVW	00047100
	ICAM=1	00047200
127	CONTINUE	00047300
	IF(ICAM.EQ.1)GO TO 126	00047400
200	CONTINUE	00047500
250	CONTINUE	00047600
300	CONTINUE	00047700
	DO 2000 IXY=1,ND+1	00047800
	IX1=INI(IXY)	00047900
	IX2=INI(IXY)+FI(IXY)-1	00048000
	DO 2000 IX=IX1,IX2	00048100
2000	WRITE(6,3000)MAT(IMP,IXY)(1:14)	00048200
3000	FORMAT(1X,' MAT = ',A14)	00048300
	IF(IA.NE.1)GO TO 50	00048400
		00048500
		00048600
		00048700
		00048800



IF (IBAN3.EQ.0)GO TO 3300	00046900
IULT=1	00049000
GO TO 50	00049100
3300 DO 4000 IXT=1,100	00049500
4000 WRITE(6,5000)IVD(IXT),IXT	00049600
5000 FORMAT(10X,' IVD=',I2,' IXT=',I3)	00049700
5010 II=0	00049800
I=INP+1	00049900
IMEH=1	00050000
6010 J=1	00050100
6020 IF (MAT(I,J)(11:13).EQ.SPA2(1:3))GO TO 6040	00050200
IF (MAT(I,J)(10:10).EQ.CC)GO TO 6030	00050300
II=II+1	00050400
WRITE(6,6027)I	00050500
6027 FORMAT(I1)	00050600
TAB(II)(1:14)=SPA	00050700
TAB(II)(2:2)=B(1:1)	00050800
TAB(II)(3:3+N-1)=MAT(I,J)(1:N)	00050900
TAB(II)(11:13)=MAT(I,J)(11:13)	00051000
WRITE(6,6028)I1,TAB(II)(1:14)	00051100
6028 FORMAT(1X,' I1=',I2,' TAB=',A14)	00051200
6030 J=J+1	00051300
GO TO 6020	00051400
6040 I=I-1	00051500
IF (I.NE.0)GO TO 6010	00051600
DO 6060 I=1,IFIN	00051700
IBAN=0	00051800
DO 6050 J=1,II	00051900
B(1:1)=TAB(J)(2:2)	00052000
READ(B,6027)IK	00052100
IKL=2*(IK-1)	00052200
B(1:3)=TAB(J)(11:13)	00052300
READ(B,36)J1	00052400
J2=J1+IKL-1	00052500
DO 6045 M4=J1,J2	00052600
IF (IVD(I).NE.IVD(M4))GO TO 6045	00052700
JJ=J	00052800
IBAN=IBAN+1	00052900
6045 CONTINUE	00053000
IF (IBAN.GT.1)GO TO 6060	00053100
6050 CONTINUE	00053200
TAB(JJ)(1:1)='*'	00053300
6060 CONTINUE	00053500
DO 6070 I=1,II	00053600
6070 WRITE(6,6028)I,TAB(I)(1:14)	00053700
DO 6080 I=1,II	00053800
IF (TAB(I)(1:1).EQ.'*')GO TO 6080	00053900
NOP=NOP+1	00054000
B(1:1)=TAB(I)(2:2)	00054100
READ(B,6027)IK	00054200
IKL=2*(IK-1)	00054300
B(1:3)=TAB(I)(11:13)	00054400
READ(B,36)J1	00054500
J2=J1+IKL-1	00054600
DO 6075 LJ=J1,J2	00054700
IND1=IVD(LJ)	00054800
6075 IVEC(IND1)=1	00054900
6080 CONTINUE	00055000
DO 6090 I=1,IFIN	00055100
6090 WRITE(6,6092)IVEC(I)	00055200
6092 FORMAT(1X,' IVEC=',I3)	00055300
IXJ=0	00055400
DO 7200 I=1,IFIN	00055500
IF (IVEC(I).EQ.1)GO TO 7200	00055600

	IXJ=IXJ+1	00055700
	IF(IXJ.EQ.1)GO TO 6099	00055800
	WRITE(6,6095)IXJ-1,MAT1(IXJ-1)(1:10)	00055900
6095	FORMAT(3X, I3, 3X, A10)	00056000
6096	FORMAT(I2)	00056100
6099	WRITE(8,6096)I	00056200
	MAT1(IXJ)(1:2)=B(1:2)	00056300
	WRITE(6,7002)IXJ,MAT1(IXJ)(1:2)	00056400
7002	FORMAT(1X, 'IXJ=', I3, 'MAT1=', 2A1)	00056500
	IA=1	00056600
	IU=2	00056700
	DO 7198 J=1, I1	00056800
	IF(TAB(J)(1:1).EQ.'*')GO TO 7198	00056900
	BC(1:3)=TAB(J)(11:13)	00057000
	READ(8,36)IFT	00057100
	BC(1:1)=TAB(J)(2:2)	00057200
	READ(8,6027)ITP	00057300
	ITPP=2*(ITP-1)	00057400
	ITPP=IFT+ITPP-1	00057500
	DO 7100 IJ=IFT,ITPP	00057600
	IF(IVD(IJ).NE.I)GO TO 7100	00057700
	IA=IA+2	00057800
	IU=IU+2	00057900
	WRITE(8,6096)J	00058000
	MAT1(IXJ)(IA:IU)=B(1:2)	00058100
	MAT1(IXJ)(IU+1:IU+2)=SPA2(1:2)	00058200
	WRITE(6,7003)IXJ,MAT1(IXJ)(1:10)	00058300
7003	FORMAT(1X, 'IXJ=', I3, 'MAT1', A10)	00058400
	GO TO 7198	00058500
7100	CONTINUE	00058600
	GO TO 7198	00058700
7198	NOP=NOP+1	00058800
7198	CONTINUE	00058900
7200	CONTINUE	00059000
	IF(IXJ.EQ.0)GO TO 9800	00059100
	IF(IXJ.EQ.1)GO TO 7550	00059200
	MAT1(IXJ+1)(3:6)=''	00059300
	DO 7210 I=1, IXJ	00059400
7210	WRITE(6,7003)I,MAT1(I)(1:10)	00059500
	IXX=0	00059502
	I3=3	00059504
	I2=4	00059506
	IA=3	00059508
	IU=4	00059510
	IXT=2	00059512
	IH=4	00059514
7300	IF(MAT1(1)(I3:I2).NE.SPA2(1:2))THEN	00059516
	IF(MAT1(2)(IA:IU).NE.SPA2(1:2))THEN	00059518
	IXX=IXX+1	00059520
	MAT2(IXX)(1:2)=MAT1(1)(I3:I2)	00059522
	MAT2(IXX)(3:4)=MAT1(2)(IA:IU)	00059524
	MAT2(IXX)(5:6)=SPA2(1:2)	00059526
	IA=IA+2	00059528
	IU=IU+2	00059530
	ELSE	00059532
	I3=I3+2	00059534
	I2=I2+2	00059536
	IA=3	00059538
	IU=4	00059540
	ENDIF	00059542
	ELSE	00059544
	GO TO 7590	00059546
	ENDIF	00059548
	GO TO 7300	00059550

7550	I3=1	00062000
	I2=2	00062100
	IXX=0	00062200
	MAT1(I3)(I3:I4)=*	00062300
7560	I3=I3+2	00062400
	I2=I2+2	00062500
	IF(MAT1(I1)(I3:I2).EQ.SPA2(1:2))GO TO 7590	00062600
	IXX=IXX+1	00062700
	MAT2(IXX)(1:2)=MAT1(I1)(I3:I2)	00062800
	MAT2(IXX)(3:10)=*	00062900
	GO TO 7560	00063000
7590	DO 7620 I=1,IXX	00063100
7620	WRITE(6,7630)I,MAT2(I)(1:6)	00063200
7630	FORHAT(5X,I3,5X,A6)	00063300
	IF(IXJ.EQ.1)GO TO 9035	00063400
7640	IXI=IXI+1	00063500
	I3=3	00063600
	I2=4	00063700
	IF(MAT1(IXI)(I3:I2).NE.SPA2(1:2))GO TO 7700	00063800
	GO TO 8000	00063900
7700	DO 7750 I=1,IXX	00064000
	MAT2(I)(IM+1:IM+2)=MAT1(IXI)(I3:I2)	00064100
7750	MAT2(I)(IM+3:IM+4)=SPA2(1:2)	00064200
	I=IXX	00064300
7800	I3=I3+2	00064400
	I2=I2+2	00064500
	IF(MAT1(IXI)(I3:I2).NE.SPA2(1:2))GO TO 7850	00064600
C	IYY=IXX	00064610
	GO TO 8700	00064700
7850	IYY=I	00064800
C	IMX=IYY+1	00064810
	DO 7900 I=1,IXX	00064900
	IYY=IYY+1	00065000
7900	MAT2(IYY)(1:IM)=MAT2(I)(1:IM)	00065100
	IN=I	00065200
	DO 7950 I=IN,IYY	00065300
	MAT2(I)(IM+1:IM+2)=MAT1(IXI)(I3:I2)	00065400
7950	MAT2(I)(IM+3:IM+4)=SPA2(1:2)	00065500
	GO TO 7800	00065600
8700	IXX=IYY	00065700
	IM=IM+2	00065800
	GO TO 7640	00065900
8800	DO 8890 IL1=1,IXX	00066000
8890	WRITE(6,8895)IL1,MAT2(IL1)(1:IM)	00066100
8895	FORMAT(1X,'IL1=',I3,' MAT2=',A30)	00066200
	DO 9030 I=1,IXX	00066300
	IA=1	00066304
	I2=2	00066306
	I3=3	00066308
	I4=4	00066310
9005	IF(MAT2(I)(I3:I4).EQ.SPA2(1:2))GO TO 9030	00066312
	IF(MAT2(I)(IA:I2).EQ.MAT2(I)(I3:I4))THEN	00066314
9010	MAT2(I)(I3:I4)=MAT2(I)(I3+2:I4+2)	00066316
	IF(MAT2(I)(I3:I4).EQ.SPA2(1:2))GO TO 9015	00066318
	I3=I3+2	00066320
	I4=I4+2	00066322
	GO TO 9010	00066324
	ENDIF	00066326
9015	IA=IA+2	00066328
	I2=I2+2	00066330
	I3=IA+2	00066332
	I4=I2+2	00066334
	GO TO 9005	00066336
9030	CONTINUE	00066338

9035	DD 9200 I=1, IXX	00069300
	IG=1	00069400
	I2=2	00069500
9040	IF(MAT2(I)(IG:I2).EQ.SPA2(I:2))GO TO 9200	00069600
	U(I:2)=MAT2(I)(IG:I2)	00069700
	READ(B,6096)J	00069800
	B(I:1)=TAB(J)(2:2)	00069900
	READ(B,6027)JJ	00070000
	COS(I)=COS(I)+JJ	00070100
	NMI(I)=NMI(I)+1	00070200
	IG=IG+2	00070300
	I2=I2+2	00070400
	GO TO 9040	00070500
9200	CONTINUE	00070600
	MEN=NMI(I)	00070700
	DD 9300 I=2, IXX	00070800
	IF(MEN.LE.NMI(I))GO TO 9300	00070900
	MEN=NMI(I)	00071000
9300	CONTINUE	00071100
	MEN1=5000	00071200
	DD 9400 I=1, IXX	00071300
	IF(NMI(I).NE.MEN)GO TO 9400	00071400
	IF(MEN1.LE.COS(I))GO TO 9400	00071500
	MEN1=COS(I)	00071600
9400	CONTINUE	00071700
	DD 9500 I=1, IXX	00071800
	IF(NMI(I).NE.MEN)GO TO 9500	00071900
	IF(COS(I).NE.MEN1)GO TO 9500	00072000
	KJ=KJ+1	00072100
	DIR(KJ)=I	00072200
9500	CONTINUE	00072300
	GO TO 9920	00072400
9920	DD 9600 I=1, KJ	00073600
9600	WRITE(6,9709)I,DIR(I)	00073700
9700	FORMAT(1X,' DIR',I2,'=',I3)	00073800
	NIS=KJ	00073900
	NOP=NOP+1	00074000
9800	WRITE(6,5003)IMP,IFIN	00074100
5003	FORMAT(1X,'IMP=',I2,' IFIN=',I3,/)	00074105
	DD 5005 I=1, IMP	00074110
5005	WRITE(6,5006)I,IIM(I),LIN(I),LIN(I)	00074115
5006	FORMAT('I=',I2,' IIM=',I2,' LIN=',I3,' LIM=',I3)	00074120
	DD 5008 I=1, IMP	00074125
	IMPI=IIM(I)	00074130
	DD 5008 JIX=1, IMPI	00074135
5008	WRITE(6,5009)I,JIX,IN2(I,JIX),ILON2(I,JIX)	00074137
5009	FORMAT(2X,'I=',I2,' J=',I2,' INN2=',I3,' ILO2=',I3)	00074140
	RETURN	00074145
	END	00074200
*****		
	SUBROUTINE LIMPIA	00074210
	COMMON CONTRL(28),CURSOR(80)	00074300
	CHARACTER CONTRL*1,CURSOR*1	00074400
	CHARACTER BLCK(80)*4	00074500
	DATA BLCK/80*' '/	00074600
	WRITE(8,100)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1),	00074700
	*(BLCK(I),I=1,20),CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19),	00074800
	*(BLCK(I),I=1,80)	00074900
100	FORMAT(4A1,20A4,4A1,80A4)	00075000
	RETURN	00075100
	END	00075200
*****		
	SUBROUTINE MINTER(ODD,M,IFI,N)	00075300
	DIMENSION IOD(64),M(64),IREL(6),ICOL(6)	00075310
		00075400
		00075500

```

COMMON CTRL(28),CURSOR(80)
COMMON /B/ IC(4,5),ICT(4,5),IG(4,2),IGT(4,2)
COMMON /C/ IMPZ,ICH
CHARACTER CTRL*1,CURSOR*1
CHARACTER M*6
DATA (IREL(I),I=1,6)/16,32,48,64,80,96/
DATA (ICOL(I),I=1,6)/3,14,25,36,47,58/
DATA IRI/1/
J=1
ICO=ICOL(J)
WRITE(8,40)CTRL(8)
DO 20 I=1,IFI
10 IRI=IRI+1
IF(IRI.LE.16)GO TO 15
J=J+1
ICO=ICOL(J)
IRI=1
GO TO 10
15 WRITE(8,40)CTRL(23),CTRL(28),CURSOR(ICO),CURSOR(IRI)
, ,IDD(I),M(I)(1:N)
20 CONTINUE
WRITE(8,45)CTRL(23),CTRL(28),CURSOR(1),CURSOR(19)
40 FORMAT(4A1,I2,1X,A6)
45 FORMAT(4A1,' ESTOS SON LOS MINTERMINOS QUE PROPORCIONASTE CON SU EQUIVALENTE EN BINARIO')
IF(IMPZ.EQ.1)CALL INPR
RETURN
END
*****
SUBROUTINE CUROS0(IDD,M,IN22,ILON2,IMP,IIM,IFIN,N)
DIMENSION IDD(64),M(64),IN22(4,10),ILON2(4,10),ICOL(8),IIM(4)
COMMON CTRL(28),CURSOR(80)
COMMON /B/ IC(4,5),ICT(4,5),IG(4,2),IGT(4,2)
COMMON /C/ IMPZ,ICH
CHARACTER CTRL*1,CURSOR*1
CHARACTER M*6,ALFA*6,BLC*7,BETA*8
DATA (ICOL(I),I=1,8)/1,11,21,31,41,51,61,71/
DATA ALFA,BLC,J/'GRUPO ',' ','0/'
WRITE(8,30)CTRL(8),CTRL(23),CTRL(28),CURSOR(1),CURSOR(19)
30 FORMAT(5A1,'ESTOS SON LOS MINTERMINOS EN SU REPRESENTACION BINARIA',
,/,/, 'AGRUPADOS SEGUN NUMERO DE UNOS EN ESA REPRESENTACION')
IAM=IIM(1)
DO 50 I=1,IAM
IRI=20
I1=IN22(1,I)
I2=IN22(1,I)*ILON2(1,I)-1
DO 40 K=I1,I2
IRI=IRI+1
IF(IRI.GT.17)THEN
J=J+1
ICO=ICT(1,J)
ICOL=IC(1,J)
IRI=2
WRITE(8,60)CTRL(23),CTRL(28),CURSOR(ICO),CURSOR(IRI),
ALFA,I
IRI=IRI+1
WRITE(8,62)CTRL(23),CTRL(28),CURSOR(ICO1),CURSOR(IRI),
M(K)(1:N)
ELSE
WRITE(8,62)CTRL(23),CTRL(28),CURSOR(ICO1),CURSOR(IRI),
M(K)(1:N)
ENDIF
40 CONTINUE
50 CONTINUE

```

```

00075600
00075700
00075710
00075800
00075900
00076000
00076100
00076200
00076300
00076400
00076500
00076600
00076700
00076800
00076900
00077000
00077100
00077200
00077300
00077400
00077500
00077600
00077700
00077800
00077900
00077910
00078000
00078100
00078110
00078200
00078300
00078400
00078500
00078510
00078600
00078700
00078800
00078900
00079000
00079100
00079200
00079500
00079600
00079700
00079800
00079900
00080200
00080300
00080400
00080500
00080600
00080700
00080800
00080900
00081000
00081100
00081200
00081300
00081400
00081500
00081600
00081700
00081800
00081900

```

60	FORMAT(4A1,A6,I2)	00082000
62	FORMAT(4A1,A4)	00082100
61	WRITE(8,63)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00082200
63	FORMAT(4A1,'CONTINUA')	00082300
	READ(8,64,ERR=61)BETA	00082400
64	FORMAT(A8)	00082500
	IF(IMPZ.EQ.1)CALL IMPR	00082510
	WRITE(8,83)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00082600
	IBAN=0	00082700
	WRITE(8,66)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00082800
66	FORMAT(4A1,'AHORA SE AGREGA EL EQUIVALENTE DECIMAL DE LOS HINTERMIEN .NOS AGRUPADOS',/,60(1H))	00082900
65	J=3	00083000
	DO 75 I=1,IAM	00083100
	IR1=20	00083200
	I1=IN22(1,I)	00083300
	I2=IN22(1,I)+ILON2(1,I)-1	00083400
	DO 72 K=1,I2	00083500
	IR1=IR1+1	00083600
	IF(IR1.LE.17)GO TO 70	00083700
	J=J+1	00083800
	ICO=ICT(1,J)	00083900
	ICO1=IC(1,J)	00084000
	IR1=3	00084100
70	IF(IBAN.EQ.0)THEN	00084200
	WRITE(8,80)CONTRL(23),CONTRL(28),CURSOR(ICO1+4),CURSOR(IR1), IDDK)	00084300
	ELSE	00084400
	WRITE(8,82)CONTRL(23),CONTRL(28),CURSOR(ICO1),CURSOR(IR1)	00084500
	.,BLC(1:5)	00084600
	ENDIF	00084700
72	CONTINUE	00084800
75	CONTINUE	00084900
81	WRITE(8,63)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00085000
	READ(8,64,ERR=81)BETA	00085100
	IF(IMPZ.EQ.1)CALL IMPR	00085200
	WRITE(8,83)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00085300
83	FORMAT(4A1,'')	00085400
	IF(IBAN.EQ.1)GO TO 90	00085500
	IBAN=1	00085600
	WRITE(8,79)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00085700
79	FORMAT(4A1,'DE AQUI EN ADELANTE SOLO SE TRABAJA CON LA REPRESENTACION .ION DECIMAL')	00085710
	GO TO 65	00085720
80	FORMAT(4A1,'-(',I2,')')	00085800
82	FORMAT(4A1,A5)	00085900
90	CONTINUE	00086000
	RETURN	00086100
	END	00086200
*****		00086300
	SUBROUTINE QUINE(N,IMP,IDD,IVD,IN22,ILON2,LIN,LIM,IH,MAT)	00086400
	DIMENSION IDD(64),IVD(1000),IN22(4,5),ILON2(4,5),LIN(4),LIM(4),	00086500
	MAT(10,64),IEX(4),IIM(4),IPT(2),INUM(2,14,8),RESP(24),IDIF(2,5),	00086600
	RE(24),NST(5)	00086700
	COMMON CONTRL(28),CURSOR(80)	00086800
	COMMON /B/ IC(4,5),ICT(4,5),IG(4,3),IGT(4,3)	00086900
	COMMON /C/ IMPZ,ICH	00087000
	COMMON /D/ A,BETA	00087100
	CHARACTER V*2,A*38,MAT*14,BB*1,B1*10,B2*18,B3*32,BETA*8,ALFA*6	00087200
	CHARACTER RESP*1	00087300
	CHARACTER CONTRL*1,CJRSOR*1	00087400
	INTEGER RE	00087500
	DATA NST(1),I=1,5)/7,13,21,35,0/	00087600
	DATA IEX,IPST/8,4,2,1,0/	00087710

	DATA ALFA/'GRUPO ' /	00087800
	AC(1:1)='('	00087810
	DO 1000 I=1,INP	00087900
	ISTART=LIN(I)	00088000
	LONG=2** (I-1)	00088100
	NGPS=IIN(I)	00088200
	IF(I.EQ.1)GO TO 960	00088300
63	FORMAT(4A1,'CONTINUA')	00088500
64	FORMAT(A8)	00088700
65	FORMAT(5A1,' ')	00088900
61	KHAT=0	00089000
	DO 950 IR=1,NGPS	00089100
	ICO=ICT(I,IR)	00089200
	ICO1=IC(I,IR)	00089300
	WRITE(8,7D)CONTRL(23),CONTRL(28),CURSOR(ICO),CURSOR(2)	00089400
	,ALFA-IR	00089500
70	FORMAT(4A1,A6,I2)	00089600
	IR1=2	00089700
	ITX=2.0** (I-1)	00089800
	IST1=ISTART+IN22(I,IR)*ITX-ITX	00089900
	IST2=IST1+LONG	00090000
	ILF=ILON2(I,IR)	00090100
	DO 900 KJ=1,ILF	00090200
	KHAT=KHAT+1	00090300
	IPV=0	00090400
	KC=-1	00090500
	DO 600 N1=IST1,IST2-1	00090600
	KC=KC+3	00090700
	IM=IVD(N1)	00090800
	IF(IPV.EQ.0)GO TO 610	00090900
610	WRITE(V,620)IDD(IM)	00091000
620	FORMAT(I2)	00091100
	A(KC:KC+1)=V	00091200
	IPV=1	00091300
600	CONTINUE	00091400
	IF(I.EQ.1)THEN	00091500
	A(KC+2:KC+2)=''	00091520
	KC=KC+2	00091530
	ELSE	00091540
	A(KC+2:KC+3)='('	00091560
	KC=KC+2	00091700
	ENDIF	00091710
	IF(I.EQ.1)GO TO 632	00091720
	IPV=0	00091800
	DO 630 II=N1,-1	00091900
	IF(MAT(I,KHAT)(II:II).NE.'X')GO TO 630	00092000
	KC=KC+2	00092100
	IF(IPV.EQ.0)GO TO 625	00092200
	A(KC-1:KC-1)=''	00092300
624	FORMAT(I1)	00092400
625	WRITE(BB,624)IEX(II)	00092500
	A(KC:KC)=BB	00092600
	IPV=1	00092700
630	CONTINUE	00092800
	KC=KC+1	00092900
	A(KC:KC)=''	00093000
632	IR1=IR1+1	00093100
	IF(IPV.EQ.1)THEN	00093102
	IF(MAT(I,KHAT)(1:1).EQ.'C')THEN	00093110
	KC=KC+1	00093120
	A(KC:KC)='C'	00093130
	ENDIF	00093140
	ENDIF	00093145

631	GO TO (655,631,635,638),IMP	00093200
	B1=A(1:KC)	00093300
	WRITE(8,645)CONTRL(23),CONTRL(28),CURSOR(IC01),CURSOR(IR1),	00093400
	81	00093500
	GO TO 640	00093600
635	B2=A(1:KC)	00093700
	WRITE(8,646)CONTRL(23),CONTRL(28),CURSOR(IC01),CURSOR(IR1),	00093800
	82	00093900
	GO TO 640	00094000
638	B3=A(1:KC)	00094100
	WRITE(8,647)CONTRL(23),CONTRL(28),CURSOR(IC01),CURSOR(IR1),	00094200
	83	00094300
	GO TO 640	00094400
655	STOP 44	00094500
645	FORMAT(4A1,A10)	00094600
646	FORMAT(4A1,A18)	00094700
647	FORMAT(4A1,A32)	00094800
640	CONTINUE	00094900
	IST1=IST1+LONG	00095000
900	IST2=IST1+LONG	00095100
950	CONTINUE	00095200
	WRITE(8,63)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00095201
	READ(8,64,ERR=61)BETA	00095202
	IF(IMPZ.EQ.1)CALL INPR	00095203
	WRITE(8,65)CONTRL(8),CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00095204
	IF(IPST.EQ.1)THEN	00095205
	IPST=0	00095210
	GO TO 1999	00095215
	ENDIF	00095220
960	KMAT=0	00095600
	KMAT2=0	00095610
	NGPSW=NGPS-1	00095700
	IFF=0	00095800
	IF(NGPSW.NE.0)GO TO 2100	00095900
	NGPSW=1	00096000
	IFF=1	00096100
2100	IR=0	00096105
2102	IR=IR+1	00096110
	IF(IR.GT.NGPSW)GO TO 4001	00096200
	WRITE(8,65)CONTRL(8)	00096210
	IPASO=0	00096300
	IRV=0	00096400
	IRR=IR	00096500
	IRF=1	00096600
2200	IC0=IGT(I,IRF)	00096700
	IC01=IG(I,IRF)	00096800
	WRITE(8,70)CONTRL(23),CONTRL(28),CURSOR(IC0),CURSOR(2)	00096900
	,ALFA,IRR	00097000
	IR1=2	00097100
	ITX=2.0**(I-1)	00097200
	IST1=ISTART+IN22(I,IRR)*ITX-ITX	00097300
	IST2=IST1+LONG	00097400
	ILF=ILON2(I,IRR)	00097500
	IPY(IRF)=ILF	00097600
	DO 2900 KJ=1,ILF	00097700
	KMAT=KMAT+1	00097800
	IPV=0	00097900
	KC=1	00098000
	KRF=0	00098100
	DO 2600 N1=IST1,IST2-1	00098200
	KC=KC+3	00098300
	IN=IVD(N1)	00098400
	KRF=KRF+1	00098500
	INUM(IRF,KJ,KRF)=IDD(IN)	00098600



	IF(IPV.EQ.0)GO TO 2610	00098700
	A(KC-1:KC-1)=''	00098800
2610	WRITE(V,620)IDD(IM)	00098900
	A(KC:KC+1)=V	00099000
	IPV=1	00099100
2600	CONTINUE	00099200
	A(KC+2:KC+3)='')'	00099300
	KC=KC+2	00099400
	IPV=0	00099500
	IF(I.NE.1)GO TO 2620	00099600
	A(KC+1:KC+1)=''	00099700
	GO TO 2631	00099800
2620	DD 2630 II=N,1,-1	00099900
	IF(KMAT(I,KMAT)(II:II).NE.'X')GO TO 2630	00100000
	KC=KC+2	00100100
	IF(IPV.EQ.0)GO TO 2625	00100200
	A(KC-1:KC-1)=''	00100300
2625	WRITE(00,624)IEX(II)	00100400
	A(KC:KC)=00	00100500
	IPV=1	00100600
2630	CONTINUE	00100700
	KC=KC+1	00100800
	A(KC:KC)='')	00100900
2631	IR1=IR1+1	00101000
	GO TO (2655,2632,2635,2638),IMP	00101100
2632	B1=A(1:KC)	00101200
	WRITE(0,2645)CONTRL(23),CONTRL(28),CURSOR(IC01),CURSOR(IR1),	00101300
	CONTRL(20),CONTRL(21),B1	00101400
	GO TO 2640	00101500
2635	B2=A(1:KC)	00101600
	WRITE(0,2646)CONTRL(23),CONTRL(28),CURSOR(IC01),CURSOR(IR1),	00101700
	CONTRL(20),CONTRL(21),B2	00101800
	GO TO 2640	00101900
2638	B3=A(1:KC)	00102000
	WRITE(0,2647)CONTRL(23),CONTRL(28),CURSOR(IC01),CURSOR(IR1),	00102100
	CONTRL(20),CONTRL(21),B3	00102200
	GO TO 2640	00102300
2645	FORMAT(5A1,1X,A1,A10)	00102400
2646	FORMAT(5A1,1X,A1,A10)	00102500
2647	FORMAT(5A1,1X,A1,A32)	00102600
2655	STOP 444	00102700
2640	CONTINUE	00102800
	IST1=IST1+LONG	00102900
2900	IST2=IST1+LONG	00103000
	IF(IFF.EQ.1)GO TO 4000	00103100
	IF(IRV.EQ.1)GO TO 2940	00103200
	IRF=2	00103300
	IRV=1	00103400
	IRR=IRR+1	00103500
	KMAT=KMAT	00103600
	GO TO 2200	00103700
2940	IR1=IR1+2	00103707
	WRITE(0,2647)CONTRL(23),CONTRL(28),CURSOR(IC01),CURSOR(IR1)	00103708
	CONTRL(20),CONTRL(21)	00103709
	IF(IMPZ.EQ.1)CALL IMPR	00103711
	IST2=LIN(I+1)	00103720
	IC02=IGT(I,3)	00103730
	IC012=IG(I,3)	00103735
	IR2=2	00103740
	I2T=I+1	00103745
	ITX2=2.0**(I2T-1)	00103750
	IST11=IST2+IN22(I2T,IRR-1)*ITX2-ITX2	00103755
	IST22=IST11+ITX2	00103760
	II=0	00103800

	J=)	00103900
2945	II=II+1	00104000
	J=0	00104100
	IF(II.GT.IPT(1))GO TO 3000	00104200
2950	J=J+1	00104300
	INEM=0	00104400
	JNEM=0	00104500
	IF(J.GT.IPT(2))GO TO 2945	00104600
	IF(I.EQ.1)THEN	00104700
	IF(INUM(1,II,1).GT.INUM(2,J,1))GO TO 2950	00104800
	IDF=ABS(INUM(2,J,1)-INUM(1,II,1))	00104900
	DO 2955 IXM=1,4	00105000
	IVXT=2.0**(IXM-1)	00105100
	IF(IDF.EQ.IVXT)GO TO 2957	00105200
2955	CONTINUE	00105300
	GO TO 2950	00105400
2957	CONTINUE	00105500
	ELSE	00105600
	IB=1	00105700
	IW=I-1	00105800
	ICT3=0	00105900
	ITEM=II	00106000
	ITEMP=II	00106100
2960	DO 2970 IWT=1,IW	00106200
	ICT3=ICT3+1	00106300
	JEX=2.0**(ICT3-1)	00106400
	IDIF(IB,ICT3)=ABS(INUM(IB,ITEM,1)-INUM(IB,ITEM,1+JEX))	00106500
2970	CONTINUE	00106600
	IF(IB.EQ.2)GO TO 2980	00106700
	IB=2	00106800
	ICT3=0	00106900
	ITEM=J	00107000
	GO TO 2960	00107100
2980	DO 2990 IWT=1,IW	00107200
	IF(IDIF(1,IWT).NE.IDIF(2,IWT))GO TO 2950	00107300
	ICON=2.0**(I-1)	00107400
	DO 2985 I1P=1,ICON	00107500
	DO 2985 I2P=1,ICON	00107600
	IF(INUM(2,ITEM,I1P).LT.INUM(1,ITEMP,I2P))GO TO 2950	00107700
2985	CONTINUE	00107800
2990	CONTINUE	00107900
	ENDIF	00108000
	IPASO=1	00108100
	JNEM=J	00108200
	INEM=II	00108300
3000	NDE=IPT(1)+IPT(2)	00108700
3010	CALL LIMPIA	00108710
	WRITE(8,3012)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1),CONTRL(23),	00108800
	CONTRL(26)	00108900
3012	FORMAT(4A1,'DATOS',2A1)	00109000
	READ(8,3015)(RESP(KR),KR=1,NDE)	00109100
3015	FORMAT(24A1)	00109200
	IPP=1	00109300
	JPP=1	00109400
	ICON=0	00109500
	JCON=0	00109600
	KR=1	00109700
3020	IF(KR.GT.NDE)GO TO 3025	00109800
	IF(IPP.LE.IPT(1))THEN	00109900
	IF(RESP(KR).NE.' ')THEN	00110000
	ICON=ICON+1	00110100
	IP0=IPP	00110200
	IPP=IPP+1	00110300
	ELSE	00110400

	IPP=IPP+1	00110500
	ENDIF	00110600
	KR=KR+1	00110700
	ENDIF	00110800
	IF(JPP.LE.(PT(2)))THEN	00110900
	IF(RESPC(KR).NE.'*')THEN	00111000
	JCON=JCON+1	00111100
	JPO=JPP	00111200
	JPP=JPP+1	00111300
	ELSE	00111400
	JPP=JPP+1	00111500
	ENDIF	00111600
	KR=KR+1	00111700
	ENDIF	00111800
	GO TO 3020	00111900
3025	CONTINUE	00112000
	CALL DIAG1 (ICON,JCON,IPASO,IMEN,JMEN,IP0,JPO,IND1)	00112100
	IF(IMPZ.EQ.1)CALL IMPR	00112200
	GO TO (3010,3990,3901),IND1	00112300
3901	JM=JM+1	00112400
	IRRT=IP0+2	00112500
	DO 3903 KIX=1,2	00112600
	NOS=NST(I)	00112700
	ICDT=IG(I,KIX)	00112800
	WRITE(8,3902)CONTRL(23),CONTRL(28),CURSOR(ICDT+NOS),CURSOR(IRRT)	00112900
3902	FORNAT(4A1,'C')	00113000
3903	IRRT=JPO+2	00113100
	KMAT2=KMAT2+1	00113200
	IPV2=0	00113300
	KC2=-1	00113400
	DO 3920 N1=IST11,IST22-1	00113500
	KC2=KC2+3	00113600
	IF(N1.LE.0.OR.N1.GT.100)STOP 1	00113700
	IM2=IV0(N1)	00113800
	IF(IPV2.EQ.0)GO TO 3916	00113900
	A(KC2-1:KC2-1)='',	00114000
3916	IF(IM2.LE.0.OR.IM2.GT.64)THEN	00114100
	WRITE(8,9980)IM2,N1,IST11,IST22,IPV2	00114200
9988	FORNAT(/,'I3','I3','I3','I3','I3','I3','I3',	00114210
	'IPV2',I4)	00114220
	STOP 2	00114230
	ENDIF	00114240
	WRITE(V,620)I0D(IM2)	00114250
	A(KC2:KC2+1)=V	00114300
	IPV2=1	00114400
3920	CONTINUE	00114500
	A(KC2+2:KC2+3)='')(	00114600
	KC2=KC2+2	00114700
	IPV2=0	00114800
	DO 3940 III=N+1,-1	00114900
	IF(I2T.LE.0.OR.I2T.GT.10)STOP 3	00115000
	IF(KMAT2.LE.0.OR.KMAT2.GT.64)STOP 4	00115100
	IF(KMAT(I2T,KMAT2)(III:III).NE.'*')GO TO 3940	00115200
	KC2=KC2+2	00115300
	IF(IPV2.EQ.0)GO TO 3925	00115400
	A(KC2-1:KC2-1)='',	00115500
	IF(III.LE.0.OR.III.GT.4)STOP 5	00115600
3925	WRITE(80,624)IEX(III)	00115700
	A(KC2:KC2)=88	00115800
	IPV2=1	00115900
3940	CONTINUE	00116000
	KC2=KC2+1	00116100
	A(KC2:KC2)='')	00116200
	IR2=IR2+1	00116300
		00116400

83=A(1:K2)	00116500
IF(IC012.LE.0.OR.IC012.GT.80)STOP 6	00116600
IF(IR2.LE.0.OR.IR2.GT.80)STOP 7	00116700
WRITE(8,3942)CONTRL(23),CONTRL(28),CURSOR(IC012),CURSOR(IR2)	00116800
83	00116900
IST11=IST11+ITX2	00117000
IST22=IST11+ITX2	00117100
IF(IMPZ.EQ.1)CALL IMPR	00117200
GO TO 2950.	00117300
3942 FORMAT(4A1,A32)	00117400
3990 KNAT=KMATH	00117500
4000 GO TO 2102	00117600
4001 CONTINUE	00117605
IPST=1	00117700
GO TO 61	00117800
1000 CONTINUE	00117900
WRITE(8,1005)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(23)	00118000
1005 FORMAT(4A1)	00118100
END	00118200
*****	00118210
SUBROUTINE IMPR	00118300
COMMON /C/ IMPZ,ICH	00118310
COMMON CONTRL(28),CURSOR(80)	00118400
DIMENSION A(26),LEE(1920)	00118500
CHARACTER CONTRL*1,CURSOR*1	00118600
CHARACTER A*82,LEE*1	00118700
WRITE(8,3)CONTRL(23),CONTRL(28),CURSOR(77),CURSOR(24)	00118800
CONTRL(14),CONTRL(16),CONTRL(17)	00118900
3 FORMAT(6A1,'I',A1)	00119000
DO 5 I=1,82	00119100
A(I)(1:1)='*'	00119200
5 A(26)(1:1)='*'	00119300
DO 10 I=2,25	00119400
A(I)(1:1)='*'	00120100
10 A(I)(82:82)='*'	00120200
READ(8,30)(LEE(I),I=1,1920)	00120500
30 FORMAT(160(A1,A1,A1,A1,A1,A1,A1,A1,A1,A1,A1,A1))	00120600
J=1	00120700
DO 40 I=2,25	00120800
DO 35 JK=2,81	00120900
A(I)(JK:JK)=LEE(J)(1:1)	00121000
J=J+1	00121100
35 CONTINUE	00121200
40 CONTINUE	00121300
ICH=ICH+1	00121310
IF(ICH.GT.2)THEN	00121320
WRITE(9,44)	00121330
44 FORMAT(1H1)	00121340
ICH=1	00121350
ENDIF	00121360
WRITE(9,45)	00121400
45 FORMAT(////)	00121500
DO 60 I=1,26	00121600
60 WRITE(9,70)A(I)(1:82)	00121700
70 FORMAT(6X,A82)	00121800
WRITE(8,75)CONTRL(23),CONTRL(28),CURSOR(77),CURSOR(24)	00121900
75 FORMAT(4A1,' ')	00122000
RETURN	00122100
END	00122200
*****	00122210
SUBROUTINE IMPLIC(TA8,IVD,IDD,II,N)	00122300
COMMON CONTRL(28),CURSOR(80)	00122400
COMMON /C/ IMPZ,ICH	00122500
COMMON /D/ A,BETA	00122600

DIMENSION TAB(40),IVD(1000),IDD(64),MINT(17),IEX(4)	00122700
CHARACTER A*38,DALF*43,D*3,BETA*8,IALF*1,TAB*14,TABB*14,SPA*43	00122800
CHARACTER CONTRL*1,CURSOR*1	00122900
DATA INDW,IBN,IR1/2*0,5/,IEX/8,4,2,1/	00123000
DATA DALF(7:7),DALF(9:9),DALF(8:8),DALF(43:43)/2** **,2** **/	00123010
DATA SPA/**	00123012
3 FORMAT(A1)	00123015
4 WRITE(8,5)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00123100
5 FORMAT(4A1,'CONTINUA')	00123200
READ(8,6,ERR=7)BETA	00123300
6 FORMAT(A8)	00123400
GO TO 9	00123410
7 WRITE(8,8)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00123500
8 FORMAT(4A1,'')	00123600
GO TO 4	00123700
9 WRITE(8,3)CONTRL(8)	00123800
WRITE(8,10)CONTRL(23),CONTRL(28),CURSOR(25),CURSOR(5)	00123900
10 FORMAT(4A1,36('**'))	00124000
DO 100 I=11,1,-1	00124100
IALF=TAB(I)(2:2)	00124200
B(1:3)=TAB(I)(11:13)	00124300
TABB=TAB(I)(3:14)	00124310
READ(8,11)IDIR	00124400
11 FORMAT(I3)	00124500
READ(IALF,11)IND	00124600
111 FORMAT(I1)	00124610
IF(INO.NE.INDW)THEN	00124700
IR1=IR1+1	00124800
WRITE(8,12)CONTRL(23),CONTRL(28),CURSOR(25),CURSOR(IR1)	00124900
12 FORMAT(4A1,'**')	00125000
DALF(1:5)='CUBD**'	00125100
IT=IND*1	00125200
WRITE(IALF,11)IF	00125300
DALF(6:6)=IALF	00125400
CALL DCUB(IND,IDIR,KC,IJ,MINT,IBN,IVD,IDD,N,TABB)	00125500
IBN=1	00125600
DALF(10:10+KC-1)=A(1:KC)	00125700
DALF(10+KC:42)=SPA(KC:42)	00125710
IR1=IR1+1	00125800
WRITE(8,13)CONTRL(23),CONTRL(28),CURSOR(18),CURSOR(IR1),DALF	00125900
13 FORMAT(4A1,A43)	00126000
DALF(1:6)=''	00126100
INDW=IND	00126200
ELSE	00126300
CALL DCUB(IND,IDIR,KC,IJ,MINT,IBN,IVD,IDD,N,TABB)	00126400
DALF(10:10+KC-1)=A(1:KC)	00126500
DALF(10+KC:42)=SPA(KC:42)	00126510
IR1=IR1+1	00126600
WRITE(8,13)CONTRL(23),CONTRL(28),CURSOR(18),CURSOR(IR1),DALF	00126700
ENDIF	00126800
100 CONTINUE	00126900
IR1=IR1+1	00126910
WRITE(8,12)CONTRL(23),CONTRL(28),CURSOR(25),CURSOR(IR1)	00127000
IR1=IR1+1	00127010
WRITE(8,10)CONTRL(23),CONTRL(28),CURSOR(25),CURSOR(IR1)	00127100
IF(IHPZ.EQ.1)CALL IMPR	00127110
RETURN	00127200
END	00127300
*****	00127310
SUBROUTINE DCUB(ICUB,IDIR,KC,IJ,MINT,IBN,IVD,IDD,N,TABB)	00127400
COMMON /D/ A,BETA	00127500
DIMENSION IVD(1000),IDD(64),MINT(16),IEX(4)	00127600
CHARACTER V*2,A*38,BETA*8,TABB*14	00127700
DATA IEX/8,4,2,1/	00127710

	LONG=2.0*(ICUB=1)	00127800
	IFIN=IDIR*LONG-1	00127900
	IPV=0	00128000
	KC=-1	00128100
	DO 40 N1=IDIR,IFIN	00128200
	KC=KC+3	00128300
	IM2=IVD(N1)	00128400
	IF(IGN.EQ.0)THEN	00128500
	IJ=IJ+1	00128600
	MINT(IJ)=IDD(IM2)	00128700
	ELSE	00128800
	DO 20 IC=1,IJ	00128900
	IF(MINT(IC).EQ.IDD(IM2))GO TO 30	00129000
20	CONTINUE	00129100
	IJ=IJ+1	00129200
	MINT(IJ)=IDD(IM2)	00129300
30	CONTINUE	00129400
	ENDIF	00129410
	IF(IPV.EQ.0)GO TO 35	00129500
	A(KC-1:KC-1)=','	00129600
35	WRITE(V,36)IDD(IM2)	00129700
36	FORMAT(I2)	00129800
	A(KC:KC+1)=V	00129900
	IPV=1	00130000
40	CONTINUE	00130100
	IF(LONG.EQ.1)THEN	00130110
	A(KC+2:KC+2)=','	00130120
	ELSE	00130130
	A(KC+2:KC+3)=','	00130200
	ENDIF	00130210
	KC=KC+2	00130300
	IF(LONG.EQ.1)GO TO 70	00130303
	IPV=0	00130310
	DO 60 III=N,1,-1	00130315
	IF(TAB8(III:III).NE.'X')GO TO 60	00130320
	KC=KC+2	00130325
	IF(IPV.EQ.0)GO TO 50	00130330
	A(KC-1:KC-1)=','	00130335
50	WRITE(V,51)IEX(III)	00130340
51	FORMAT(I1)	00130345
	A(KC:KC)=V(1:1)	00130350
	IPV=1	00130355
60	CONTINUE	00130360
	KC=KC+1	00130365
	A(KC:KC)=','	00130370
70	CONTINUE	00130380
	RETURN	00130400
	END	00130500
*****		
	SUBROUTINE IMPEXE(ID,IVD,TAB,II,IFI,IVEC)	00130510
	COMMON CONTRL(28),CURSOR(80)	00130600
	COMMON /C/ IMPZ,ICH	00130700
	DIMENSION IO(64),IVD(100),TAB(40),IDN(64),ALFA(10),ND(64),	00130710
	IRP(8),AST(16),GUION(64),IRES(16),BLK(32),MM(16),	00130800
	IVEC(64),IVECN(32),TAB2(16)	00130900
	CHARACTER TAB=14,CONTRL=1,CURSOR=1,0=43,ALFA=1,AST=1,GUION=1,	00130910
	IRES=1,BLK=1,BETA=8,TAB2=64,BB=2	00131000
	DATA NIVN,IR1/2=0/	00131100
	DATA (ALFA(I),I=1,10)/'A','B','C','D','E','F','G','H','I','J'/	00131200
	DATA AST,GUION,BLK/16,'',',',64,'',',',32,'',','/	00131300
	4 WRITE(8,5)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00131400
	5 FORMAT(4A1,'CONTINUA')	00131500
	READ(8,6,ERR=7)BETA	00131550
	6 FORMAT(A8)	00131600
		00131650

	GO TO 9	00131700
7	WRITE(8,8)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)	00131750
8	FORMAT(4A1,' ')	00131800
	GO TO 4	00131850
9	ITAB2=0	00131900
	DO 10 I=1,IFI	00131950
	IF(IVEC(I).EQ.0)ITAB2=1	00132000
	IVECW(I)=IVEC(I)	00132050
10	IDW(I)=ID(I)	00132100
	WRITE(6,5000)(ID(I),I=1,IFI)	00132150
	WRITE(6,5010)(IVEC(I),I=1,IFI)	00132200
5000	FORMAT(1X, 20I2)	00132250
5010	FORMAT(1X, 20I2)	00132300
12	IBAN=0	00132350
	DO 15 I=1,IFI-1	00132400
	IF(IDW(I).GT.IDW(I+1))THEN	00132450
	IT=IDW(I+1)	00132500
	ITT=IVECW(I+1)	00132550
	IDW(I+1)=IDW(I)	00132600
	IVECW(I+1)=IVECW(I)	00132650
	IDW(I)=IT	00132700
	IVECW(I)=ITT	00132750
	IBAN=1	00132800
	ENDIF	00132850
15	CONTINUE	00132900
	IF(IBAN.EQ.1)GO TO 12	00132950
	WRITE(6,5000)(IDW(I),I=1,IFI)	00133000
	WRITE(6,5010)(IVECW(I),I=1,IFI)	00133050
	NG=IFI+4+1	00133100
	WRITE(8,17)CONTRL(8),CONTRL(23),CONTRL(28),CURSOR(8),CURSOR(2),	00133150
	(GUION(I),I=1,NG)	00133200
17	FORMAT(65A1)	00133250
	WRITE(8,20)CONTRL(23),CONTRL(28),CURSOR(8),CURSOR(3),	00133300
	(IDW(I),I=1,IFI)	00133350
20	FORMAT(4A1,'*',16(I3,'*'))	00133400
	WRITE(8,98)CONTRL(23),CONTRL(28),CURSOR(60),CURSOR(18),CONTRL(20),	00133450
	CONTRL(21)	00133500
98	FORMAT(4A1,'EXT',A1,1X,A1)	00133550
	IF(ITAB2.EQ.1)THEN	00133600
	IR2=2	00133650
	I2=5	00133700
	TAB2(2)(1:4)=' '*	00133750
	DO 100 I=1,IFI	00133800
	IF(IVECW(I).EQ.0)THEN	00133850
	ICON=ICON+1	00133900
	WRITE(88,91)IDW(I)	00133950
91	FORMAT(I2)	00134000
	TAB2(2)(I2:I2+1)=DB(I:2)	00134050
	TAB2(2)(I2+2:I2+2)='*'	00134100
	I2=I2+3	00134150
	ENDIF	00134200
100	CONTINUE	00134250
	NG1=(ICON+4+1)+2	00134300
	TAB2(1)(1:2)=' '*	00134350
	DO 105 I=3,NG1	00134400
105	TAB2(1)(I:I)='*'	00134450
	ENDIF	00134500
	DO 500 I=1,II	00134550
	WRITE(8,58)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00134600
58	FORMAT(4A1,60(1H))	00134650
	B(1:1)=TAB(I)(2+2)	00134700
	READ(8,60)NIVL	00134750
60	FORMAT(I1)	00134800
	IF(NIVL.NE.NIVW)THEN	00134850

```

IF(ITAB2.EQ.1)THEN
  I03=1
  IR2=IR2+1
  TAB2(IR2)(1:2)=* *
  DO 62 IJ=3,NG1
    TAB2(IR2)(IJ:IJ)=*-*
62  ENDIF
  IR1=IR1+1
  WRITE(8,17)CONTRL(23),CONTRL(28),CURSOR(8),CURSOR(IR1+3)
  , (GUION(K),K=1,NG)
  NIVM=NIVL
  ENDIF
  IR1=IR1+1
  IEXP=NIVL-1
  ILOM=2.0**IEXP
  B(1:3)=TAB(I)(11:13)
  READ(8,70)IPI
70  FORMAT(I3)
  DO 80 K=1, ILOM
  NI=IVD(IPI)
  ND(K)=ID(NI)
80  IPI=IPI+1
  IST=0
  DO 85 K=1, ILOM
  DO 85 L=1, IFI
  IF(ND(K).EQ.IDM(L))THEN
    IST=IST+1
    IRP(IST)=L
  ENDIF
85  CONTINUE
  IF(ITAB2.EQ.1)THEN
    IF(TAB(I)(1:1).EQ.* *)THEN
      I03=0
      IR2=IR2+1
      TAB2(IR2)(1:1)=ALFA(I)
      TAB2(IR2)(2:3)=* *
      I2=4
    ENDIF
    WRITE(8,90)CONTRL(23),CONTRL(28),CURSOR(12),CURSOR(19),ALFA(I)
    ,(ND(K),K=1,ILOM)
90  FORMAT(4A1,'IMPLICANTE PRIMO ',A1,3X,16I3)
  WRITE(8,95)CONTRL(23),CONTRL(28),CURSOR(6),CURSOR(IR1+3),ALFA(I)
  ,(CONTRL(2),CONTRL(21),AST(K),K=1,IFI)
95  FORMAT(4A1,A1,1X,'*',16(A1,1X,A1,A1))
97  WRITE(8,17)CONTRL(23),CONTRL(26)
  READ(8,17)(IRES(K),K=1,IFI)
  WRITE(8,99)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(21)
99  FORMAT(4A1,80(1H))
  CALL DIAG2(IFI,IRES,MH,ILOM,IRP,INDI)
  IF(IMPZ.EQ.1)CALL IMPR
  IF(IND1.EQ.1)GO TO 97
  IN=1
  INF=4
  ISTD=1
  DO 128 J=1,IFI
  I02=0
  B(IN:INF)=* *
  IF(IRP(ISTC).EQ.J)THEN
    I02=1
    B(INF-2:INF-2)=*1*
    ISTD=ISTC+1
  ENDIF
  IN=INF+1

```

```

00134501
00134503
00134504
00134505
00134506
00134507
00134508
00134510
00134600
00134700
00134800
00134900
00134910
00135000
00135100
00135200
00135300
00135400
00135500
00135600
00135700
00135800
00135900
00136000
00136100
00136200
00136300
00136400
00136500
00136600
00136605
00136610
00136612
00136615
00136620
00136625
00136630
00136635
00136640
00136700
00136800
00136900
00137000
00137100
00137200
00137300
00137400
00137500
00137600
00137700
00137800
00137900
00140110
00140120
00140130
00140140
00140142
00140145
00140150
00140152
00140155
00140160
00140165
00140170

```



INF=INF+4	00140175
IF (ITAB2.EQ.1) THEN	00140180
IF (TAB(I)(1:1).EQ.'*') THEN	00140185
IF (IVECW(J).EQ.0) THEN	00140190
IF (ID2.EQ.1) THEN	00140192
TAB2(IR2)(I2:I2+3)=' 1 '*	00140195
ELSE	00140197
TAB2(IR2)(I2:I2+3)=' '*	00140198
ENDIF	00140199
I2=I2+4	00140200
ENDIF	00140205
ENDIF	00140210
128 CONTINUE	00140215
WRITE(8,135)CONTRL(23),CONTRL(28),CURSOR(9),CURSOR(IR1+3),	00140220
(BK:K),K=1,IF1+4)	00140225
135 FORMAT(4A1,43A1)	00140230
500 CONTINUE	00140400
WRITE(8,58)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19)	00140500
IR1=IR1+1	00140510
WRITE(8,17)CONTRL(23),CONTRL(28),CURSOR(8),CURSOR(IR1+3),	00140700
(GUION(K),K=1,NG)	00140800
IF (ITAB2.EQ.1.AND.ID3.EQ.0) THEN	00140900
IR2=IR2+1	00140901
TAB2(IR2)(1:2)=' '	00140902
DO 505 I=3,NG1	00140903
505 TAB2(IR2)(I:1)='*'	00140904
ENDIF	00140905
IR1=0	00140906
IST=0	00141000
DO 520 I=1,II	00141100
IF (TAB(I)(1:1).EQ.'*') THEN	00141200
IST=IST+1	00141300
IRP(IST)=I	00141400
ENDIF	00141500
520 CONTINUE	00141600
NIVW=0	00141700
B(1:1)=TAB(1)(2:2)	00141710
READ(8,60)NIVL	00141800
DO 540 I=1,II	00141900
B(1:1)=TAB(1)(2:2)	00142000
READ(8,60)NIVL	00142100
IF (NIVL.NE.NIVW) THEN	00142200
IR1=IR1+1	00142300
NIVW=NIVL	00142400
ENDIF	00142500
IR1=IR1+1	00142600
WRITE(8,521)CONTRL(23),CONTRL(28),CURSOR(3),CURSOR(IR1+3),	00142700
CONTRL(20),CONTRL(21)	00142800
521 FORMAT(5A1,1X,A1)	00142900
540 CONTINUE	00143000
542 WRITE(8,17)CONTRL(23),CONTRL(26)	00143100
READ(8,17)(IRES(K),K=1,II)	00143600
WRITE(8,99)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(21)	00143700
CALL DIAG3(II,IRP,IRES,MM,IST,IND1)	00143800
IF (IMPZ.EQ.1) CALL IMPR	00143810
IF (IND1.EQ.1) GO TO 542	00143820
IR1=0	00143830
NIVW=0	00146396
B(1:1)=TAB(1)(2:2)	00146397
READ(8,60)NIVL	00146398
DO 562 I=1,II	00146399
B(1:1)=TAB(1)(2:2)	00146400
READ(8,60)NIVL	00146401
	00146402

```

IF(NIVL.NE.NIVW)THEN                                00146403
  IR1=IR1+1                                          00146404
  NIVW=NIVL                                          00146405
ENDIF                                                00146406
IR1=IR1+1                                           00146407
IF(TAB(I)(1:1).EQ.'*')THEN                          00146408
  B(1:3)=' ' ' ' ' '                               00146409
ELSE                                                 00146410
  B(1:3)=' ' ' ' ' '                               00146411
ENDIF                                                00146412
WRITE(8,561)CONTRL(23),CONTRL(28),CURSOR(3),CURSOR(IR1+3),B(1:3) 00146413
561 FORMAT(4A1,A3)                                   00146414
562 CONTINUE                                         00146415
IR1=IR1+2                                           00146416
WRITE(8,565)CONTRL(23),CONTRL(28),CURSOR(8),CURSOR(IR1+3),    00146500
      (CONTRL(20),CONTRL(21),AST(K),K=1,IFI)        00146600
565 FORMAT(4A1,'*',16(A1,1X,A1,A1))                00146700
IR1=IR1+1                                           00146800
WRITE(8,17)CONTRL(23),CONTRL(28),CURSOR(8),CURSOR(IR1+3),    00146900
      (GUION(K),K=1,NG)                              00147000
C   WRITE(8,98)CONTRL(23),CONTRL(28),CURSOR(60),CURSOR(18),    00147100
C   CONTRL(20),CONTRL(21)                            00147200
568 WRITE(8,17)CONTRL(23),CONTRL(28)                00147300
READ(8,17)(IRES(K),K=1,IFI)                         00147400
WRITE(8,99)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(21)    00147500
DO 575 I=1,IFI                                       00147600
IF((IRES(I).EQ.' ' .AND. IVECW(I).EQ.0).OR.(IRES(I).NE.' ' .AND.
  IVECW(I).EQ.1))GO TO 575                          00147700
WRITE(8,569)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(21)    00147900
569 FORMAT(4A1,'ERROR: ESTA INCORRECTA TU RESPUESTA')    00148000
IF(IMPZ.EQ.1)CALL IMPR                               00148010
GO TO 568                                             00148100
575 CONTINUE                                         00148300
IF(IMPZ.EQ.1)CALL IMPR                               00148310
IF(ITAB2.EQ.1)THEN                                  00148312
600 WRITE(8,5)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)    00148314
    READ(8,6,ERR=605)BETA                            00148318
    GO TO 610                                          00148320
605 WRITE(8,8)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(1)    00148322
    GO TO 600                                          00148324
610 WRITE(8,8)CONTRL(8)                              00148326
    DO 650 I=1,IR2                                    00148328
650 WRITE(8,17)CONTRL(23),CONTRL(28),CURSOR(6),CURSOR(I+1),  00148330
      (TAB2(I)(K:K),K=1,NG1)
    IF(IMPZ.EQ.1)CALL IMPR                            00148332
ENDIF                                                00148334
RETURN                                               00148400
END                                                  00148500
*****00148505
SUBROUTINE DIAG1(ICON,JCON,IPASO,INEM,JHEM,IPO,JPO,IND1) 00148510
COMMON CONTRL(28),CURSOR(80)                        00148520
CHARACTER CONTRL*1,CURSOR*1                          00148530
IND1=0                                                00148540
IJCON=ICON+JCON                                      00148600
WRITE(8,3027)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(22),    00148700
      IPO,JPO,ICON,JCON                              00148800
3027 FORMAT(4A1,' IPO=',I2,' JPO=',I2,' ICON=',I2,' JCON=',I2) 00148900
IF(IJCON.GT.2)THEN                                   00149000
  WRITE(8,3019)CONTRL(23),CONTRL(28),CURSOR(1),CURSOR(19) 00149100
3019 FORMAT(4A1,'ERROR: MAS DE DOS CUBOS SE ALADOS; MARCA SOLO DOS')00149200
  IND1=1                                              00149400
ELSE                                                  00149500
  IF(IJCON.EQ.0)THEN                                  00149600
    IF(IPASO.EQ.0)THEN                                00149700

```







\* \* \* \* \*

11	1000
2	0010
3	0011
7	0111
8	1000

\* \* \* \* \*

\* ESTOS SON LOS MINITERMINOS QUE PROPORCIONASTE CON SU EQUIVALENTE EN BINARIO \*

\* \* \* \* \*

??I?\*

\* CONTINUA \*

	GRUPO 1	GRUPO 2	GRUPO 3	GRUPO 4
0000	0010	0011	0111	
	1000			

\* \* \* \* \*

\* ESTOS SON LOS MINITERMINOS EN SU REPRESENTACION BINARIA AGRUPADOS SEGUN NUMERO  
\* DE UNOS EN ESA REPRESENTACION \*

\* \* \* \* \*

??I?\*







GRUPO 1  
?\*?( 0)C

GRUPO 2  
?\*?( 2)C  
? ?( 8)

( 0, 2)(2)

? ?

PARA INDICAR QUE SE COMBINAN SE MARCAN (CARACTER "C")

??I?

GRUPO 1  
?\*?( 0)C

GRUPO 2  
? ?( 2)C  
?\*?( 8)

( 3, 2)(2)

? ?

BIEN, ESTOS CUBOS SI SE COMBINAN

??I?





\* GRUPO 2  
\*? ?( 2)C  
\*? ?( 8)

GRUPO 3  
? ?( 3)C  
? ?

( 2, 3)(1)

\*PARA INDICAR QUE SE COMBINAN SE MARCAN (CARACTER "C")

??I?\*

\* GRUPO 2  
\*? ?( 2)C  
\*? ?( 8)

GRUPO 3  
? ?( 3)C  
? ?

( 2, 3)(1)

\*CORRECTO: YA NO EXISTEN CUBOS POR COMBINAR

??I?\*



\* GRUPO 3  
\* ?( 3)C

GRUPO 4  
?( 7)C

( 3, 7)(4)

? ?

\*BIEN, ESTOS CUBOS SI SE COMBINAN

???

\* GRUPO 3  
\* ?( 3)C

GRUPO 4  
?( 7)C

( 3, 7)(4)

? ?

\*CORRECTO: YA NO EXISTEN CUBOS POR COMBINAR

???

\*\*\*\*\*  
\*CONTINUA

GRUPO 1	GRUPO 2	GRUPO 3	GRUPO 4
( 0 )C	( 2 )C	( 3 )C	( 7 )C
	( 8 )C		

\*LOS CUBOS QUE SE COMBINARON APARECEN MARCADOS (CARACTER "C")

\*\*\*\*\*  
\*??I?\*

\*\*\*\*\*  
\*CONTINUA

GRUPO 1	GRUPO 2	GRUPO 3
( 0, 2 )(2)	( 2, 3 )(1)	( 3, 7 )(4)
( 0, 8 )(8)		

\*GRUPOS DE CUBOS-1

\*\*\*\*\*  
\*??I?\*

GRUPO 1  
\*? ?( 0, 2)(2)  
\*? ?( 0, 8)(8)

GRUPO 2  
? ?( 2, 3)(1)  
? ?

??1?

GRUPO 1  
\*? ?( 0, 2)(2)  
\*? ?( 0, 8)(8)

GRUPO 2  
? ?( 2, 3)(1)  
? ?

\*CORRECTO: LOS CUBOS DE ESTOS GRUPOS NO SE COMBINAN

??1?





CONTINUA

GRUPO 1  
( 0, 2)(2)  
( 0, 8)(8)

GRUPO 2  
( 2, 3)(1)

GRUPO 3  
( 3, 7)(4)

LOS CUBOS QUE SE COMBINARON APARECEN MARCADOS (CARACTER "C")

??I?

CUBO-1	- ( 3, 7)(4)
	- ( 2, 3)(1)
	- ( 0, 8)(8)
	- ( 0, 2)(2)

EN EL CUADRO ESTAN LOS IMPLICANTES PRIMOS AGRUPADOS DE ACUERDO A LA CLASE  
DE CUBOS A LA QUE PERTENECEN

??I?

TABLA DE IMPLICANTES PRIMOS

\* 0\* 2\* 3\* 7\* 8\*

A \*? \*? \*? \*? \*? \*? \*? \*?

IMPLICANTE PRIMO A 0 2

EXT? ?

??I?\*

TABLA DE IMPLICANTES PRIMOS

\* 0\* 2\* 3\* 7\* 8\*

A \* 1 \* 1 \* \* \* \*  
B \*? \*? \*? \*? \*? \*? \*? \*?

IMPLICANTE PRIMO B 0 8

EXT? ?

??I?\*

TABLA DE IMPLICANTES PRIMOS

	0*	2*	3*	7*	8*
A	* 1 * 1 *	*	*	*	*
B	* 1 *	*	*	* 1 *	
C	* ? ? * ? * ? * ? * ? * ? * ? *				

IMPLICANTE PRIMO C

2 3

EXT ? ?

??I?

TABLA DE IMPLICANTES PRIMOS

	0*	2*	3*	7*	8*
A	* 1 * 1 *	*	*	*	*
B	* 1 *	*	*	* 1 *	
C	* * 1 * 1 *	*	*	*	*
D	* ? ? * ? * ? * ? * ? * ? * ? *				

IMPLICANTE PRIMO D

3 7

EXT ? ?

??I?

.....  
 \*  
 \* TABLA DE IMPLICANTES PRIMOS  
 \*  
 \* .....

	* 0*	2*	3*	7*	8*
* ? ?A	* 1	* 1	*	*	*
* ?*7B	* 1	*	*	*	* 1
* ? ?C	*	* 1	* 1	*	*
* ?*7D	*	*	* 1	* 1	*

.....

\* IDENTIFICA LAS COLUMNAS QUE CONTENGAN UNA SOLA MARCA  
 \* Y MARCA EL IMPLICANTE PRIMO AL QUE PERTENESCA

EXT? ?

??I?\*

.....

.....  
 \*  
 \* TABLA DE IMPLICANTES PRIMOS  
 \*  
 \* .....

	* 0*	2*	3*	7*	8*
* A	* 1	* 1	*	*	*
* B	* 1	*	*	*	* 1
* C	*	* 1	* 1	*	*
* D	*	*	* 1	* 1	*

.....  
 \* ?I?=? ?\*?I?=??I?=?I?\*

\* IDENTIFICA LAS COLUMNAS QUE CONTENGAN UNA SOLA MARCA  
 \* Y MARCA EL IMPLICANTE PRIMO AL QUE PERTENESCA

EXT? ?

??I?\*

.....

```

-----
* 2*
-----
A * 1 *
C * 1 *
-----

```

\*ESTA EL LA TABLA REDUCIDA DE LA QUE PUEDES OBTENER LOS IMPLICANTES PRINCS  
\*SECUNDARIOS , APLICANOD DOMINIO.

\*USAREMOS UN MCTODO LLAMADO DE PETRICK PARA RESOLVER LA TABLA DE IMPLICANTES  
\*PRIMOS SECUNDARIOS.

$$(A+C)=1$$

\*LA SOLUCION OPTIMA PUEDE SER CUALQUIERA DE LAS SIGUIENTES:

$$F(A,B,C,D) = \overset{-}{A}CD + \overset{-}{B}CD + \overset{-}{A}BD$$

$$F(A,B,C,D) = \overset{-}{A}CD + \overset{-}{B}CD + \overset{-}{A}BC$$

### V.3 MANUAL DE USUARIO

El comando necesario para correr este programa dependera de la maquina que se utilice, por lo que debera investigarse en el lugar correspondiente.

Siempre que se conteste algo a la maquina, debe terminarse presionando la tecla <RETURN>. Si se trata de una decision (SI o NO) deberan proporcionarse los dos caracteres.

De entrada el programa dara la oportunidad de conocer una pequeña introduccion al metodo.

A continuacion pedira el numero de variables que tiene la funcion que se va a minimizar. Pidiendo seguidamente los minterminos (cubos-0) que forman la funcion.

Estos cubos agrupados en funcion del numero de '1's de su representacion binaria se muestran al usuario.

A partir de este momento, al usuario se le presentaran los grupos por parejas indicandole los cubos que se analizan y los que con ellos se tratan de formar.

El analisis de los cubos se divide propiamente en dos pasos:

- 1) Combinar todos los cubos posibles
- 2) Indicar aquellos que no se combinaron.

Para lo cual se preguntara al usuario si hay cubos que se combinen en esos grupos.

1) Si hay cubos que se combinen. Debera darse una pareja de ellos. Ejemplo:

GRUPO 1	GRUPO 2
0,1 (1)	4,6(2)
0,2 (2)	
0,4 (4)	

como a continuacion se indica:

0,2,4,6;

Notese que se proporciona primero el cubo del grupo menor y despues el del grupo mayor; todos los elementos se separaron por comas y se termino con el caracter ";".

2) Existen cubos que no se combinan. Terminados de combinar los cubos que son posibles, el programa preguntara si existen cubos que no se combinan; habiendo contestado afirmativamente, pedira estos cubos, del ejemplo anterior, la forma en que se proporcionara cada uno sera:

0,1;

Se observa que se expresa un cubo por vez.

Todos los cubos que se van combinado son marcados con el caracter 'C' y aquellos que no pueden serlo con '\*'.

Este proceso se repite hasta encontrar todos los cubos(n+1) posibles para cada funcion.

Terminado el analisis de cada una de ellas, se presentan al usuario los implicantes que se acaban de generar y los que se van acumulando de los anteriores.

Cuando se han obtenido todos los implicantes de todas las funciones, el programa arma para el usuario la tabla correspondiente con ellos. Ejemplo:

De una funcion dada por  $F(C;B;A) = S(0,1,3,4,5)$ , su tabla de implicantes es:

	0	1	4	6	7
1a					
1b					
1c					
1d					

Donde:

a = {6,7}  
 b = {4,6}  
 c = {0,4}  
 d = {0,1}



y pedirle llenar el ultimo renglon.

El usuario se posicionara con el cursor en cada columna y la marcara con los caracteres "0" o "1" segun sea el caso. El caracter "b" indica que esa columna se mantiene en blanco.

Verificada la respuesta, se piden los implicantes esenciales los cuales se daran con las letras que los identifican separados por un espacio. Del ejemplo anterior, se daria:

a d

Si existen implicantes primos secundarios, se armara la tabla correspondiente, que sera mostrada al usuario, y procedera a presentarle los pasos que se lleven a cabo para resolverla a traves del metodo de Petrick.

Por ultimo se daran a conocer los implicantes esenciales y secundarios que forman la solucion de cada funcion, asi como la expresion en funcion de las variables A,B,C,.. donde el peso de las variables es  $A \leq B \leq C$ .

Si la variable se acompaãa de "1", se considerara su valor cierto, si se acompaãa de "0", se considerara su valor errado.

Con esto concluye el programa.

## VI. QUINE MC.CLUSKEY MÚLTIPLES SALIDAS

### VI.1 ACCIONES

Este programa es una generalización del que se describe en el capítulo anterior.

Analiza un número determinado de funciones del mismo conjunto de variables tratando de hacerlas compartir sus circuitos lógicos.

De lo anterior se deduce que con ciertas modificaciones, parte del programa principal del método de Quine Mc.Cluskey para una salida, se usa como subrutina del que se presenta en este capítulo.

Solo se describirán las modificaciones que se hicieron en algunas subrutinas; para informarse de la función de aquellas que no se alteraron, referirse a los capítulos IV, V y estas son:

- TEXTO
- VARELIM
- EXPONENTE
- RAIZ
- BUSCA
- GRAFIC
- BURBUJA
- DECIMALES
- RESUELVEAL
- ACONJUNTO
- SECUNDARIOS
- SOLUMAQ

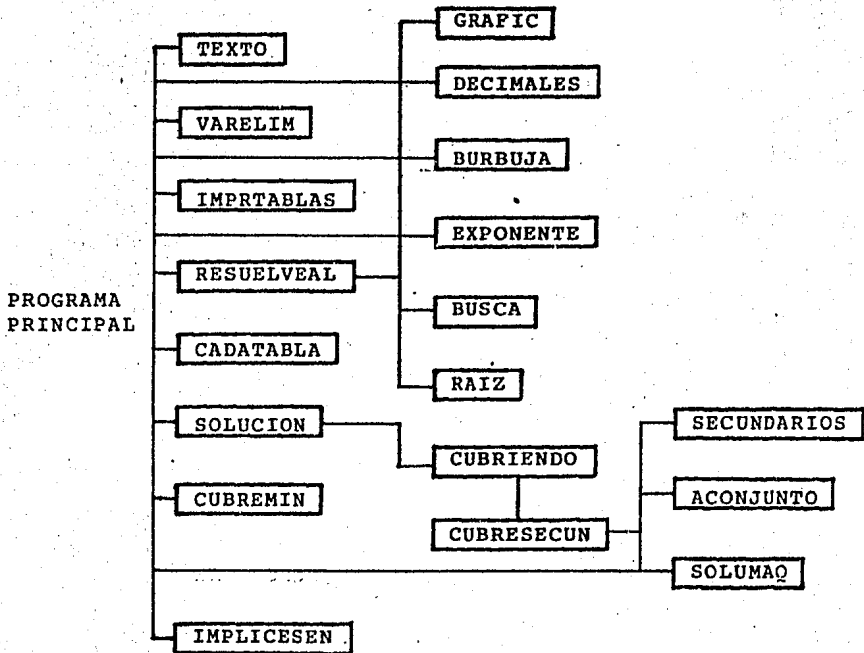


FIG.VI.1DIAGRAMAJERARQUICO

- CUBRESECUN
- CUBRIENDO
- SOLUCION
- CUBREMIN
- IMPLICESEN

A continuacion se presentan las subrutinas que sufrieron modificaciones

#### - ACONJUNTO

En el programa de Quine Mc.Cluskey para una salida, esta subrutina obtiene todas las combinaciones de los implicantes secundarios que cubren los cubos-0 que no se habian cubierto hasta ese momento y marca aquellos que dan las soluciones optimas. En el de Multiples Salidas realiza la misma tarea, con la diferencia de que solo usara una de las combinaciones que dan una solucion optima y no todas aquellas que puedan generarla.

#### - CADATABLA

Variables de entrada

CONT: entero escalar <numero de renglon de la matriz que contiene la funcion que se analizara en ese momento>

FUNC1: entero vectorial <cubos-0 de la funcion o interseccion de ellas que se analizara en ese momento>

NFUNC1: entero escalar <numero de cubos-0 de FUNC1>

Variable de salida

CONT1: entero escalar <renglon de la matriz de implicantes acumulados en que se guardo alguno de ellos>

Informa de la funcion o interseccion de funciones que va a presentar al usuario para su analisis. Llama a la subrutina SOLUCION para minimizarla; llama a la subrutina RESUELVEAL que se encarga de dirigir la solucion del usuario, le presenta los implicantes que se generaron de ese analisis y los que se van acumulando.

El programa principal se auxilia de DECIMALES para leer las funciones y validarlas. Verifica si existen cubos-0 comunes en ellas, y si es asi, genera funciones de las intersecciones.

Se auxilia tambien de CADATABLA para dirigir la solucion de cada una de las funciones y de sus intersecciones.

Dirise la impresion de tablas y la validacion de los datos para completarlas.

Muestra la expresion que minimiza el conjunto de funciones orisinales.

```

*
#FILE (VRB3)QUINE/MULTIPLE ON UNAM2
100 PROGRAM QUINEMULTIPLE(INPUT,OUTPUT);
200 (* ESTE PROGRAMA RESUELVE FUNCIONES A TRAVES DE QUINE MC.
300 Y GUIA A EL USUARIO A RESOLVERLO, VALIDANDO TODAS SUS SUGERENCIAS.
400 EL PROGRAMA USA PARA SU SOLUCION, PROCEDURES CUYA FUNCION SE EXPLICA
500 A CONTINUACION:
600
700 - TEXTO: CONTIENE UNA PEQUE&A INTRODUCCION AL METODO.
800 - VARELIM: SE MARCAN CON X TODAS LAS VARIABLES ELIMINADAS, DE
900 LOS CUBOS QUE INTEGRAN LA(S) SOLUCION(ES) OPTIMA(S).
1000 - RAIZ: OBTIENE LA RAIZ N(ESIMA) DE UN NUMERO.
1100 - ACONJUNTO: SI HAY HAY IMPLICANTES PRIMOS SECUNDARIOS, OBTIENE
1200 LOS QUE DAN LA SOLUCION OPTIMA Y LOS ANEXA A LOS
1300 ESENCIALES.
1400 - EXPONENTE: OBTIENE LA POTENCIA DE UN NUMERO EN BASE 2.
1500 - GRAFIC: GRAFICA LOS CUBOS DE LOS GRUPOS QUE ANALIZA EL USUARIO
1600 - RESUELVEAL: PRESENTA GRAFICAS Y VALIDA LAS RESPUESTAS DEL
1700 USUARIO EN FORMACION DE CUBOS(N+1) PARA CADA FUN
1800 CION Y SUS COMBINACIONES.
1900 - CUBREMIN: VALIDA LA RESPUESTA DEL USUARIO PARA LLENAR EL
2000 ULTIMO RENGLON DE LA TABLA DE IMPLICANTES PRIMOS.
2100 - CUBRESECUN: REvisa SI EXISTEN IMPLICANTES ESCENCIALES Y OBTIENE
2200 LA SOLUCION FINAL.
2300 - IMPRTABLAS: GRAFICA LAS TABLAS DE IMPLICANTES PRIMOS.
2400 - CADATABLA: INFORMA DE LA FUNCION O COMBINACION DE ELLAS QUE
2500 SE ESTA ANALIZANDO, INFORMA DE LOS IMPLICANTES
2600 GENERADOS POR CADA UNA DE ELLAS Y DE LOS QUE SE
2700 VAN ACUMULANDO PARA OBTENER LA SOLUCION.
2800 - SECUNDARIOS : DEFINE LA TABLA DE IMPLICANTES PRIMOS SECUNDARIOS
2900 - SOLUMAQ : RESUELVE A TRAVES DE Q.H. A PARTIR DE TOMAR LOS CUBOS
3000 CUYA DIFERENCIA SEA POTENCIA DE 2 Y OBTIENE LA
3100 O SOLUCIONES POSIBLES
3200 - BURBUJA : ORDENA LOS MINTERMINOS POR EL METODO DE BURBUJA
3300 - SOLUCION : PROCEDURE FUNDAMENTAL DEL PROGRAMA YA QUE ES DONDE SE
3400 RESUELVE EL PROGRAMA LAS FUNCIONES BAJO LA SIGUIENTE
3500 SECUENCIA:
3600 SEPARA LOS MINTERMINOS POR EL NUMERO DE UNOS DE SU REPRESENTACION
3700 BINARIA ALMACENANDOLOS EN SOLUCI.ARREGLO Y EL NUMERO DE ELLOS EN
3800 SOLUCI.NELEM
3900 HACE LA COMBINACION DE LOS MINTERMINOS CUYA DIFERENCIA SEA POTEN
4000 CIA DE 2, MARCANDO SIMULTANEAMENTE LOS QUE SE COMBINAN Y LOS QUE
4100 NO, EN EL PRIMER CASO LOS MARCA CON UNA "C", EN CASO CONTRARIO,
4200 CON UN "*".
4300 SOLUCI.NELEM ESTA ORGANIZADO DE TAL MANERA QUE LOS RENGLONES IN-
4400 DICAN EL NUMERO DE UNOS QUE TIENEN LOS CUBOS DE LOS RENGLONES
4500 1-10 Y A PARTIR DEL 11 EN ADELANTE, LAS DECENAS INDICAN EL NUMERO
4600 DE ELEMENTOS DEL CUBO Y LAS UNIDADES EL RENGLON DEL CUBO QUE SE
4700 COMBINO CON EL RENGLON SIGUIENTE. EJEMPLO :
4800 EL RENGLON 13 INDICA:
4900 2**1 : QUIERE DECIR QUE TIENE DOS ELEMENTOS POR CUBO
5000 3 : QUIERE DECIR QUE SE COMBINARON UN CUBO DEL RENGLON 3 CON
5100 - DECIMALES : VERIFICA LA ENTRADA DE LOS MINTERMINOS DE LA
5200 FUNCION EN FORMA DECIMAL
5300 - BUSCA : USADO EN LA SOLUCION DEL USUARIO PARA BUSCAR LOS CUBOS
5400 PROPUESTOS POR EL EN EL ARREGLO SOLUCI.ARREGLO
5500 ESTAS SON TODOS LOS PROCEDURES USADOS EN EL PROGRAMA
5600 EL PROGRAMA PRINCIPAL COMANDA TODA LA SOLUCION DEL ALUMNO
5700 *)
5800 TYPE CHARACTER = ARRAY(0..32) OF CHAR;
5900 CARAC = ARRAY(0..200) OF CHAR;
5901 (* *)
5902 (* *)

```

```

5903 (* *)
5904 (* *)
6000 MATRICES = ARRAY[0..70,0..65] OF INTEGER;
6100 MATRICESC = ARRAY[0..70,0..64] OF CHAR;
6200 VECTOR = ARRAY[-1..100] OF INTEGER;
6300 MATENT = ARRAY[0..60,0..60] OF INTEGER;
6400 MATCARAC = ARRAY[0..70,0..79] OF CHAR;
6500 NUMEROS = SET OF INTEGER;
6600 STRING = ARRAY [0..6] OF CHAR;
6700 LETRAS = RECORD
6800     LETRA : STRING;
6900     NOASOC : VECTOR;
7000     END;
7100 SOLUC = RECORD
7200     ARREGLO : MATRICES;
7300     ARRMARCA : MATRICESC;
7400     NOCUBIERTO : MATCARAC;
7500     MATSOL,IMPSEC,TABIMP : MATENT;
7600     SOLTOTAL,CUBIERTO : MATCARAC;
7700     MARC : CHARACTER;
7800     NELEM,NMATSOL,NSOLTOTAL,NTABSEC,NTABIMP : VECTOR;
7900     END;
8000 VAR
8100 FUNCION, (* CONTIENE LOS MINTERMINOS DE CADA FUNCION *)
8200 FUNCINTER, (* CONTIENE LOS MINTERMINOS DE LAS INTERSECCION DE
8300     FUNCIONES *)
8400 IMPLICANTES (* IMPLICANTES ACUMULADOS DEL ANALISIS DE LAS
8500     FUNCIONES *)
8600     : MATENT;
8700 CONT,CONT1,CONT2,CONT3,CONT4,CONT5,CONT6,CONT7,CONT8,CONT9,CONT10,
8800 NSALIDAS, (* NUMERO DE SALIDAS *)
8900 CONTAR, (* CONTADORES *)
9000 IMPLICA, (* INDICA SI HAY INTERSECCIONES O NO DE FUNCIONES *)
9100 NVARIA (* NUMERO DE VARIABLES DE LA FUNCION *),
9200 NDIGITOS (* TOTAL DE MINTERMINOS *),
9300 NIMPSEC (* TOTAL DE IMPLICANTES SECUNDARIOS *),
9400 NIMPLIC (* TOTAL DE IMPLICANTES PRIMOS *),
9500 NDIGSEC (* *),
9600 NCASILLAS (* NUMERO DE MINTERMINOS PARA SOLUCIONES PARCIALES *),
9700 TOTSOL (* TOTAL DE SOLUCIONES FINALES *),
9800 POTENCIA (* NUMERO MAXIMO DE MINTERMINOS PARA NVARIA *),
9900 T1 (* NUMERO DE 1S DE LOS MINTERMINOS *),
10000 T (* DIRECCIONAMIENTO DE RENGLONES *),
10100 MAX (* *),
10200 NSECUN,
10300 DECIMAL (* MINTERMINO EN FORMA DECIMAL *),
10400 DIFERENCIA,DIFERENCIA1 (* RESULTADOS DE DIFERENCIA ENTRE MINTERMI
10500     NOS PARA REVISAR SI ES POTENCIA DE 2 *),
10600 TSOL
10700     : INTEGER;
10800 ARRVAR1,ARRVAR2 (* POSICION DE LAS VARIABLES POR PESO *),
10900 CAMBIO,METODO (* LEE DECISIONES DEL USUARIO DE SI O NO *)
11000     : STRING;
11100 SOLUCIONES : MATCARAC (* TODAS LAS SOLUCIONES DE LA
11200     MAQUINA *)
11300 TABALUM : MATENT;
11400 SIMBOLD : LETRAS;
11500 SOLUCI : SOLUC;
11600 MARCAMIN, (* MARCA IMPLICANTES SECUNDARIOS QUE DAN SOL. OPTIMA*)
11700 IMPS, (* IMPLICANTES PRIMOS QUE PROPONE EL USUARIO *)
11800 CUBALUM, (* ULTIMO RENGLON DE LA TABLA DE IMPLICANTES PRIMOS
11900     QUE LLENA EL USUARIO Y SE VALIDA *)
12000 MARCAS, (* MARCA DE IMPLICANTES PRIMOS PROPUESTOS POR EL
12100     USUARIO *)
12101 (* *)

```

```

12102 (* *)
12103 (* *)
12104 (* *)
12200 CUBO, (* NOMBRE QUE ASIGNA EL PROGRAMA A LOS CUBOS DE LA
12300 TABLA DE IMPLICANTES PRIMOS ORIGINAL *)
12400 CUBOS, (* NOMBRE QUE ASIGNA EL PROGRAMA A LOS CUBOS DE LA
12500 TABLA DE IMPLICANTES SECUNDARIOS *)
12600 CASDIG (* SE MARCAN LOS CUBOS QUE CUBRE EL USUARIO *)
12700 : CHARACTER;
12800 NSOLUCIONES (* NUMERO TOTAL DE SOLUCIONES *);
12900 NCONJEXTEN (* NUMERO DE POSIBLES SOLUCIONES PARA IMPL. SEC *);
13000 NCONJUNTO ,
13100 PBURB (* CUBOS ENVIADOS A ORDENARSE *),
13200 DIGITOS (* MINTERMINOS *),
13300 DIGSEC (* MINTERMINOS DE LOS IMPLICANTES SECUNDARIOS *),
13400 NTABALUM (* NUMERO DE REDUCCIONES DEL ALUMNO *),
13500 CASILLAS,CASI (* CUBOS MANEJADOS POR EL ALUMNO *),
13600 POSI (* POSICION EN NDIGITOS DE NDIGSEC *),
13700 NIMP, (* NUMERO DE IMPLICANTES ACUMULADOS *)
13800 NFUNC, (* NUMERO DE FUNCIONES DADAS POR EL USUARIO *)
13900 NFUNCINTER,
14000 NOMFUNS, (* FUNCIONES QUE CONTIENE UN IMPLICANTE *)
14100 NOMFUNSS
14200 : VECTOR;
14300 BINARYO : ARRAY [0..30,0..30] OF CHAR;
14400 UNO (* MINTERMINOS CON 1 UNO EN SU REPRESENTACION BINARIA *),
14500 CONJ1,CONJ2 (* CONJUNTOS AUXILIARES PARA VERIFICACION DE INTERSECCIONES O IGUALDAD DE CUBOS *)
14600 : NUMEROS;
14700 CONJUNTO (* ARREGLO DE CONJUNTOS DE CUBOS *)
14900 : ARRAY [0..50,0..150] OF NUMEROS;
15000 CDIGITOS : ARRAY [0..70,0..1,0..72] OF CHAR;
15100 SOLALUM : CHAR;
15200 OPCION,AUX1 : CHAR;
15300 BANDERA2,BANDITOTAL,BANDERA,BANDERA1 : BOOLEAN;
15400
15500 PROCEDURE TEXTO;
15600 BEGIN
15700 WRITELN('LOS MINTERMINOS DE UNA FUNCION SON LLAMADOS CUBOS-0.');
```

15800 WRITELN('EL METODO CONSISTE EN UNA BUSQUEDA EXHAUSTIVA DE TODAS '
15900 ' LAS COMBINACIONES');

16000 WRITELN('POSIBLES DE LOS CUBOS-N PARA FORMAR LOS CUBOS-(N+1) Y '
16100 ' LUEGO SELECCIONAR');

16200 WRITELN('LA COMBINACION MINIMA DE CUBOS PARA REALIZAR LA(S) FUNCION '
16300 '(ES).');

16400 WRITELN('SE ANALIZA LA INTERSECCION DE LAS FUNCIONES Y CADA UNA DE '
16500 ' ELLAS INDEPEN');

16600 WRITELN('DIENTEMENTE.');

16700 WRITELN('SE INICIA OBTENIENDO LA REPRESENTACION BINARIA DE LOS '
16800 ' CUBOS-0 Y SE');

16900 WRITELN('AGRUPAN EN FUNCION DEL NUMERO DE '1 S.');

17000 WRITELN('LOS CUBOS-1 SE OBTIENEN COMPARANDO GRUPOS CONSECUTIVOS '
17100 ' POR PARES.EJEMPLO:');

17200	WRITELN('	GRUPO 1	GRUPO 2');
17300	WRITELN('	001 1	011 3');
17400	WRITELN('	010 2	110 6');

17500 WRITELN('SI COMPARAMOS UN CUBO DEL GRUPO 2 CON OTRO DEL GRUPO '
17600 ' 1 Y LA DIFERENCIA');

17700 WRITELN('ES POTENCIA DE 2. HECHOS OBTENIDO UN CUBO-1.');

17800 WRITELN(' CUBO-0(GRUPO 2) - CUBO-0(GRUPO 1) = 3 - 2 = 1');

17900 WRITELN('OBSERVA QUE SE HA HECHO LA DIFERENCIA DEL GRUPO MAYOR '
18000 ' - GRUPO MENOR, LO');

18100 WRITELN('CONTRARIO NO ES CIERTO.');

18200 WRITELN('ESTE PROCEDIMIENTO SE CONTINUA HASTA CUBRIR TODOS LOS '
18300 ' CUBOS DE TODOS LOS');

18301 (\* \*)



```

18302 (* *)
18303 (* *)
18304 (* *)
18400 WRITELN('GRUPOS.');
```

18500 WRITELN('PARA OBTENER LOS CUBOS-2 HASTA CUBOS-N DEBES OBSERVAR ',  
18600 'LA MISMA CONDICION');

18700 WRITELN('DE DIFERENCIA DE POTENCIA DE 2 ENTRE LOS CUBOS DE CADA ',  
18800 'GRUPO CONSECUTIVO');

18900 WRITELN('EN SUS ELEMENTOS CORRESPONDIENTES Y ADEMAS QUE ',  
19000 'LA(S) DIFERENCIA(S)');

19100 WRITELN('ANTERIOR(ES), LA CUAL TE PRESENTARE EN ESTE PROGRAMA ',  
19200 'ENTRE PARENTESIS');

19300 WRITELN('SEA LA MISMA.');

19400 WRITELN('CONTINUA...'); READLN;

19500 WRITELN('SI ALGUN CUBO DE ALGUN GRUPO NO SE COMBINA, LO MARCAR ',  
19600 'AS, ESTE FORMARA');

19700 WRITELN('PARTE DE LOS IMPLICANTES PRIMOS.');

19800 WRITELN('LOS IMPLICANTES PRIMOS SON TODOS AQUELLOS CUBOS QUE ',  
19900 'DURANTE EL PROCESO');

20000 WRITELN('DE COMPARACION NO LOGRARON COMBINARSE, DE ENTRE ELLOS ',  
20100 'SE SELECCIONARAN');

20200 WRITELN('AQUELLOS QUE NOS DARAN LA SOLUCION MINIMA PARA CADA UNA DE ',  
20300 'LAS FUNCIONES.');

20400 WRITELN('ESTOS IMPLICANTES SE ORDENAN EN UNA TABLA EN FUNCION ',  
20500 'DE LA CANTIDAD DE');

20600 WRITELN('CUBOS-O QUE CONTENGAN.');

20700 WRITELN('ESTA TABLA, TENDRA EN SUS RENGLONES LOS IMPLICANTES QR ',  
20800 'DENADOS Y EN SUS');

20900 WRITELN('COLUMNAS TODOS LOS CUBOS-O DE TODAS LAS FUNCIONES.');

21000 WRITELN('DE CADA IMPLICANTE SE BUSCARAN EN LAS COLUMNAS, LOS ',  
21100 'CUBOS-O QUE LO FORMAN');

21200 WRITELN('Y SE MARCARA LA COLUMNA CORRESPONDIENTE.');

21300 WRITELN('QUE LE PERTENEZCAN.');

21400 WRITELN('EL PASO SIGUIENTE ES VERIFICAR CADA COLUMNA QUE CONTENGA ',  
21500 'SOLO UNA MARCA.');

21600 WRITELN('CUANDO SE ENCUENTREN, SE MARCARAN EN UN RENGLON ADICIONAL ',  
21700 'AL IGUAL QUE');

21800 WRITELN('TODAS AQUELLAS QUE ESTEN MARCADAS EN EL RENGLON QUE SE EN ',  
21900 'CONTRO SOLO UNA');

22000 WRITELN('MARCA, ESTE IMPLICANTE SE CONVIERTE EN ESCENCIAL Y FORMA ',  
22100 'RA PARTE DE LA');

22200 WRITELN('SOLUCION.');

22300 WRITELN('SI AL TERMINAR DE REVISAR TODAS LAS COLUMNAS Y MAR ',  
22400 'CADO LOS IMPLICANTES');

22500 WRITELN('ESCENCIALES HAY COLUMNAS QUE NO SE MARCARON, PASAREMOS ',  
22600 'ESTAS COLUMNAS A');

22700 WRITELN('OTRA TABLA ASI COMO LOS IMPLICANTES QUE LAS CONTENGAN.');

22800 WRITELN('CONTINUA...'); READLN;

22900 WRITELN('EXISTEN DIVERSOS METODOS PARA RESOLVER ESTA ULTIMA TABLA ',  
23000 ', EN ESTE PROGRA');

23100 WRITELN('MA LO HAREMOS A TRAVES DE UN METODO LLAMADO DE PETRICK, EL ',  
23200 'CUAL CONSISTE');

23300 WRITELN('EN TOMAR LOS IMPLICANTES QUE FORMAN ESTA TABLA E INTER ',  
23400 'PRETARLAS COMO');

23500 WRITELN('VARIABLES BOOLEANAS, ESTO ES TOMARAN VALORES CIERTO O ',  
23600 'FALSO, DEPENDIENDO');

23700 WRITELN('DE SI SE ESCOGE O NO PARA EL ANALISIS.');

23800 WRITELN('A PARTIR DE LA PRIMERA COLUMNA SE DEBEN SELECCIONAR ',  
23900 'CUALQUIERA DE LOS');

24000 WRITELN('IMPLICANTES QUE TENGAN MARCA EN ELLA PARA CUBRIRLO ',  
24100 'DE MANERA QUE SI');

24200 WRITELN('REPRESENTAMOS CADA IMPLICANTE POR UNA LETRA Y a ES EL ',  
24300 'PRIMER IMPLICANTE');

24400 WRITELN('Y d EL TERCERO DE LA TABLA SECUNDARIA, Y AMBOS TIENEN ',  
24500 'MARCADOS EN ALGUNA');

24501 (\* \*)

```

24503 (* *)
24504 (* *)
24600 WRITELN('DE LAS COLUMNAS, SE EXPRESARAN POR:');
24700 WRITELN(' (a+d)=1');
24800 WRITELN('YA QUE CUALQUIERA DE LOS DOS CUBRE ESE CUBO. DEBIDO A ',
24900 ' TODOS ELLOS DEBEN');
25000 WRITELN('CUBRIRSE, SE UNIRAN POR EL CONECTOR 'Y', ESTO ES, ',
25100 ' SI OTRA COLUMNA CONTIENE');
25200 WRITELN('POR EJEMPLO LOS CUBOS c y f MARCADOS, SE TENDRA LA ',
25300 ' EXPRESION:');
25400 WRITELN(' (a+b)(c+f)=1');
25500 WRITELN('ESTE PROCESO SE SIGUE HASTA INVOLUCRAR TODAS LAS COLUM',
25600 ' NAS. ');
25700 WRITELN('ENSEGUIDA SE REALIZARAN LOS PRODUCTOS INDICADOS Y SE ',
25800 ' SIMPLIFICARA LA');
25900 WRITELN('EXPRESION RESULTANTE POR MEDIO DE LOS TEOREMAS FUNDA',
26000 ' MENTALES DEL ALGEBRA');
26100 WRITELN('DE BOOLE, POR LO MENOS ALGUNO DE LOS PRODUCTOS DEBE ',
26200 ' SER CIERTO (1). DE');
26300 WRITELN('ESTOS SE SELECCIONA ALGUNO CUYO COSTO SEA MENOR, ESTO ES, ',
26400 ' EL NUMERO DE');
26500 WRITELN('ENTRADAS ES MAS REDUCIDO. ');
26600 WRITELN('LOS CUBOS QUE CONTENGA ESTE PRODUCTO MAS LOS IMPLICANTES ',
26700 ' DE LA PRIMERA');
26800 WRITELN('TABLA, DAN UNA SOLUCION MINIMA. ');
26900 WRITELN('CONTINUA...'); READLN;
27000 WRITELN('ANALIZANDO CADA IMPLICANTE ESCENCIAL Y SECUNDARIO EN ',
27100 ' FORMA INDEPENDIENTE');
27200 WRITELN('COMPARAMOS LAS POSICIONES CORRESPONDIENTES DE TODAS ',
27300 ' LAS REPRESENTACIONES');
27400 WRITELN('BINARIAS DE LOS CUBOS-0 QUE LO COMPONEN. ');
27500 WRITELN('LOS VALORES DE ESTAS POSICIONES QUE SE MANTENGAN CONS',
27600 ' TANTES SE ASOCIAN A ');
27700 WRITELN('LA VARIABLE QUE LE CORRESPONDA POR SU POSICION Y ES, ',
27800 ' TO NOS DA LA SOLUCION');
27900 WRITELN('DE LA FUNCION CON LAS VARIABLES DE QUE ESTA DEFINIDA. ');
28000 WRITELN('EJEMPLO:');
28100 WRITELN('SUPONIENDO QUE LA SOLUCION ESTA EXPRESADA POR LOS IM',
28200 ' PLICANTES:');
28300 WRITELN(' F(A,B,C) = a+b');
28400 WRITELN('SIENDO a = 0,1,2,3');
28500 WRITELN(' b = 0,1,4,5');
28600 WRITELN('ANALIZANDO a; ANALIZANDO b:');
28700 WRITELN(' CBA CBA');
28800 WRITELN(' 0 = 000 0 = 000');
28900 WRITELN(' 1 = 001 1 = 001');
29000 WRITELN(' 2 = 010 4 = 100');
29100 WRITELN(' 3 = 011 5 = 101');
29200 WRITELN('OBSERVAMOS QUE C SE MANTIENE CONSTANTE EN LOS 4 CUBOS-0, ',
29300 ' DE a y B en b');
29400 WRITELN('Y EL VALOR ASOCIADO ES 0 (EN ESTE CASO) PARA AMBOS');
29500 WRITELN(' => a=C0 Y b=B0');
29600 WRITELN('Y LA SOLUCION DE LA FUNCION EXPRESADA CON SUS VARIABLES ES:');
29700 WRITELN(' f(A,B,C) = a+b = C0 + B0');
29800 WRITELN('CONTINUA...'); READLN;
29900 WRITE('AHORA BIEN ');
30000 END; (* TEXTO *)
30100
30200 PROCEDURE VARELIN(CUBO:CHAR; VAR CONT1:INTEGER;RENGLON:INTEGER);
30300 VAR CONT2,CONT4: INTEGER;
30400 BEGIN
30500 SOLUCIONES[CONT10,0]:=CUBO; SOLUCIONES[CONT10,1]:=' ';
30600 SOLUCIONES[CONT10,2]:='='; SOLUCIONES[CONT10,3]:=' ';
30700 CONT4:=4;
30701 (* *)
30702 (* *)

```

```

30703 (* *)
30704 (* *)
30800 FOR CONT2:=NVARIA DOWNT0 1 DO
30900 BEGIN
31000 IF SOLUCI.SOLTOTALCRENGLON,CONT1] = SIMBOLO.LETRACCONT2] THEN
31100 BEGIN
31200 SOLUCIONES[CONT10,CONT4]:=SOLUCI.SOLTOTALCRENGLON,CONT1+1];
31300 IF SOLUCI.SOLTOTALCRENGLON,CONT1+2] = '+' THEN
31400 CONT1:=CONT1+3
31500 ELSE
31600 CONT1:=CONT1+2;
31700 END
31800 ELSE
31900 SOLUCIONES[CONT10,CONT4]:= 'X';
32000 CONT4:=CONT4+1;
32100 END;
32200 CONT10:=CONT10+1;
32300 END; (* VARELIM *)
32400
32500 FUNCTION EXPONENTE(N:INTEGER) : INTEGER;
32600 BEGIN
32700 EXPONENTE:=ROUND(EXP(N* LN(2)));
32800 END;
32900
33000 FUNCTION RAIZ(NUMERO:INTEGER) : INTEGER;
33100 VAR I,COCIENTE : INTEGER;
33200 BEGIN
33300 COCIENTE:=0;
33400 I:=NUMERO;
33500 WHILE I > 1 DO
33600 BEGIN
33700 I:=I DIV 2;
33800 COCIENTE:=COCIENTE+1;
33900 END;
34000 RAIZ:=COCIENTE;
34100 END; (* RAIZ *)
34200
34300 PROCEDURE BUSCA(VAR T,UBICA:INTEGER; INICIO,FIN:INTEGER);
34400 VAR
34500 CONT,CONT1: INTEGER;
34600 BEGIN
34700 WHILE (SOLUCI.NELEMCT] = 0) AND (T > (T1*10)) DO
34800 T:=T-1;
34900 BANDERA:=FALSE;
35000 IF (T > T1*10) OR (T=0) THEN
35100 WHILE (NOT BANDERA) AND ((T>T1*10) OR (T=0)) DO
35200 BEGIN
35300 FOR CONT:=0 TO SOLUCI.NELEMCT]-1 DO
35400 BEGIN
35500 IF SOLUCI.ARREGLOCT,CONT] = CASILLAS[INICIO] THEN
35600 BEGIN
35700 BANDERA:=TRUE;
35800 FOR CONT1:=INICIO TO .FIN DO
35900 BEGIN
36000 IF SOLUCI.ARREGLOCT,CONT]=CASILLAS[CONT1] THEN
36100 BANDERA:=BANDERA AND TRUE
36200 ELSE
36300 BANDERA:=FALSE;
36400 CONT:=CONT+1;
36500 END;
36600 CONT:=CONT-1;
36700 END
36800 ELSE
36900 BEGIN
36901 (* *)
36902 (* *)

```

```

36903 (* *)
36904 (* *)
37000          BANDERA:=FALSE;
37100          CONT:=CONT+(FIN-INICIO);
37200          END;
37300          IF BANDERA THEN
37400          BEGIN
37500              UBICA:=CONT-(FIN-INICIO);
37600              CONT:=SOLUCI.NELEMETJ;
37700          END;
37800          END;
37900          T:=T-1;
38000          END;
38100 END; (* BUSCA *)
38200
38300 PROCEDURE GRAFIC(CONT,NGRUP,CONT6:INTEGER);
38400 VAR
38500     CONT1,CONT3,CONT4,CONT6,CONT7 :INTEGER;
38600 BEGIN
38700     CONT7:=CONT6;
38800     IF CONT<10 THEN
38900         T1:=1
39000     ELSE
39100         T1:=EXPONENTE(CONT DIV 10);
39200     CASE CONT OF
39300         0,1,2,3,4: BEGIN
39400             FOR CONT1:=1 TO NGRUP DO
39500                 BEGIN
39600                     WRITE(' ':4,'GRUPO ',CONT7:2,' ');
39700                     CONT7:=CONT7+1;
39800                 END;
39900                 CONT6:=--1;
40000             END;
40100         11,12,13,14: BEGIN
40200             FOR CONT1:=1 TO NGRUP DO
40300                 BEGIN
40400                     WRITE(' ':7,'GRUPO ',CONT7:2,' ':5);
40500                     CONT7:=CONT7+1;
40600                 END;
40700                 CONT6:=0;
40800             END;
40900         21,22,23,24: BEGIN
41000             FOR CONT1:=1 TO NGRUP DO
41100                 BEGIN
41200                     WRITE(' ':12,'GRUPO ',CONT7:2,' ':5);
41300                     CONT7:=CONT7+1;
41400                 END;
41500                 CONT6:=1;
41600             END;
41700         31,32,33,34: BEGIN
41800             FOR CONT1:=1 TO NGRUP DO
41900                 BEGIN
42000                     WRITE(' ':18,'GRUPO ',CONT7:2,' ':5);
42100                     CONT7:=CONT7+1;
42200                 END;
42300                 CONT6:=2;
42400             END;
42500         41,42,43,44: BEGIN
42600             FOR CONT1:=1 TO NGRUP DO
42700                 BEGIN
42800                     WRITE(' ':24,'GRUPO ',CONT7:2);
42900                     CONT7:=CONT7+1;
43000                 END;
43100                 CONT6:=3;
43101 (* *)
43102 (* *)

```

```

43103 (* *)
43104 (* *)
43200
43300
43400 51,52,53,54: BEGIN
43500     FOR CONT1:=1 TO NGRUP DO
43600         BEGIN
43700             WRITE(' '124,'GRUPD ',CONT7:2);
43800             CONT7:=CONT7+1;
43900         END;
44000     CONT4:=4;
44100     END; (*CASE *)
44200     WRITELN;
44300     IF SOLUCI.NELEM[CONTJ] > SOLUCI.NELEM[CONTJ+1] THEN
44400     BEGIN
44500         FOR CONT4:=0 TO SOLUCI.NELEM[CONTJ+1]-1 DO
44600         BEGIN
44700             WRITE(' '6);
44800             FOR CONT1:=CONT4 TO (CONT4+T1-2) DO
44900                 WRITE(SOLUCI.ARREGLOC[CONT,CONT1]:2,',');
45000             WRITE(SOLUCI.ARREGLOC[CONT,CONT4+T1-1]:2);
45100             IF CONT > 10 THEN
45200             BEGIN
45300                 WRITE(' (');
45400                 FOR CONT1:=0 TO (CONT6-1) DO
45500                     WRITE(CDIGITOS[CONT,1,CONT4+CONT1]:1,',');
45600                 WRITE(CDIGITOS[CONT,1,CONT6+CONT4]:1,')');
45700             END;
45800             WRITE(CDIGITOS[CONT,0,CONT4]:1,' '10);
45900             FOR CONT1:=CONT4 TO (CONT4+T1-2) DO
46000                 WRITE(SOLUCI.ARREGLOC[CONT+1,CONT1]:2,',');
46100             WRITE(SOLUCI.ARREGLOC[CONT+1,CONT4+T1-1]:2);
46200             IF CONT > 10 THEN
46300             BEGIN
46400                 WRITE(' (');
46500                 FOR CONT1:=0 TO (CONT6-1) DO
46600                     WRITE(CDIGITOS[CONT+1,1,CONT4+CONT1]:1,',');
46700                 WRITE(CDIGITOS[CONT+1,1,CONT6+CONT4]:1,')');
46800             END;
46900             WRITELN(CDIGITOS[CONT+1,0,CONT4]:1);
47000             CONT4:=CONT4+T1-1;
47100         END;
47200     IF SOLUCI.NELEM[CONTJ] <> SOLUCI.NELEM[CONTJ+1] THEN
47300     FOR CONT3:=CONT4 TO SOLUCI.NELEM[CONTJ]-1 DO
47400     BEGIN
47500         WRITE(' '6);
47600         FOR CONT1:=CONT3 TO (CONT3+T1-2) DO
47700             WRITE(SOLUCI.ARREGLOC[CONT,CONT1]:2,',');
47800         WRITE(SOLUCI.ARREGLOC[CONT,CONT3+T1-1]:2);
47900         IF CONT > 10 THEN
48000         BEGIN
48100             WRITE(' (');
48200             FOR CONT1:=0 TO (CONT6-1) DO
48300                 WRITE(CDIGITOS[CONT,1,CONT3+CONT1]:1,',');
48400             WRITE(CDIGITOS[CONT,1,CONT6+CONT3]:1,')');
48500         END;
48600         WRITELN(CDIGITOS[CONT,0,CONT3]:1);
48700         CONT3:=CONT3+T1-1;
48800     END;
48900     END
49000     ELSE
49100     BEGIN
49200     FOR CONT4:=0 TO SOLUCI.NELEM[CONTJ]-1 DO
49300     BEGIN
49301 (* *)
49302 (* *)

```

```

49303 (* *)
49304 (* *)
49400 WRITE(' ':6);
49500 FOR CONT1:=CONT4 TO (CONT4+T1-2) DO
49600   WRITE(SOLUCI.ARREGLOCCONT,CONT1:2,',');
49700 WRITE(SOLUCI.ARREGLOCCONT,CONT4+T1-1:2);
49800 IF CONT > 10 THEN
49900   BEGIN
50000     WRITE(' (');
50100     FOR CONT1:=0 TO (CONT6-1) DO
50200       WRITE(CDIGITOSECONT,1,CONT4+CONT1:1,',');
50300       WRITE(CDIGITOSECONT,1,CONT6+CONT4:1,',');
50400     END;
50500     WRITE(CDIGITOSECONT,0,CONT4:1,' ':10);
50600     FOR CONT1:=CONT4 TO (CONT4+T1-2) DO
50700       WRITE(SOLUCI.ARREGLOCCONT+1,CONT1:2,',');
50800       WRITE(SOLUCI.ARREGLOCCONT+1,CONT4+T1-1:2);
50900     IF CONT > 10 THEN
51000       BEGIN
51100         WRITE(' (');
51200         FOR CONT1:=0 TO (CONT6-1) DO
51300           WRITE(CDIGITOSECONT+1,1,CONT4+CONT1:1,',');
51400           WRITE(CDIGITOSECONT+1,1,CONT6+CONT4:1,',');
51500         END;
51600         WRITELN(CDIGITOSECONT+1,0,CONT4:1);
51700         CONT4:=CONT4+T1-1;
51800       END;
51900     IF SOLUCI.NELEMCCONT <> SOLUCI.NELEMCCONT+1 THEN
52000     FOR CONT3:=CONT4 TO SOLUCI.NELEMCCONT+1:1-1 DO
52100       BEGIN
52200         WRITE(' ':19);
52300         FOR CONT1:=0 TO CONT6 DO
52400           WRITE(' ':7);
52500         FOR CONT1:=CONT3 TO (CONT3+T1-2) DO
52600           WRITE(SOLUCI.ARREGLOCCONT+1,CONT1:2,',');
52700           WRITE(SOLUCI.ARREGLOCCONT+1,CONT3+T1-1:2);
52800         IF CONT > 10 THEN
52900           BEGIN
53000             WRITE(' (');
53100             FOR CONT1:=0 TO (CONT6-1) DO
53200               WRITE(CDIGITOSECONT+1,1,CONT3+CONT1:1,',');
53300               WRITE(CDIGITOSECONT+1,1,CONT6+CONT3:1,',');
53400             END;
53500             WRITELN(CDIGITOSECONT+1,0,CONT3:1);
53600             CONT3:=CONT3+T1-1;
53700           END;
53800         END;
53900     END; (* GRAFIC *)
54000
54100 PROCEDURE BURBUJA(VAR DIGITOS:VECTOR;NDIGITOS:INTEGER);
54200   VAR
54300     CONT1,AUX,CONT2 : INTEGER;
54400   BEGIN
54500     FOR CONT2:=0 TO (NDIGITOS-1) DO
54600       FOR CONT1:=(CONT2+1) TO NDIGITOS DO
54700         IF DIGITOSECONT1 < DIGITOSECONT2 THEN
54800           BEGIN
54900             AUX:=DIGITOSECONT2;
55000             DIGITOSECONT2:=DIGITOSECONT1;
55100             DIGITOSECONT1:=AUX;
55200           END;
55300     END; (* BURBUJA *)
55400
55500 PROCEDURE DECIMALES(VAR DIGS:VECTOR;VAR NDIGS:INTEGER);
55501 (* *)
55502 (* *)

```

```

55503 (* *)
55504 (* *)
55600 LABEL 60;
55700 VAR
55800     AUX1,CNUMERO,CCERO,I,I1,T1,J,I2:INTEGER;
55900     MINTER:CHAR;
56000 BEGIN
56100     60:
56200     READLN;
56300     READ(MINTER);
56400     J:=0;
56500     CCERO:=ORD('0');
56600     WHILE MINTER <> ' ' DO
56700     BEGIN
56800         WHILE MINTER = ' ' DO
56900             READ (MINTER);
57000             I:=1;
57100             AUX1:=0;
57200             IF((MINTER <='9') AND (MINTER >='0')) THEN
57300             BEGIN
57400                 WHILE ((MINTER<='9') AND (MINTER >='0')) DO
57500                 BEGIN
57600                     CNUMERO:=ORD(MINTER);
57700                     AUX1:=AUX1*I+(CNUMERO - CCERO);
57800                     I:=10*I;
57900                     READ(MINTER);
58000                 END;
58100                 IF (AUX1>=POTENCIA) OR (AUX1<0) THEN
58200                 BEGIN
58300                     WRITE('UN NUMERO DE LOS PROPORCIONADOS, NO ESTA EN UN');
58400                     WRITELN(' RANGO PERMISIBLE, VERIFICA');
58500                     GOTO 60;
58600                 END
58700                 ELSE
58800                 IF J>0 THEN
58900                     FOR I2:=J-1 DOWNT0 0 DO
59000                         IF DIGSC[I2] = AUX1 THEN
59100                         BEGIN
59200                             WRITELN('HAY UN MINTERMINO REPTIDO,VERIFICA ');
59300                             GOTO 60;
59400                         END;
59500                         DIGSC[J]:=AUX1;
59600                         J:=J+1;
59700                         WHILE MINTER = ' ' DO
59800                             READ(MINTER);
59900                         IF MINTER = ' ' THEN
60000                             READ (MINTER);
60100                         END
60200                         ELSE
60300                         BEGIN
60400                             WRITELN('HAY UN CARACTER NO VALIDO, VERIFICA');
60500                             GOTO 60;
60600                         END;
60700                         END;
60800                         NDIGS:=J-1;
60900                     END; (* DECIMALES *)
61000
61100     PROCEDURE RESUELVEALUM;
61200     VAR
61300         CONT,CONT1 :INTEGER;
61400     BEGIN
61500         WRITELN;
61600         WRITELN('TUS MINTERMINOS (CUBOS=0) ORDENADOS EN FUNCION DEL ',
61700             'NUMERO DE UNOS CON SU');
61701 (* *)
41702 (* *)

```

```

61703 (* *)
61704 (* *)
61800 WRITELN('REPRESENTACION BINARIA Y DECIMAL SON:');
61900 CONT6:=NVARIA;
62000 WHILE SOLUCI.NELEMCONT6] = 0 DO
62100   CONT6:=CONT6-1;
62200   CONT1:=0;
62300   WHILE SOLUCI.NELEMCONT1] = 0 DO
62400     CONT1:=CONT1+1;
62500     CONT2:=0;
62600     FOR CONT:=CONT1 TO CONT6 DO
62700       IF SOLUCI.NELEMCONT] > CONT2 THEN
62800         CONT2:=SOLUCI.NELEMCONT];
62900         CONT4:=1;
63000         FOR CONT:=CONT1 TO CONT6 DO
63100           BEGIN
63200             WRITE(' GRUPO ',CONT:1);
63300             CONT4:=CONT4+1;
63400           END;
63500         WRITELN;
63600         FOR CONT:=0 TO CONT2 DO
63700           BEGIN
63800             WRITE(' ');
63900             FOR CONT4:=CONT1 TO CONT6 DO
64000               IF CONT < SOLUCI.NELEMCONT4] THEN
64100                 BEGIN
64200                   DECIMAL:=SOLUCI.ARREGLOC[CONT4,CONT];
64300                   FOR CONT3:=1 TO NVARIA DO
64400                     BEGIN
64500                       IF (DECIMAL MOD 2) = 1 THEN
64600                         BINARYOCO,CONT3]:='1'
64700                       ELSE
64800                         BINARYOCO,CONT3]:='0';
64900                       DECIMAL:=DECIMAL DIV 2;
65000                     END;
65100                       FOR CONT3:=NVARIA DOWNT0 1 DO
65200                         WRITE(BINARYOCO,CONT3]:1);
65300                       CONT3:=NVARIA;
65400                       WHILE CONT3 < 4 DO
65500                         BEGIN
65600                           WRITE(' ');
65700                           CONT3:=CONT3+1;
65800                         END;
65900                           WRITE(' ',SOLUCI.ARREGLOC[CONT4,CONT]:2,' ':5);
66000                         END
66100                       ELSE
66200                         WRITE(' ':12);
66300                       WRITELN;
66400                     END;
66500                   WRITELN('DE AQUI EN ADELANTE, USAREMOS LOS MINTERMINOS EN',
66600                     ' SU REPRESENTACION');
66700                   WRITELN('DECIMAL');
66800                   FOR CONT1:=0 TO 20 DO
66900                     FOR CONT3:=0 TO 72 DO
67000                       C[BITOS[CONT1,0,CONT3]:=' ':
67100                   CONT6:=0;
67200                   WHILE (CONT6 < (MAX*10 +10)) DO
67300                     BEGIN WRITELN;
67400                     WRITELN('ANALISIS DE LOS CUBOS ',CONT6 DIV 10:1,' PARA OBTENER ',
67500                       'LOS CUROS ',CONT6 DIV 10 +1:1);
67600                     IF (CONT6 DIV 10) > 0 THEN
67700                       BEGIN WRITELN;
67800                         WRITELN('PARA QUE SE PUEDAN COMBINAR CUBOS DE LOS SI',
67900                           ' GUIENTES GRUPOS, SE DEBE');
67901 (* *)
67902 (* *)

```



```

67903 (* *)
67904 (* *)
68000   WRITELN('CUMPLIR CON:');
68100   WRITELN('A) LA(S) DIFERENCIA(S) INDICADA(S) EN EL PARENTESIS',
68200     ' DEBE(N) SER LA(S)');
68300   WRITELN('MISMA(S), YA QUE CORRESPONDE A LA MISMA VARIABLE ',
68400     'ELIMINADA');
68500   WRITELN('B) LA(S) DIFERENCIA(S) ENTRE LOS ELEMENTOS CO',
68600     'RRESPONDIENTES ENTRE LOS ');
68700   WRITELN('CUBOS DE DIFERENTES GRUPOS DEBE SER IGUAL Y POTENCIA ',
68800     'DE 2');
68900   END;
69000   FOR CONT:=CONT6 TO (CONT6+NVARIA-1) DO
69100     BEGIN
69200       IF (SOLUCI.NELEM[CONT] > 0) AND (SOLUCI.NELEM[CONT+1] > 0) THEN
69300         BEGIN
69400           CONT7:=CONT MOD 10;
69500           GRAFIC(CONT,2,CONT7);
69600         IF CONT < 10 THEN
69700           BEGIN
69800             WRITELN('EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA',
69900               ' UNO, CUYA DIFERENCIA SEA UNA');
70000             WRITELN('POTENCIA DE 2?');
70100           END
70200         ELSE
70300           WRITELN('EXISTEN EN ESTOS GRUPOS CUBOS QUE SE COMBINEN?');
70400         READLN;
70500         READ(CAMBIOC1,CAMBIOC2);
70600         WHILE NOT (((CAMBIOC1 = 'S') AND (CAMBIOC2 = 'I')) OR
70700           ((CAMBIOC1 = 'N') AND (CAMBIOC2 = 'O'))) DO
70800           BEGIN
70900             WRITELN('REPITE SI HAY CUBOS POR COMBINAR O NO');
71000             READLN;
71100             READ(CAMBIOC1,CAMBIOC2);
71200           END;
71300           REPEAT
71400             IF CAMBIOC1 = 'S' THEN
71500               BEGIN
71600                 BANDERA1:=TRUE;
71700                 FOR CONT1:=0 TO SOLUCI.NELEM[CONT]-1 DO
71800                   IF SOLUCI.ARRMARCA[CONT,CONT1] <> '*' THEN
71900                     BANDERA1:=FALSE;
72000                   IF NOT BANDERA1 THEN
72100                     BEGIN
72200                       BANDERA1:=TRUE;
72300                       FOR CONT1:=0 TO SOLUCI.NELEM[CONT+1]-1 DO
72400                         IF SOLUCI.ARRMARCA[CONT+1,CONT1] <> '*' THEN
72500                           BANDERA1:=FALSE;
72600                       END;
72700                       IF BANDERA1 THEN
72800                         BEGIN
72900                           WRITELN('OBSERVA BIEN LOS GRUPOS! NO SE PUEDEN COMBINAR');
73000                           CAMBIOC1:='N';
73100                         END;
73200                       END
73300                     ELSE
73400                       BANDERA1:=FALSE;
73500                   IF CAMBIOC1 = 'S' THEN
73600                     BEGIN
73700                       WRITELN;
73800                       WRITELN('DAME UNA PAREJA DE ELLOS');
73900                       DECIMALES(CASILLAS,NCASILLAS);
74000                       FOR CONT2:=0 TO NCASILLAS DO
74100                         CASI[CONT2]:=CASILLAS[CONT2];
74101 (* *)
74102 (* *)

```



```

80303 (* *)
80304 (* *)
80400 FOR CONT1:=CONT4 TO (NCASILLAS DIV 2 + CONT4) DO
80500 CDIGITOSECONT+1,0,CONT1J:='C';
80600 FOR CONT1:=CONT2 TO (NCASILLAS DIV 2 + CONT2) DO
80700 CDIGITOSECONT,0,CONT1J:='C';
80800 WRITELN('BIEN, ESTOS CUBOS SI SE COMBINAN');
80900 GRAFIC(CONT,2,CONT7);
81000 END;
81100 BANDERA:=TRUE;
81200 WRITELN('AUN HAY CUBOS POR COMBINAR?');
81300 READLN;
81400 READ(CAMBIOE1J,CAMBIOE2J);
81500 WHILE NOT (((CAMBIOE1J = 'S') AND (CAMBIOE2J = 'I')) OR
81600 ((CAMBIOE1J = 'N') AND (CAMBIOE2J = 'O')) DO
81700 BEGIN
81800 WRITELN('REPITE SI HAY CUBOS POR COMBINAR O NO');
81900 READLN;
82000 READ(CAMBIOE1J,CAMBIOE2J);
82100 END;
82200 BANDERA1:=TRUE;
82300 FOR CONT1:=0 TO SOLUCI.NELEMCONTJ-1 DO
82400 IF NOT ((CDIGITOSECONT,0,CONT1J='C') OR (CDIGITOSECONT,0,CONT1J
82500 = '*')) THEN
82600 BANDERA1:=FALSE;
82700 FOR CONT1:=0 TO SOLUCI.NELEMCONTJ-1 DO
82800 IF NOT ((CDIGITOSECONT+1,0,CONT1J='C') OR (CDIGITOSECONT+1,0,
82900 CONT1J = '*')) THEN
83000 BANDERA1:=FALSE;
83100 IF BANDERA1 THEN
83200 IF CAMBIOE1J = 'S' THEN
83300 BEGIN
83400 WRITELN('YA CUBRISTE TODOS LOS CUBOS DE ESTOS GRUPOS, SOLO',
83500 ' GENERARIAS COMBINACIONES');
83600 WRITELN('DE LOS QUE YA TIENES, ASI QUE SEGUIREMOS CON LOS',
83700 ' SIGUIENTES GRUPOS');
83800 CAMBIOE1J:='F';
83900 END
84000 ELSE
84100 BEGIN
84200 WRITELN('BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS',
84300 ' DE ESTOS GRUPOS');
84400 IF SOLUCI.NELEMCONT+2J <> 0 THEN
84500 WRITELN('TE PRESENTARE LOS SIGUIENTES');
84600 WRITELN;
84700 CAMBIOE1J:='F';
84800 END;
84900 END
85000 ELSE
85100 BEGIN
85200 BANDERA:=TRUE;
85300 WRITELN('LOS CUBOS QUE PROPONES NO SE COMBINAN, VERIFICA');
85400 CAMBIOE1J:='S';
85500 END;
85600 END
85700 ELSE
85800 BEGIN
85900 IF NOT BANDERA1 THEN
86000 BEGIN
86100 WRITELN('HUBO CUBOS QUE NO SE PUDIERON COMBINAR?');
86200 READLN;
86300 READ(CAMBIOE1J,CAMBIOE2J);
86400 WHILE NOT (((CAMBIOE1J='S') AND (CAMBIOE2J='I')) OR ((CAMBIOE1J='N'
86500 AND (CAMBIOE2J='O')) DO
86501 (* *)
86502 (* *)

```

```

86503 (* *)
86504 (* *)
86600 BEGIN
86700 WRITELN('REPITE SI HUBO CUBOS QUE NO SE PUDIERON COMBINAR?');
86800 READLN; READ(CAMBIOC1J,CAMBIOC2J);
86900 END;
87000 END
87100 ELSE
87200 CAMBIOC1J:='S';
87300 IF CAMBIOC1J = 'S' THEN
87400 BEGIN
87500 CAMBIOC1J:='N';
87600 WHILE CAMBIOC1J = 'N' DO
87700 BEGIN
87800 BANDERA1:=TRUE;
87900 FOR CONT1:=0 TO SOLUCI.NELEM(CONTJ)-1 DO
88000 IF NOT ((CDIGITOSCCONT,0,CONT1)='C') OR (CDIGITOSCCONT,0,CONT1
88100 = '*') THEN
88200 BANDERA1:=FALSE;
88300 FOR CONT1:=0 TO SOLUCI.NELEM(CONTJ)-1 DO
88400 IF NOT ((CDIGITOSCCONT+1,0,CONT1)='C') OR (CDIGITOSCCONT+1,0,
88500 CONT1 = '*') THEN
88600 BANDERA1:=FALSE;
88700 IF BANDERA1 THEN
88800 BEGIN
88900 WRITELN('YA CUBRISTE TODOS LOS CUBOS DE ESTOS GRUPOS, SOLO',
89000 ' GENERARIAS COMBINACIONES');
89100 WRITELN('DE LOS QUE YA TIENES, ASI QUE SEGUIREMOS CON LOS',
89200 ' SIGUIENTES GRUPOS');
89300 CAMBIOC1J:='F';
89400 END;
89500 IF CAMBIOC1J <> 'F' THEN
89600 BEGIN
89700 WRITELN;
89800 WRITELN('DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA');
89900 DECIMALES(CASILLAS,NCASILLAS);
90000 BURBUJA(CASILLAS,NCASILLAS);
90100 IF NCASILLAS = 0 THEN
90200 T1:=0
90300 ELSE
90400 IF NCASILLAS = 1 THEN
90500 T1:=1
90600 ELSE
90700 T1:=RAIZ(NCASILLAS+1);
90800 T:=T1*10+10;
90900 BUSCA(T,CONT2,0,NCASILLAS);
91000 T:=T+1;
91100 WRITELN;
91200 IF BANDERA THEN
91300 IF SOLUCI.ARRMARCACT,CONT2J <> '*' THEN
91400 BEGIN
91500 IF T = CONT THEN
91600 BEGIN
91700 WRITELN('ESTE CUBO SI SE PUEDE COMBINAR, INTENTALO. ');
91800 BANDERA:=FALSE;
91900 END
92000 ELSE
92100 IF T <> (CONT + 1) THEN
92200 BEGIN
92300 WRITELN('ESTE CUBO NO PERTENECE A LOS GRUPOS PRE',
92400 'SENTES, VERIFICA');
92500 BANDERA:=FALSE;
92600 END;
92700 IF BANDERA THEN
92701 (* *)
92702 (* *)

```

```

92703 (* *)
92704 (* *)
92800
92900
93000
93100
93200
93300
93400
93500
93600
93700
93800
93900
94000
94100
94200
94300
94400
94500
94600
94700
94800
94900
95000
95100
95200
95300
95400
95500
95600
95700
95800
95900
96000
96100
96200
96300
96400
96500
96600
96700
96800
96900
97000
97100
97200
97300
97400
97500
97600
97700
97800
97900
98000
98100
98200
98300
98400
98500
98600
98700
98800
98900
98901 (* *)
98902 (* *)

BEGIN
  IF CONT > 0 THEN
    FOR CONT1:=0 TO SOLUCI.NELEM[CONT] - 1 DO
      BEGIN
        BANDERA:=TRUE;
        DIFERENCIA1:=CASILLAS[0] - SOLUCI.ARREGLO[CONT,CONT1];
        FOR CONT3:=0 TO NCASILLAS DO
          BEGIN
            DIFERENCIA:=CASILLAS[CONT3]-SOLUCI.ARREGLO[CONT,CONT1];
            IF NOT ((DIFERENCIA = DIFERENCIA1) AND (DIFERENCIA IN
              UNO) AND (DIFERENCIA > 0)) THEN
              BEGIN
                BANDERA:=FALSE;
                CONT1:=CONT1+1;
              END;
            END;
          IF BANDERA THEN
            CONT1:=SOLUCI.NELEM[CONT];
          ELSE
            CONT1:=CONT1-1;
          END
        END
      ELSE
        IF SOLUCI.NELEM[1] = 0 THEN
          BANDERA:=FALSE;
          IF BANDERA THEN
            BEGIN
              WRITELN('ESTE CUBO SI SE PUEDE COMBINAR, INTENTALO');
              BANDERA:=FALSE;
            END
          ELSE
            BEGIN
              IF CONT = T THEN
                WRITELN('BIEN, ESTE CUBO NO SE PUEDE COMBINAR')
              ELSE
                WRITELN('BIEN, ESTE CUBO NO SE PUEDE COMBINAR (HASTA',
                  ' EL MOMENTO)');
                FOR CONT1:=CONT2 TO (CONT2+NCASILLAS) DO
                  CDIGITOS[CONT+1,0,CONT1]:='*';
                GRAFIC[CONT,2,CONT7];
              END
            END
          END
        ELSE
          IF (T = CONT) OR (T = CONT+1) THEN
            IF CDIGITOS[0,CONT2] = '*' THEN
              WRITELN('OBSERVA LOS GRUPOS, YA SE HABIA VISTO QUE ',
                'ESTE CUBO NO SE COMBINA')
            ELSE
              BEGIN
                FOR CONT1:=CONT2 TO (CONT2+NCASILLAS) DO
                  CDIGITOS[0,CONT1]:='*';
                IF CONT = T THEN
                  WRITELN('BIEN, ESTE CUBO NO SE COMBINA')
                ELSE
                  WRITELN('BIEN, ESTE CUBO NO SE PUEDE COMBINAR (HASTA',
                    ' EL MOMENTO)');
                GRAFIC[CONT,2,CONT7];
              END
            ELSE
              WRITELN('ESTE CUBO NO PERTENECE A LOS GRUPOS QUE ESTAMOS',
                ' ANALIZANDO')
            END
          ELSE

```

```

98903 (* *)
98904 (* *)
99000 WRITELN('NO EXISTE ESTE CUBO, VERIFICA');
99100 BANDERA1:=TRUE;
99200 FOR CONT1:=0 TO SOLUCI.NELEMCONT1-1 DO
99300 IF NOT ((CDIGITOSECONT,0,CONT1]='C') OR (CDIGITOSECONT,0,CONT1
99400 = '*')) THEN
99500 BANDERA1:=FALSE;
99600 FOR CONT1:=0 TO SOLUCI.NELEMCONT1+1-1 DO
99700 IF NOT ((CDIGITOSECONT+1,0,CONT1]='C') OR (CDIGITOSECONT+1,0,
99800 CONT1] = '*')) THEN
99900 BANDERA1:=FALSE;
100000 IF CAMBIOC1] = 'N' THEN
100100 BEGIN
100200 WRITELN('AUN HAY CUBOS QUE NO SE COMBINAN?');
100300 READLN;
100400 READ(CAMBIOC1],CAMBIOC2]);
100500 WHILE NOT (((CAMBIOC1] = 'S') AND (CAMBIOC2] = 'I')) OR
100600 ((CAMBIOC1] = 'N') AND (CAMBIOC2] = 'O')) DO
100700 BEGIN
100800 WRITELN('REPITE SI HAY CUBOS POR CUBRIR O NO');
100900 READLN;
101000 READ(CAMBIOC1],CAMBIOC2]);
101100 END;
101200 IF (CAMBIOC1]='S') AND (NOT BANDERA1) THEN
101300 CAMBIOC1]:='N'
101400 ELSE
101500 IF (NOT BANDERA1) AND (CAMBIOC1] = 'N') THEN
101600 BEGIN
101700 WRITELN('AUN EXISTEN CUBOS POR CUBRIR, OBSERVA LOS GRUPOS');
101800 WRITELN('SE PUEDEN COMBINAR LOS QUE NO TIENEN MARCAS?');
101900 READLN;
102000 READ(CAMBIOC1],CAMBIOC2]);
102100 WHILE NOT (((CAMBIOC1] = 'S') AND (CAMBIOC2] = 'I')) OR
102200 ((CAMBIOC1] = 'N') AND (CAMBIOC2] = 'O')) DO
102300 BEGIN
102400 WRITELN('REPITE SI HAY CUBOS QUE SE COMBINEN O NO');
102500 READLN;
102600 READ(CAMBIOC1],CAMBIOC2]);
102700 END;
102800 END
102900 ELSE
103000 BEGIN
103100 IF (CAMBIOC1] = 'S') AND BANDERA1 THEN
103200 WRITELN('YA CUBRISTE TODOS LOS CUBOS, SOLO GENERARIAS',
103300 'COMBINACIONES ');
103400 ELSE
103500 WRITELN('BIEN, COMPLETASTE CORRECTAMENTE EL ANALISIS',
103600 ' DE ESTOS GRUPOS');
103700 IF SOLUCI.NELEMCONT+2] <> 0 THEN
103800 WRITELN('TE PRESENTARE LOS SIGUIENTES');
103900 CAMBIOC1]:='F';
104000 END;
104100 END;
104200 END;
104300 END;
104400 END
104500 ELSE
104600 BEGIN
104700 BANDERA1:=TRUE;
104800 FOR CONT1:=0 TO SOLUCI.NELEMCONT1-1 DO
104900 IF NOT ((CDIGITOSECONT,0,CONT1]='C') OR (CDIGITOSECONT,0,CONT1
105000 = '*')) THEN
105100 BANDERA1:=FALSE;
105101 (* *)
105102 (* *)

```

```

105103 (* *)
105104 (* *)
105200     FOR CONT1:=0 TO SOLUCI.NELEM[CONT+1]-1 DO
105300     IF NOT ((CDIGITOSE[CONT+1,0,CONT1]='C') OR (CDIGITOSE[CONT+1,0,
105400             CONT1] = '*')) THEN
105500         BANDERA1:=FALSE;
105600     IF NOT BANDERA1 THEN
105700         BEGIN
105800             WRITELN('AUN TIENES CUBOS QUE CUBRIR, INTENTALO');
105900             WRITELN('SE PUEDEN COMBINAR LOS QUE NO TIENEN MARCAS?');
106000         READLN;
106100         READ(CAMBIOC1,CAMBIOC2);
106200         WHILE NOT (((CAMBIOC1 = 'S') AND (CAMBIOC2 = 'I')) OR
106300             ((CAMBIOC1 = 'N') AND (CAMBIOC2 = 'O')) DO
106400             BEGIN
106500                 WRITELN('REPITE SI HAY CUBOS POR COMBINAR O NO');
106600                 READLN;
106700                 READ(CAMBIOC1,CAMBIOC2);
106800             END;
106900         END
107000         ELSE
107100             IF BANDERA1 THEN
107200                 CAMBIOC1:='F';
107300         END;
107400     END;
107500     UNTIL CAMBIOC1 = 'F';
107600     END
107700     ELSE
107800         IF ((SOLUCI.NELEM[CONT] > 0) AND (SOLUCI.NELEM[CONT-1] = 0)) THEN
107900             BEGIN
108000                 CONT7:=CONT MOD 10;
108100                 GRAFIC(CONT,1,CONT7);
108200                 WRITELN('POR NO HABER UN GRUPO ANTERIOR NI UNO POSTERIOR ',
108300                     'CON EL CUAL ESTE SE PUEDA');
108400                 WRITELN('COMBINAR, SE MARCARAN TODOS SUS CUBOS COMO IMPLI',
108500                     'CANTES PRIMOS');
108600                 FOR CONT1:=0 TO SOLUCI.NELEM[CONT]-1 DO
108700                     CDIGITOSE[CONT,0,CONT1]:='*';
108800                 GRAFIC(CONT,1,CONT7);
108900                 CONT:=CONT+1;
109000                 WRITELN('CONTINUA...'); READLN;
109100             END;
109200         END; (*FOR*)
109300         IF CONT6=0 THEN
109400             CONT6:=CONT6+11
109500         ELSE
109600             CONT6:=CONT6+10;
109700         END; (* WHILE *)
109800     END; (* RESUELVEALUM *)
109900
110000     PROCEDURE ACONJUNTO (NDIGS,NIMP;INTEGER;NN;VECTOR)CUBRE;MATCARAC;
110100     VAR CONT:INTEGER;
110200     CONST ELIMINA=1000;
110300     VAR
110400         BANDERA:BOOLEAN;
110500         TOPE:VECTOR;
110600         J1,I,J,K,JJ,K1,K2,POSICION,RELATIVO,MINIMO : INTEGER;
110700     BEGIN
110800         K1:=-1;
110900         FOR I:=0 TO NDIGS DO
111000             BEGIN
111100                 JJ:=0;
111200                 FOR J:=1 TO NIMP DO
111300                     IF CUBRE[J,I]='0' THEN
111301 (* *)
111302 (* *)

```

```

111303 (* *)
111304 (* *)
111400 BEGIN
111500 IF JJ = 0 THEN
111600 WRITE('(',CUBOSEJJ:1)
111700 ELSE
111800 WRITE('+',CUBOSEJJ:1);
111900 CONJUNTOCI,JJ:=CJJ;
112000 JJ:=JJ+1
112100 END;
112200 IF JJ > 0 THEN
112300 WRITE(')');
112400 NCONJUNTOCIJ:=JJ;
112500 END;
112600 WRITELN('=1');
112700 FOR I:=0 TO NDIGS DO
112800 NCONJEXTENCIJ:=NCONJUNTOCIJ;
112900 (* PREPARA LOS CONJUNTOS PARA UNIRSE *)
113000 FOR I:=0 TO (NDIGS-1) DO
113100 FOR J:=(I+1) TO NDIGS DO
113200 BEGIN
113300 JJ:=0;
113400 FOR K:=NCONJEXTENCJJ TO (NCONJEXTENCJJ+NCONJEXTENCIJ-1) DO
113500 BEGIN
113600 CONJUNTOCJ,KJ:=CONJUNTOCJ,JJJ;
113700 JJ:=JJ+1;
113800 END;
113900 NCONJEXTENCJJ:=K;
114000 CONJUNTOCJ,KJ:=CELIMINAJ;
114100 END;
114200 CONJUNTOCO,NCONJUNTOCOJJ:=CELIMINAJ;
114300 FOR I:=0 TO NDIGS-1 DO
114400 BEGIN
114500 J:=0;
114600 WHILE CONJUNTOCI,JJ <> CELIMINAJ DO
114700 BEGIN
114800 FOR K:=J*NCONJUNTOCI+1 TO (NCONJUNTOCI+1)*J(1+J) - 1 DO
114900 CONJUNTOCI+1,KJ:=CONJUNTOCI+1,KJ+CONJUNTOCI,JJ;
115000 J:=J+1;
115100 END;
115200 END;
115300 I:=0;
115400 IF K <> -1 THEN
115500 BEGIN
115600 WRITELN('EFECTUANDO LAS OPERACIONES INDICADAS LLEGAMOS A:');
115700 FOR J:=0 TO K-1 DO
115800 BEGIN
115900 IF J > 0 THEN
116000 BEGIN
116100 I:=I+1;
116200 WRITE('+');
116300 END;
116400 FOR JJ:=1 TO NIMP DO
116500 IF JJ IN CONJUNTOCNDIGS,JJ THEN
116600 BEGIN
116700 I:=I+1;
116800 WRITE(CUBOSEJJJ:1);
116900 END;
117000 IF I >= 72 THEN
117100 BEGIN
117200 WRITELN;
117300 I:=0;
117400 END;
117500 END;
117501 (* *)
117502 (* *)

```



```

117503 (* *)
117504 (* *)
117600      WRITELN('=1');
117700      END;
117800      JJ:=0;
117900      FOR I:=0 TO (NCONJEXTENCNDIGS) DO
118000      BEGIN
118100          BANDERA:=FALSE;
118200          K1:=I;
118300          IF CONJUNTOCNDIGS,IJ <> [ELIMINA] THEN
118400          BEGIN
118500              FOR J:=0 TO (NCONJEXTENCNDIGS-1) DO
118600              IF (J <> I) AND (CONJUNTOCNDIGS,JJ <> [ELIMINA] AND
118700              (CONJUNTOCNDIGS,IJ*CONJUNTOCNDIGS,JJ = CONJUNTOCNDIGS,IJ) THEN
118800              BEGIN
118900                  BANDERA:=TRUE;
119000                  CONJUNTOCNDIGS,JJ:=[ELIMINA];
119100              END;
119200              FOR J:=0 TO NCONJEXTENCNDIGS-1] DO
119300              IF (J <> I) AND (CONJUNTOCNDIGS,JJ <> [ELIMINA] AND
119400              (CONJUNTOCNDIGS,JJ*CONJUNTOCNDIGS,IJ = CONJUNTOCNDIGS,JJ) THEN
119500              BEGIN
119600                  BANDERA:=TRUE;
119700                  CONJUNTOCNDIGS,IJ:=[ELIMINA];
119800                  K1:=J;
119900                  FOR J1:=0 TO NCONJEXTENCNDIGS-1] DO
120000                  IF (J1 <> K1) AND (CONJUNTOCNDIGS,J1] <> [ELIMINA] AND
120100                  (CONJUNTOCNDIGS,K1]*CONJUNTOCNDIGS,J1] =CONJUNTOCNDIGS,K1]) THEN
120200                  CONJUNTOCNDIGS,J1]:=[ELIMINA];
120300              END; BANDERA:=TRUE; WHILE BANDERA DO BEGIN BANDERA:=FALSE;
120400              FOR J:=0 TO (JJ-1) DO
120500              IF CONJUNTOCNDIGS+1,JJ*CONJUNTOCNDIGS,K1] = CONJUNTOCNDIGS,K1]
120600              THEN BEGIN
120700                  FOR J1:=J TO JJ-2 DO
120800                  CONJUNTOCNDIGS+1,J1]:=CONJUNTOCNDIGS+1,J1+1];
120900                  JJ:=JJ-1; BANDERA:=TRUE; J:=JJ;
121000              END;
121100              END;
121200              CONJUNTOCNDIGS+1,JJ]:=CONJUNTOCNDIGS,K1];
121300              JJ:=JJ+1;
121400              IF K1 <> I THEN
121500              CONJUNTOCNDIGS,K1]:=[ELIMINA];
121600          END;
121700      END;
121800      K2:=JJ-1;
121900      WRITELN('FINALMENTE, SIMPLIFICANDO, SE LLEGA A:');
122000      FOR J:=0 TO K2 DO
122100      BEGIN
122200          IF J>0 THEN
122300          WRITE('+');
122400          FOR JJ:=1 TO NIMP DO
122500          IF JJ IN CONJUNTOCNDIGS+1,J] THEN
122600          WRITE(CUBOS[JJ];);
122700      END;
122800      WRITELN('=1');
122900      FOR I:=0 TO K2 DO
123000      BEGIN
123100          J:=1;
123200          K1:=0;
123300          TOPE[I]:=0;
123400          CONJ2:=CONJUNTOCNDIGS+1,I];
123500          WHILE J<=NIMP DO
123600          BEGIN
123700              IF (J IN CONJ2) THEN
123701 (* *)
123702 (* *)

```

```

123703 (* *)
123704 (* *)
123800 BEGIN
123900 TOPECIJ:=TOPECIJ+SOLUCI.NTABSECCJJ;
124000 SOLUCI.MATSOLCI,K1J:=J;
124100 K1:=K1+1;
124200 END;
124300 J:=J+1;
124400 END;
124500 SOLUCI.NMATSOLCIJ:=K1;
124600 END;
124700 (* SE BUSCA EL CONJUNTO CON COSTO MINIMO *)
124800 FOR I:=0 TO K2 DO
124900 MARCAMINCIJ:=' ';
125000 MINIMO:=TOPECOJ;
125100 POSICION:=0;
125200 FOR I:=1 TO K2 DO
125300 IF TOPECIJ<MINIMO THEN
125400 BEGIN
125500 MINIMO:=TOPECIJ;
125600 POSICION:=I;
125700 END;
125800 MARCAMIN(POSICIONJ):='*';
125900 CONT:=K2;
126000 END ; (* ACONJUNTO *)
126100
126200 PROCEDURE SECUNDARIOS(J:INTEGER;VAR II:INTEGER);
126300 VAR
126400 I,JJ,CONT,K : INTEGER;
126500 BEGIN
126600 (* SE OBTIENE LA TABLA DE IMPLICANTES SECUNDARIOS *)
126700 FOR K:=0 TO (SOLUCI.NTABIMPEJJ-1) DO
126800 SOLUCI.IMPSECCII,KJ:=SOLUCI.TABIMPEJJ,KJ;
126900 CONT:=0;
127000 FOR I:=0 TO NDIGITOS DO
127100 IF (SOLUCI.CUBIERTO(I,J,II)='@') AND (SOLUCI.CUBIERTO(CNIMPLIC+1,II,
127200 = ' ') THEN
127300 BEGIN
127400 JJ:=0;
127500 WHILE POSIC(I,J) <> I DO
127600 JJ:=JJ+1;
127700 SOLUCI.NOCUBIERTO(I,II,JJ):='@';
127800 SOLUCI.NTABSECCIIJ:=SOLUCI.NTABIMPEJJ;
127900 NOMFUNSSCIIJ:=NOMFUNSCJJ;
128000 CUBOSCIIJ:=CUBOSCJJ;
128100 CONT:=CONT+1;
128200 END;
128300 IF CONT > 0 THEN
128400 II:=II+1;
128500 END ; (* SECUNDARIOS *)
128600
128700 PROCEDURE SOLUMAQ(VAR CONT:INTEGER;NELEMEN,I:INTEGER;MATRIZ:MATENT);
128800 VAR
128900 BINARYO:ARRAY[0..30,0..30] OF CHAR;
129000 LIM2,K,CONT1,CONT2,CONT3,DIR,DIR1,DECIMAL,DELTA,LIN:INTEGER;
129100 AUXILIAR:STRING;
129200 AUX: CHAR;
129300 BEGIN
129400 DIR:=0;
129500 FOR CONT1:=0 TO (NELEMEN-1) DO
129600 BEGIN
129700 DECIMAL:=MATRIZ(I,CONT1);
129800 FOR CONT2:=0 TO (NVARIA-1) DO
129900 BEGIN
129901 (* *)
129902 (* *)

```

```

129903 (* *)
129904 (* *)
130000 IF (DECIMAL MOD 2) = 1 THEN
130100     BINARYOCDIR,CONT2J:='1'
130200 ELSE
130300     BINARYOCDIR,CONT2J:='0'
130400     DECIMAL:=DECIMAL DIV 2
130500     END
130600     DIR:=DIR+1
130700     END
130800     DELTA:=2
130900     LIM2:=NELEMEN DIV 2
131000     LIM:=LIM2
131100     IF NELEMEN > 1 THEN
131200     FOR CONT3:=1 TO LIM2 DO
131300     BEGIN
131400         WHILE NOT(CONT3 IN UNO) DO
131500             CONT3:=CONT3+1
131600         FOR CONT1:=0 TO (LIM-1) DO
131700         BEGIN
131800             DIR1:=CONT1*DELTA
131900             DIR:=DIR1+CONT3
132000             FOR CONT2:=0 TO (NVARIA-1) DO
132100             IF (BINARYOCDIR,CONT2J) <> (BINARYOCDIR1,CONT2J) THEN
132200                 BINARYOCDIR1,CONT2J:='X'
132300             END
132400             LIM:=LIM DIV 2
132500             DELTA:=DELTA*2
132600         END
132700     FOR CONT1:=1 TO NVARIA DO
132800         AUXILIARECONT1J:=SIMBOLO.LETRACCONT1J
132900     FOR CONT1:=1 TO (NVARIA-1) DO
133000     FOR CONT2:=(CONT1+1) TO NVARIA DO
133100     IF AUXILIARECONT2J < AUXILIARECONT1J THEN
133200     BEGIN
133300         AUX:=AUXILIARECONT1J
133400         AUXILIARECONT1J:=AUXILIARECONT2J
133500         AUXILIARECONT2J:=AUX
133600     END
133700     CONT1:=0
133800     FOR K:=(NVARIA-1) DOWNTO 0 DO
133900     IF BINARYOCD0,KJ<>'X' THEN
134000     BEGIN
134100         SOLUCI.SOLTOTAL[CONT,CONT1J]=AUXILIAREK+1J
134200         SOLUCI.SOLTOTAL[CONT,CONT1+1J]=BINARYOCD0,KJ
134300         CONT1:=CONT1+2
134400         SOLUCI.NSOLTOTAL[CONTJ]=CONT1
134500     END
134600     CONT:=CONT+1
134700 END (* SOLUMAQ *)
134800
134900 PROCEDURE CUBRESECUN
135000 VAR
135100     CONTCUBRE,I,J,II,J1,K,CONT:INTEGER
135200 BEGIN
135300     CONTCUBRE:=0
135400     FOR I:=0 TO NDIGITOS DO
135500     IF SOLUCI.CUBIERTOENIMPLIC+1,II = ' ' THEN
135600         CONTCUBRE:=CONTCUBRE+1
135700     IF CONTCUBRE > 0 THEN BANDERA2:=TRUE ELSE BANDERA2:=FALSE
135800     II:=1
135900     NSECUN:=0
136000     IF CONTCUBRE > 0 THEN
136100     BEGIN
136101 (* *)
136102 (* *)

```

```

136103 (* *)
136104 (* *)
136200 I:=0;
136300 FOR J:=0 TO NDIGITOS DO
136400 IF SOLUCI.CUBIERTO<NIMPLIC+1,JJ = ' ' THEN
136500 BEGIN
136600 DIGSECCIJ:=DIGITOSCJJ;
136700 POSICIJ:=J;
136800 I:=I+1;
136900 END;
137000 NDIGSEC:=I-1;
137100 FOR J:=1 TO NIMPLIC DO
137200 IF SOLUCI.MARCEJJ <> '*' THEN
137300 SECUNDARIOS(J,II);
137400 NIMPSEC:=II-1;
137500 IF BANDTOTAL THEN
137600 ACONJUNTO(NDIGSEC,NIMPSEC,SOLUCI.NTABSEC,SOLUCI.NOCUBIERTO,NSECUN);
137700 END; (* IF *)
137800 CONT:=1;
137900 FOR I:=1 TO (NIMPLIC) DO
138000 IF SOLUCI.MARCEIJ = '*' THEN
138100 SOLUMAQ(CONT,SOLUCI.NTABIMPCIJ,I,SOLUCI.TABIMP);
138200 FOR I:=0 TO NSECUN DO
138300 BEGIN
138400 K:=0;
138500 FOR J:=1 TO CONT-1 DO
138600 BEGIN
138700 J1:=0;
138800 WHILE J1<SOLUCI.NSOLTOTALEJJ DO
138900 BEGIN
139000 SOLUCIONESCI,KJ:=SOLUCI.SOLTOTALEJ,J1;
139100 K:=K+1;
139200 J1:=J1+1;
139300 END;
139400 SOLUCIONESCI,KJ:='+';
139500 K:=K+1;
139600 END;
139700 NSOLUCIONESCIJ:=K-1;
139800 END;
139900 IF (CONTCUBRE > 0) AND BANDTOTAL THEN
140000 BEGIN
140100 FOR I:=0 TO NSECUN DO
140200 FOR J:=0 TO SOLUCI.NMATSOLCIJ-1 DO
140300 BEGIN
140400 SOLUMAQ(CONT,SOLUCI.NTABSECEI,SOLUCI.MATSOLCI,JJ,SOLUCI.MATSOLCI,JJ,
140500 SOLUCI.IMPSEC);
140600 SOLUCIONESCI,NSOLUCIONESCIJ:='+';
140700 K:=0;
140800 J1:=NSOLUCIONESCIJ+2;
140900 WHILE K<SOLUCI.NSOLTOTAL[CONT-1] DO
141000 BEGIN
141100 SOLUCIONESCI,J1:=SOLUCI.SOLTOTAL[CONT-1,KJ];
141200 K:=K+1;
141300 J1:=J1+1;
141400 END;
141500 NSOLUCIONESCIJ:=J1;
141600 END;
141700 END;
141800 TOTSOL:=CONT;
141900 END; (* CUBRESECUN *)
142000
142100 PROCEDURE CUBRIENDO;
142200 VAR
142300 CONT,I1,J1,CONTCUBRE,ESTADO,TOPE,I,J,K,II: INTEGER;
142301 (* *)
142302 ' * *

```

```

142303 (* *)
142304 (* *)
142400 BEGIN
142500   FOR I:=1 TO NIMPLIC DO
142600     BEGIN
142700       TOPE:=0;
142800       CONT:=0;
142900       FOR J:=0 TO NDIGITOS DO
143000         FOR K:=0 TO (SOLUCI.NTABIMFCIJ-1) DO
143100           BEGIN
143200             IF J>0 THEN
143300               IF DIGITOS[J-1] > DIGITOS[J] THEN
143400                 BEGIN
143500                   IF CONT <> SOLUCI.NTABIMFCIJ THEN
143600                     FOR CONT3:=TOPE TO (J-1) DO
143700                       SOLUCI.CUBIERTOICI,CONT3J:=' ';
143800                       TOPE:=J;
143900                       CONT:=0;
144000                     END;
144100                     IF SOLUCI.TABIMFCIJ,KJ = DIGITOS[J] THEN
144200                       BEGIN
144300                         K:=SOLUCI.NTABIMFCIJ;
144400                         SOLUCI.CUBIERTOICI,JJ:='@';
144500                         CONT:=CONT+1;
144600                       END;
144700                     END;
144800                   IF CONT <> SOLUCI.NTABIMFCIJ THEN
144900                     FOR CONT3:=TOPE TO (J-1) DO
145000                       SOLUCI.CUBIERTOICI,CONT3J:=' ';
145100                   END;
145200                   FOR I:=0 TO NDIGITOS DO
145300                     BEGIN
145400                       CONTCUBRE:=0;
145500                       FOR J:=1 TO NIMPLIC DO
145600                         IF SOLUCI.CUBIERTOICJ,IJ = '@' THEN
145700                           BEGIN
145800                             CONTCUBRE:=CONTCUBRE+1;
145900                             ESTADO:=J;
146000                           END;
146100                           IF CONTCUBRE = 1 THEN
146200                             BEGIN
146300                               SOLUCI.MARCEESTADJ:='*';
146400                               FOR K:=0 TO NDIGITOS DO
146500                                 IF SOLUCI.CUBIERTOCESTADJ,KJ = '@' THEN
146600                                   SOLUCI.CUBIERTOICNIMPLIC+1,KJ:='@';
146700                                 END; (* IF *)
146800                               END; (* FOR *)
146900                             CUBRESECUN;
147000                           END; (* CUBRIENDO *)
147100                         END;
147200                       PROCEDURE SOLUCION;
147300                       VAR
147400                         I3,I2,K1,CONT,T1,MODUL,AUX,MED,DIFERENCIA,J1,K2,L,T,I,J,K,I1 : INTEGER;
147500                       BEGIN
147600                         BANDERA:=TRUE;
147700                         SOLUCI.NELEHC-1J:=0;
147800                         FOR I:=0 TO 70 DO
147900                           BEGIN
148000                             FOR J:=0 TO 50 DO
148100                               BEGIN
148200                                 SOLUCI.ARREGLOICI,JJ:=0;
148300                                 SOLUCI.ARRMARCACI,JJ:='0';
148400                               END;
148500                             SOLUCI.NELEHCIJ:=0;
148501 (* *)
148502 (* *)

```

```

148503 (* *)
148504 (* *)
148600 SOLUCI.NSOLTOTALI1:=0;
148700 END;
148800 FOR I:=1 TO 30 DO
148900 BEGIN
149000 FOR J:=0 TO 50 DO
149100 BEGIN
149200 SOLUCI.NOCUBIERTO1,J1:= ' ';
149300 SOLUCI.CUBIERTO1,J1:= ' ';
149400 END;
149500 SOLUCI.MARCCI1:= ' ';
149600 END;
149700 FOR I:=0 TO NDIGITOS DO
149800 CASE DIGITOSI1 OF
149900 0: SOLUCI.NELEM01:=1;
150000 1,2,4,8,16,32,64: BEGIN
150100 SOLUCI.ARREGLOC1,SOLUCI.NELEM11:=DIGITOSI1;
150200 SOLUCI.NELEM11:=SOLUCI.NELEM11+1;
150300 END;
150400 3,5,6,9,10,12,17,18,20,24,33,34,36,40,48: BEGIN
150500 SOLUCI.ARREGLOC2,SOLUCI.NELEM21:=DIGITOSI1;
150600 SOLUCI.NELEM21:=SOLUCI.NELEM21+1;
150700 END;
150800 7,11,13,14,19,21,22,25,26,28,35,37,38,41,42,44,49,50,52,56: BEGIN
150900 SOLUCI.ARREGLOC3,SOLUCI.NELEM31:=DIGITOSI1;
151000 SOLUCI.NELEM31:=SOLUCI.NELEM31+1;
151100 END;
151200 15,23,27,29,30,39,43,45,46,51,53,54,57,58,60: BEGIN
151300 SOLUCI.ARREGLOC4,SOLUCI.NELEM41:=DIGITOSI1;
151400 SOLUCI.NELEM41:=SOLUCI.NELEM41+1;
151500 END;
151600 31,47,55,59,61,62: BEGIN
151700 SOLUCI.ARREGLOC5,SOLUCI.NELEM51:=DIGITOSI1;
151800 SOLUCI.NELEM51:=SOLUCI.NELEM51+1;
151900 END;
152000 63: BEGIN
152100 SOLUCI.ARREGLOC6,SOLUCI.NELEM61:=DIGITOSI1;
152200 SOLUCI.NELEM61:=SOLUCI.NELEM61+1;
152300 END;
152400 END) (* CASE *)
152500 IF SOLUCI.NELEM01=1 THEN
152600 IF SOLUCI.NELEM11>0 THEN
152700 BEGIN
152800 SOLUCI.ARRHARCA10,01:= 'C';
152900 FOR I:=0 TO (SOLUCI.NELEM11*2-1) DO
153000 BEGIN
153100 SOLUCI.ARREGLOC11,I1:=0;
153200 I:=I+1;
153300 MED:=I DIV 2;
153400 CASE SOLUCI.ARREGLOC1,MED OF
153500 1: CDIGITOS11,1,I-11:= '1';
153600 2: CDIGITOS11,1,I-11:= '2';
153700 4: CDIGITOS11,1,I-11:= '4';
153800 8: CDIGITOS11,1,I-11:= '8';
153900 ELSE: CDIGITOS11,1,I-11:= 'X';
154000 END; (* CASE *)
154100 SOLUCI.ARREGLOC11,I1:=SOLUCI.ARREGLOC1,MED;
154200 SOLUCI.ARRHARCA11,MED1:= 'C';
154300 END;
154400 SOLUCI.NELEM11:=SOLUCI.NELEM11*2;
154500 END
154600 ELSE
154700 SOLUCI.ARRHARCA10,01:= '*';
154701 (* *)
154702 (* *)

```

```

154703 (* *)
154704 (* *)
154800 K:=11;
154900 FOR T:=1 TO (NVARIA-1) DO
155000 BEGIN
155100 K:=K+1;
155200 L:=0;
155300 IF ((SOLUCI.NELEMCTJ>0) AND (SOLUCI.NELEMCT+1J>0)) THEN
155400 BEGIN
155500 FOR I:=0 TO (SOLUCI.NELEMCTJ-1) DO
155600 FOR J:=0 TO (SOLUCI.NELEMCT+1J-1) DO
155700 IF ((SOLUCI.ARREGLOCT+1,JJ - SOLUCI.ARREGLOCT,IJ) IN UNO)
155800 AND (SOLUCI.ARREGLOCT+1,JJ-SOLUCI.ARREGLOCT,IJ > 0) THEN
155900 BEGIN
156000 CASE (SOLUCI.ARREGLOCT+1,JJ - SOLUCI.ARREGLOCT,IJ) OF
156100 1: CDIGITOSEK,1,LJ:='1';
156200 2: CDIGITOSEK,1,LJ:='2';
156300 4: CDIGITOSEK,1,LJ:='4';
156400 8: CDIGITOSEK,1,LJ:='8';
156500 ELSE: CDIGITOSEK,1,LJ:='X';
156600 END; (* CASE *)
156700 SOLUCI.ARREGLOCK,LJ:=SOLUCI.ARREGLOCT,IJ;
156800 SOLUCI.ARREGLOCK,L+1J:=SOLUCI.ARREGLOCT+1,JJ;
156900 L:=L+2;
157000 SOLUCI.ARRMARCACT,IJ:='C';
157100 SOLUCI.ARRMARCACT+1,JJ:='C';
157200 END
157300 ELSE
157400 IF SOLUCI.ARRMARCACT,IJ <> 'C' THEN
157500 SOLUCI.ARRMARCACT,IJ:='*';
157600 END
157700 ELSE
157800 FOR I:=0 TO SOLUCI.NELEMCTJ-1 DO
157900 IF SOLUCI.ARRMARCACT,IJ <> 'C' THEN
158000 SOLUCI.ARRMARCACT,IJ:='*';
158100 SOLUCI.NELEMCKJ:=L;
158200 END;
158300 IF SOLUCI.NELEMENVARIAJ > 0 THEN
158400 FOR I:=0 TO SOLUCI.NELEMENVARIAJ-1 DO
158500 IF SOLUCI.ARRMARCAENVARIA,IJ <> 'C' THEN
158600 SOLUCI.ARRMARCAENVARIA,IJ:='*';
158700 K:=21;
158800 T:=11;
158900 K2:=1;
159000 T1:=EXPONENTE(K DIV 10);
159100 WHILE (SOLUCI.NELEMCTJ=0) AND (T<70) DO
159200 BEGIN
159300 T:=T+1;
159400 K:=K+1;
159500 END;
159600 WHILE SOLUCI.ARREGLOCT,IJ <> 0 DO
159700 BEGIN
159800 WHILE (SOLUCI.ARREGLOCT+1,IJ <> 0) OR (SOLUCI.ARREGLOCT,IJ <> 0) DO
159900 IF SOLUCI.ARREGLOCT+1,IJ <> 0 THEN
160000 BEGIN
160100 K1:=0;
160200 BANDERA:=TRUE;
160300 T1:=EXPONENTE(K DIV 10);
160400 FOR I:=0 TO (SOLUCI.NELEMCTJ-1) DO
160500 BEGIN
160600 FOR J:=0 TO (SOLUCI.NELEMCT+1J-1) DO
160700 BEGIN
160800 DIFERENCIA:=SOLUCI.ARREGLOCT+1,JJ-SOLUCI.ARREGLOCT,IJ;
160900 FOR J1:=1 TO K2 DO
160901 (* *)
160902

```

```

160903 (* *)
160904 (* *)
161000 BEGIN
161100 J:=J+1;
161200 I:=I+1;
161300 AUX:=SOLUCI.ARREGLOCT+1,JJ-SOLUCI.ARREGLOCT,IJ;
161400 IF (AUX=DIFERENCIA) AND (AUX>0) AND (AUX IN UNO) THEN
161500 BANDERA:=BANDERA AND TRUE
161600 ELSE
161700 BANDERA:=FALSE;
161800 END;
161900 I:=I-K2;
162000 IF BANDERA THEN
162100 BEGIN
162200 J:=J-K2;
162300 FOR I1:=I TO I+K2 DO
162400 SOLUCI.ARRMARCACT,I1J:='C';
162500 FOR I1:=J TO J+K2 DO
162600 SOLUCI.ARRMARCACT+1,I1J:='C';
162700 I1:=I;
162800 I2:=J;
162900 FOR I3:=K1 TO (K1+K2) DO
163000 BEGIN
163100 PBURBCI3-K1J:=SOLUCI.ARREGLOCT,I1J;
163200 PBURBCI3-K1+K2+1J:=SOLUCI.ARREGLOCT+1,I2J;
163300 I1:=I1+1;
163400 I2:=I2+1;
163500 END;
163600 BURBUJA(PBURB,T1-1);
163700 IF K1>0 THEN
163800 FOR I3:=0 TO ((K1 DIV T1) -1) DO
163900 BEGIN
164000 BANDERA:=TRUE;
164100 CONT:=0;
164200 FOR I1:=T1*I3 TO T1*(I3+1)-1 DO
164300 BEGIN
164400 IF PBURBCONTJ = SOLUCI.ARREGLOCK,I1J THE
164500 BANDERA:=BANDERA AND TRUE
164600 ELSE
164700 BANDERA:=FALSE;
164800 CONT:=CONT+1;
164900 END;
165000 IF BANDERA THEN
165100 I3:=K1;
165200 END
165300 ELSE
165400 BANDERA:=FALSE;
165500 IF NOT BANDERA THEN
165600 BEGIN
165700 CONT:=K1;
165800 FOR I3:=0 TO (T DIV 10) DO
165900 BEGIN
166000 CASE I3 OF
166100 0: I1:=1;
166200 1: I1:=2;
166300 2: I1:=4;
166400 3: I1:=8;
166500 4: I1:=16;
166600 END; (* CASE *)
166700 DIFERENCIA:=PBURBCI1J - PBURBCOJ;
166800 CASE DIFERENCIA OF
166900 1: CDIGITOSCK,1,CONTJ:='1';
167000 2: CDIGITOSCK,1,CONTJ:='2';
167100 4: CDIGITOSCK,1,CONTJ:='4';
167101 (* *)
167102 (* *)

```





```

173303 (* *)
173304 (* *)
173400     END;
173500     SOLUCI,NTABIMPCNIMPLICJ:=T1;
173600     NIMPLIC:=NIMPLIC+1;
173700     END;
173800     END;
173900     IF T>=0 THEN
174000     T:=T-1;
174100     WHILE (SOLUCI.NELEMCTJ=0) AND (T>MODUL) DO
174200     T:=T-1;
174300     END;
174400     IF T >= 0 THEN
174500     T:=T-1;
174600     WHILE (SOLUCI.NELEMCTJ=0) AND (T>0) DO
174700     T:=T-1;
174800     END;
174900     NIMPLIC:=NIMPLIC-1;
175000     CUBRIENDO;
175100     END; (* SOLUCION *)
175200
175300 PROCEDURE IMPRTABLAS(NIMPRENG,DIGS,NOMBREFUN: VECTOR; CUBRE:MATCARAC;
175400     NDIGS,NIMPS:INTEGER; MARCAS,ULCUBRE,MINT:CHARACTER);
175500 VAR
175600     CONT,CONT1: INTEGER;
175700 BEGIN
175800     WRITELN;
175900     WRITELN('M: MARCA DE IMPLICANTES PRIMOS ESCENCIALES');
176000     WRITELN('NF: FUNCIONES INVOLUCRADAS');
176100     WRITELN('IN: NOMBRE ASIGNADO AL IMPLICANTE');
176200     WRITELN;
176300     WRITE('M NF IN');
176400     FOR CONT:=0 TO NDIGS DO
176500     WRITE(' | ',DIGS[CONT];2);
176600     WRITELN(' | ');
176700     FOR CONT:=1 TO NIMPS DO
176800     BEGIN
176900     IF CONT > 1 THEN
177000     IF NIMPRENG[CONT] <> NIMPRENG[CONT-1] THEN
177100     BEGIN
177200     FOR CONT1:=0 TO NDIGS DO
177300     WRITE('-----');
177400     WRITELN('-----');
177500     END;
177600     WRITE(MARCAS[CONT];1,' | ',NOMBREFUN[CONT];3,' | ',MINT[CONT];2);
177700     FOR CONT1:=0 TO NDIGS DO
177800     WRITE(' | ',CUBRE[CONT,CONT1];1);
177900     WRITELN(' | ');
178000     END;
178100     WRITE(' ');
178200     FOR CONT1:=0 TO NDIGS DO
178300     WRITE(' | ',ULCUBRE[CONT1];1);
178400     WRITELN(' | ');
178500     END; (* IMPRTABLAS *)
178600
178700 PROCEDURE CUBREMIN(CUBIERTOS:MATCARAC;NIMPL,NDIGIS:INTEGER;DIGIS:VECTOR)
178800 ; VAR I:INTEGER;
178900 BEGIN
179000     READLN;
179100     I:=0;
179200     BANDERA:=TRUE;
179300     WHILE BANDERA DO
179400     BEGIN
179500     IF I <= NDIGIS THEN
179501 (* *)
179502 /* */

```

```

179503 (* *)
179504 (* *)
179600 READ(CUBALUMCIJ);
179700 WHILE (CUBALUMCIJ = ' ') AND (NOT EOLN) DO
179800   READ(CUBALUMCIJ);
179900   IF (NOT EOLN) OR (CUBALUMCIJ = 'e') OR (CUBALUMCIJ = 'b') THEN
180000     BEGIN
180100       CASE CUBALUMCIJ OF
180200         'e' : CUBALUMCIJ:='@';
180300         'b','b' : CUBALUMCIJ:=' ' ;
180400         ELSE: BEGIN
180500           WRITELN('CARACTER NO RECONOCIDO, REPITE LA LINEA',
180600             ' CORRECTAMENTE');
180700           READLN;
180800           I:=-1;
180900           END;
181000         END; (* CASE *)
181100         I:=I+1;
181200       END
181300     ELSE
181400       IF I <> (NDIGIS+1) THEN
181500         BEGIN
181600           WRITELN('NO CUBRISTE CORRECTAMENTE LOS MINTERMINOS; INTEN',
181700             'TALO NUEVAMENTE');
181800           READLN;
181900           I:=0;
182000           BANDERA:=TRUE;
182100         END
182200       ELSE
182300         BEGIN
182400           BANDERA:=FALSE;
182500           FOR CONT:=0 TO NDIGIS DO
182600             IF CUBALUMCIJ <> CUBIERTOSCINIMPL,CONTJ THEN
182700               BEGIN
182800                 WRITE('REVISA EL MINTERMINO ',DIGISCONTJ:2);
182900                 IF CUBIERTOSCINIMPL,CONTJ = ' ' THEN
183000                   WRITELN(' NO SE PUEDE CUBRIR')
183100                 ELSE
183200                   WRITELN(' SI SE CUBRE');
183300                 BANDERA:=TRUE;
183400               END;
183500             IF BANDERA THEN
183600               BEGIN
183700                 FOR I:=0 TO 30 DO
183800                   CUBALUMCIJ:=' ' ;
183900                   READLN;
184000                 END;
184100                 I:=0;
184200               END;
184300             END;
184400           END; (*CUBREMIN *)
184500
184600 PROCEDURE IMPLICESEN(CUBO,MARCS:CARACTER; NIMP:INTEGER);
184700 BEGIN
184800   BANDERA:=TRUE;
184900   CONT:=1;
185000   .READLN;
185100   WHILE BANDERA DO
185200     BEGIN
185300       FOR CONT1:=1 TO NIMP DO
185400         BEGIN
185500           MARCASCCONT1J:=' ' ;
185600           IMPSCCONT1J:=' ' ;
185700         END;
185701 (* *)
185702 (* *)

```

```

185703 (* *)
185704 (* *)
185800 WHILE NOT EOLN DO
185900 BEGIN
186000 READ(IMPSCONTJ);
186100 WHILE (IMPSCONTJ = ' ') DO
186200 READ(IMPSCONTJ);
186300 IF NOT EOLN THEN
186400 BEGIN
186500 IF (IMPSCONTJ > CUBOCNIMPJ) OR (IMPSCONTJ <
186600 CUBOC1J) THEN
186700 BEGIN
186800 WRITELN(IMPSCONTJ:1, ' NO ES UN IMPLICANTE, VERIFICA');
186900 READLN;
187000 CONT:=0;
187100 END
187200 ELSE
187300 FOR CONT1:=1 TO NIMP DO
187400 IF CUBOCCONT1J = IMPSCONTJ THEN
187500 IF MARCSECONT1J <> '*' THEN
187600 BEGIN
187700 WRITELN(IMPSCONTJ:1, ' NO ES IMPLICANTE PRIMO ',
187800 ' ESCENCIAL, VERIFICA');
187900 CONT:=0;
188000 READLN;
188100 END;
188200 CONT:=CONT+1;
188300 END;
188400 END;
188500 IF EOLN THEN
188600 BEGIN
188700 FOR CONT3:=1 TO CONT DO
188800 FOR CONT4:=1 TO NIMP DO
188900 IF IMPSCONT3J = CUBOCCONT4J THEN
189000 MARCASCCONT4J:='*';
189100 BANDERA1:=TRUE;
189200 FOR CONT3:=1 TO NIMP DO
189300 IF MARCASCCONT3J <> MARCSECONT3J THEN
189400 BANDERA1:=FALSE;
189500 IF BANDERA1 THEN
189600 BANDERA1:=FALSE
189700 ELSE
189800 BEGIN
189900 WRITELN('NO MARCASTE ALGUN IMPLICANTE ESCENCIAL');
190000 CONT:=1;
190100 READLN;
190200 END;
190300 END;
190400 END;
190500 END; (* IMPLICESEN *)
190600
190700 PROCEDURE CADATABLA(NFUNC1:VECTOR;CONT:INTEGER;FUNC1:MATENT;VAR
190800 CONT1:INTEGER);
190900 BEGIN
191000 WRITELN;
191100 IF CONT > 10 THEN
191200 WRITELN('ANALISIS DE LA INTERSECCION DE LAS FUNCIONES ',CONT:2)
191300 ELSE
191400 WRITELN('ANALISIS DE LA FUNCION ',CONT:1);
191500 FOR CONT2:=0 TO 30 DO
191600 MARCANINECONT2J:=' ';
191700 FOR CONT2:=0 TO NFUNC1[CONT] DO
191800 DIGITOS[CONT2J]:=FUNC1[CONT,CONT2J];
191900 NDIGITOS:=NFUNC1[CONT]J;
191901 (* *)
191902 (* *)

```

```

191903 (* *)
191904 (* *)
192000 SOLUCION;
192100 RESUELVEALUM;
192200 FOR CONT2:=0 TO 30 DO
192300   MARCAMIN[CONT2]:=' ';
192400   WRITELN('LOS IMPLICANTES GENERADOS POR ESTA FUNCION SON:');
192500   FOR CONT4:=1 TO NIMPLIC DO
192600     BEGIN
192700       FOR CONT2:=0 TO SOLUCI.NTABIMP[CONT4]-2 DO
192800         WRITE(SOLUCI.TABIMP[CONT4,CONT2]:3,',');
192900         WRITELN(SOLUCI.TABIMP[CONT4,SOLUCI.NTABIMP[CONT4]-1]:3);
193000       END;
193100       IF CONT1 <= TSOL THEN
193200         FOR CONT4:=1 TO NIMPLIC DO
193300           FOR CONT2:=CONT1 TO TSOL DO
193400             BEGIN
193500               BANDERA1:=TRUE;
193600               IF (SOLUCI.NTABIMP[CONT4]-1) = NIMPLIC[CONT2] THEN
193700                 BEGIN
193800                   FOR CONT3:=0 TO NIMPLIC[CONT2] DO
193900                     IF IMPLICANTES[CONT2,CONT3] <> SOLUCI.TABIMP[CONT4,CONT3] THEN
194000                       BANDERA1:=FALSE;
194100                   END
194200                 ELSE
194300                   IF MARCAMIN[CONT4] <> '@' THEN
194400                     BEGIN
194500                       BANDERA1:=FALSE;
194600                       MARCAMIN[CONT4]:=' ';
194700                     END;
194800                   IF BANDERA1 THEN
194900                     MARCAMIN[CONT4]:='@';
195000                   END;
195100                   BANDERA1:=TRUE;
195200                   FOR CONT2:=1 TO NIMPLIC DO
195300                     IF MARCAMIN[CONT2] = '@' THEN
195400                       BANDERA1:=FALSE;
195500                       IF NOT BANDERA1 THEN
195600                         BEGIN
195700                           WRITELN('COMO PODRAS OBSERVAR, EXISTE(N) ALGUN(OS) ',
195800                             'CUBO(S) QUE YA SE HABIA(N)');
195900                           WRITELN('GENERADO DEL ANALISIS DE LA(S) FUNCION(ES) ANTERIOR(ES)');
196000                         END;
196100                           FOR CONT2:=1 TO NIMPLIC DO
196200                             IF MARCAMIN[CONT2] = '@' THEN
196300                               BEGIN
196400                                 CONT1:=CONT1-1;
196500                                 FOR CONT3:=0 TO SOLUCI.NTABIMP[CONT2]-1 DO
196600                                   IMPLICANTES[CONT1,CONT3]:=SOLUCI.TABIMP[CONT2,CONT3];
196700                                   NIMPLIC[CONT1]:=SOLUCI.NTABIMP[CONT2]-1;
196800                                   NOMFUNS[CONT1]:=CONT;
196900                                 END;
197000                                 WRITELN('LOS IMPLICANTES ACUMULADOS HASTA EL MOMENTO SON:');
197100                                 FOR CONT2:=CONT1 TO TSOL DO
197200                                   BEGIN
197300                                     FOR CONT4:=0 TO NIMPLIC[CONT2]-1 DO
197400                                       WRITE(IMPLICANTES[CONT2,CONT4]:3,',');
197500                                       WRITELN(IMPLICANTES[CONT2,NIMPLIC[CONT2]-1]:3);
197600                                     END;
197700                                     WRITELN('CONTINUA...'); READLN;
197800                                 END; (* CADATABLA *)
197900
198000                               (* PROGRAMA PRINCIPAL *)
198100                             BEGIN
198101                               (* *)
198102                               (* *)

```



```

204303 (* *)
204304 (* *)
204400 BEGIN
204500 WRITE('F',CONT:1,'= Sm(');
204600 FOR CONT1:=0 TO NFUNC[CONT]-1 DO
204700 WRITE(FUNCION[CONT,CONT1]:2,',');
204800 WRITELN(FUNCION[CONT,NFUNC[CONT]:2,',');
204900 END;
205000 WRITELN;
205100 WRITELN('DESEAS HACERLES ALGUN CAMBIO?');
205200 READLN; READ(CAMBIO[1],CAMBIO[2]);
205300 WHILE NOT(((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
205400 ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
205500 BEGIN
205600 WRITELN('DESEAS O NO HACER UN CAMBIO?');
205700 READLN;
205800 READ(CAMBIO[1],CAMBIO[2]);
205900 END;
206000 IF CAMBIO[1] = 'S' THEN
206100 BEGIN
206200 WRITELN; WRITELN('DAME EL NUMERO DE LA FUNCION');
206300 READLN; READ(CONT);
206400 WHILE (CONT > NSALIDAS) OR (CONT < 0) DO
206500 BEGIN
206600 WRITELN('VERIFICA EL NUMERO DE LA FUNCION');
206700 READLN; READ(CONT);
206800 END;
206900 WRITELN;
207000 WRITELN('DAME LOS MINTERMINOS DE LA FUNCION ',CONT:1);
207100 DECIMALES(DIGITOS,NFUNC[CONT]);
207200 FOR CONT1:=0 TO NFUNC[CONT] DO
207300 FUNCION[CONT,CONT1]:=DIGITOS[CONT1];
207400 END;
207500 UNTIL CAMBIO[1] = 'N';
207600 BANDERA:=TRUE;
207700 FOR CONT:=1 TO (NSALIDAS-1) DO
207800 FOR CONT1:=(CONT+1) TO NSALIDAS DO
207900 BEGIN
208000 CONT4:=0;
208100 FOR CONT2:=0 TO NFUNC[CONT] DO
208200 FOR CONT3:=0 TO NFUNC[CONT] DO
208300 IF FUNCION[CONT,CONT2] = FUNCION[CONT1,CONT3] THEN
208400 BEGIN
208500 FUNCINTER[CONT*10+CONT1,CONT4]:=FUNCION[CONT,CONT2];
208600 CONT4:=CONT4+1;
208700 CONT3:=NFUNC[CONT];
208800 END;
208900 IF CONT4 > 0 THEN
209000 BEGIN
209100 NFUNCINTER[CONT*10+CONT1]:=CONT4-1;
209200 BANDERA:=FALSE;
209300 END;
209400 END;
209500 CONT1:=20;
209600 TSOL:=20;
209700 IMPLICA:=0;
209800 IF .NOT BANDERA THEN
209900 BEGIN
210000 CONJ1:=[];
210100 CONT:=1;
210200 WHILE NFUNCINTER[CONT] = -1 DO
210300 CONT:=CONT+1;
210400 FOR CONT1:=0 TO NFUNCINTER[CONT] DO
210500 CONJ1:=CONJ1+[FUNCINTER[CONT,CONT1]];
210501 (* *)
210502 (* *)

```

```

210503 (* *)
210504 (* *)
210600 FOR CONT:=1 TO 50 DO
210700 IF NFUNCINTER[CONT] <> -1 THEN
210800 BEGIN
210900     CONJ2:=[];
211000     FOR CONT1:=0 TO NFUNCINTER[CONT] DO
211100         CONJ2:=CONJ2+[FUNCINTER[CONT,CONT1]];
211200     CONJ1:=CONJ1*CONJ2;
211300 END;
211400 CONT1:=21;
211500 IF (CONJ1 <> []) AND (NSALIDAS > 2) THEN
211600 BEGIN
211700     IMPLICA:=1;
211800     CONT1:=-1;
211900     FOR CONT:=0 TO NFUNC[1] DO
212000         IF FUNCION[1,CONT] IN CONJ1 THEN
212100             BEGIN
212200                 CONT1:=CONT1+1;
212300                 FUNCINTER[0,CONT1]:=FUNCION[1,CONT];
212400             END;
212500             NFUNCINTER[0]:=CONT1;
212600             FOR CONT:=0 TO NFUNCINTER[0] DO
212700                 DIGITOS[CONT1]:=FUNCINTER[0,CONT];
212800             NDIGITOS:=NFUNCINTER[0];
212900             WRITELN;
213000             WRITE('ANALISIS DE LA INTERSECCION DE LAS FUNCIONES ');
213100             FOR CONT:=1 TO NSALIDAS-1 DO
213200                 WRITE(CONT:1,',');
213300             WRITELN(NSALIDAS:1);
213400             SOLUCION;
213500             RESUELVEALUM;
213600             CONT1:=20;
213700             FOR CONT:=1 TO NIMPLIC DO
213800                 BEGIN
213900                     FOR CONT2:=0 TO SOLUCI.NTABIMP[CONT]-1 DO
214000                         IMPLICANTES[CONT1,CONT2]:=SOLUCI.TABIMP[CONT,CONT2];
214100                         NIMP[CONT1]:=SOLUCI.NTABIMP[CONT]-1;
214200                         CASE NSALIDAS OF
214300                             1: NOMFUNS[CONT1]:=1;
214400                             2: NOMFUNS[CONT1]:=12;
214500                             3: NOMFUNS[CONT1]:=123;
214600                         END; (* CASE *)
214700                         CONT1:=CONT1+1;
214800                     END;
214900                     WRITELN('LOS IMPLICANTES GENERADOS DE ESTE ANALISIS SON:');
215000                     FOR CONT:=20 TO CONT1-1 DO
215100                         BEGIN
215200                             FOR CONT2:=0 TO NIMP[CONT]-1 DO
215300                                 WRITE(IMPLICANTES[CONT,CONT2]:2,',');
215400                                 WRITELN(IMPLICANTES[CONT,NIMP[CONT]:2);
215500                             END;
215600                             TSOL:=CONT1-1;
215700                             CONT1:=20;
215800                         END;
215900                         BANDERA2:=FALSE;
216000                     FOR CONT:=50 DOWNT0 1 DO
216100                         IF NFUNCINTER[CONT] <> -1 THEN
216200                             CADATABLA(NFUNCINTER,CONT,FUNCINTER,CONT1);
216300                     END
216400                     ELSE
216500                         TSOL:=19;
216600                     FOR CONT:=NSALIDAS DOWNT0 1 DO
216700                         CADATABLA(NFUNC,CONT,FUNCION,CONT1);
216701 (* *)
216702 (* *)

```





```

222903 (* *)
222904 (* *)
223000 WRITELN('A CONTINUACION TE PRESENTO LA TABLA DE IMPLICANTES ',
223100 'PRIMOS, LLENA EL');
223200 WRITELN('ULTIMO RENGLON, SI EXISTE SOLO UNA MARCA EN UNA COLU',
223300 'MNA, MARCA ESTA Y');
223400 WRITELN('TODAS AQUELLAS QUE CONTENGAN A LOS MINTERMINOS DE ESE ',
223500 'IMPLICANTE, SI NO');
223600 WRITELN('SE CUMPLE LO ANTERIOR MARCALA CON EL CARACTER "B"');
223700 IMPRTABLAS(SOLUCI,NTABIMP,DIGITOS,NOMFUNS,SOLUCI.CUBIERTO,NDIGITOS,
223800 NIMPLIC,MARCAS,CUBALUM,CUBO);
223900 WRITELN;
224000 CUBREMEN(SOLUCI.CUBIERTO,NIMPLIC+1,NDIGITOS,DIGITOS);
224100 WRITELN('BIEN, TU TABLA CON EL ULTIMO RENGLON ES:');
224200 IMPRTABLAS(SOLUCI,NTABIMP,DIGITOS,NOMFUNS,SOLUCI.CUBIERTO,NDIGITOS,
224300 NIMPLIC,MARCAS,CUBALUM,CUBO);
224400 BANDERA:=TRUE;
224500 FOR CONT:=0 TO NDIGITOS DO
224600 IF SOLUCI.CUBIERTO[NIMPLIC+1,CONT] <> ' ' THEN
224700 BANDERA:=FALSE;
224800 IF NOT BANDERA THEN
224900 BEGIN
225000 WRITELN;
225100 WRITELN('AHORA DAME LOS IMPLICANTES PRIMOS ESCENCIALES, SEPARADOS ',
225200 'POR UN ESPACIO');
225300 INPLICESEN(CUBO,SOLUCI.MARC,NIMPLIC);
225400 WRITELN('BIEN, MARCASTE CORRECTAMENTE LOS IMPLICANTES PRIMOS',
225500 ' ESCENCIALES');
225600 WRITELN;
225700 WRITELN('LA TABLA COMPLETA ES:');
225800 IMPRTABLAS(SOLUCI,NTABIMP,DIGITOS,NOMFUNS,SOLUCI.CUBIERTO,NDIGITOS,
225900 NIMPLIC,MARCAS,CUBALUM,CUBO);
226000 END;
226100 IF BANDERA2 THEN
226200 BEGIN
226300 WRITELN;
226400 WRITELN('COMO NO SE CUBRIERON TODOS LOS MINTERMINOS, HAY ',
226500 'IMPLICANTES PRIMOS SECUN');
226600 WRITELN('DARIOS, A CONTINUACION TE PRESENTO LA TABLA');
226700 FOR CONT:=0 TO 30 DO
226800 BEGIN
226900 CUBALUM[CONT]:=' ';
227000 MARCASC[CONT]:=' ';
227100 END;
227200 BANDTOTAL:=TRUE;
227300 IMPRTABLAS(SOLUCI,NTABSEC,DIGSEC,NOMFUNS,SOLUCI.NOCUBIERTO,NDIGSEC,
227400 NIMPSEC,MARCAS,CUBALUM,CUBOS);
227500 WRITELN;
227600 WRITELN('USAREMOS UN METODO LLAMADO DE PETRICK PARA RESOLVER ',
227700 'LA TABLA DE IMPLICAN');
227800 WRITELN('TES PRIMOS SECUNDARIOS, A CONTINUACION TE MUESTRO ',
227900 'PASO A PASO EL PROCEDI');
228000 WRITELN('MIENTO PARA HACERLO');
228100 CUBRESECUN;
228200 WRITELN('ESCOGEMOS ARBITRARIAMENTE ALGUN PRODUCTO (CUBO) ',
228300 'QUE CUESTE MENOS, ESTO ES,');
228400 WRITELN('ALGUNO QUE TENGA EL MENOR NUMERO DE ELEMENTOS, ',
228500 'POR EJEMPLO');
228600 FOR CONT:=0 TO NSECUN DO
228700 IF MARCAMINE[CONT] = '*' THEN
228800 BEGIN
228900 FOR CONT1:=0 TO SOLUCI.NMATSOLE[CONT]-1 DO
229000 FOR CONT2:=1 TO NIMPSEC DO
229100 IF SOLUCI.MATSOLE[CONT,CONT1] = CONT2 THEN
229101 (* *)
229102 (* *)

```

```

229103 (* *)
229104 (* *)
229200     WRITE(CUBOS[CONT2]:1);
229300     WRITELN;
229400     CONT3:=CONT;
229500     END;
229600     WRITELN('ESTE PRODUCTO (CUBO) CUBRE LOS MINTERMINOS DE LA ',
229700             'TABLA DE IMPLICANTES SE');
229800     WRITELN('CUNDARIOS DE LOS CUBOS QUE CONTIENE JUNTO CON LOS DE ',
229900             'IMPLICANTES ESCEN');
230000     WRITELN('CIALES, SELECCIONO LOS QUE REQUIERE CADA FUNCION');
230100     END
230200     ELSE
230300     BEGIN
230400         CONT3:=1;
230500         SOLUCI.NMATSOL[1]:=0;
230600     END;
230700     IF (NSECUN = 0) AND (SOLUCI.NSOLTOTAL[1]=0) THEN
230800         WRITELN('LA SOLUCION ES UNA TAUTOLOGIA, ES DECIR, '1'')
230900     ELSE
231000     BEGIN
231100         CONTS:=0; CONT10:=0;
231200         IF IMPLICA = 1 THEN
231300         BEGIN
231400             CASE NSALIDAS OF
231500                 1: CONT1:=1;
231600                 2: CONT1:=12;
231700                 3: CONT1:=123;
231800             END; (* CASE *)
231900             FOR CONT:=1 TO NIMPLIC DO
232000                 IF (NOMFUNCCONTJ = CONT1) AND (SOLUCI.MARCCONTJ = '*') THEN
232100                 BEGIN
232200                     IF CONTS > 0 THEN
232300                         WRITE('+');
232400                     WRITE(CUBOS[CONTJ]:1);
232500                     SOLUHAQ(CONTS,SOLUCI.NTABIMP[CONTJ],CONT,SOLUCI.TABIMP);
232600                     CONT2:=0; VARELIM(CUBOS[CONTJ],CONT2,CONTS-1);
232700                     FOR CONT2:=0 TO SOLUCI.NTABIMP[CONTJ] - 1 DO
232800                         FOR CONT4:=0 TO NDIGITOS DO
232900                             IF DIGITOS[CONT4] = SOLUCI.TABIMP[CONT,CONT2] THEN
233000                                 SOLUCI.CUBIERTO[CONIMPLIC+2,CONT4]:='@';
233100                         END;
233200                     CONT1:=0; CONT9:=CONT10;
233300                     FOR CONT:=1 TO NSALIDAS DO
233400                     BEGIN
233500                         WRITELN('LA SOLUCION PARA LA FUNCION ',CONT:1,' ES ');
233600                         CONT4:=0;
233700                         CONT6:=CONTS; CONT10:=CONT9;
233800                         FOR CONT2:=CONT1 TO (CONT1+NFUNCC[CONTJ]) DO
233900                             IF SOLUCI.CUBIERTO[CONIMPLIC+1,CONT2] = '@' THEN
234000                                 CONT4:=CONT4+1;
234100                             IF (CONT4-1) < NFUNCC[CONTJ] THEN
234200                                 FOR CONT4:=0 TO SOLUCI.NMATSOL[CONT3]-1 DO
234300                                 BEGIN
234400                                     BANDERA:=TRUE;
234500                                     FOR CONT7:=0 TO NIMPLIC DO
234600                                         IF CUBOS[SOLUCI.MATSOL[CONT3,CONT4]] = CUBOS[CONT7] THEN
234700                                             CONT8:=CONT7;
234800                                         FOR CONT2:=CONT1 TO (CONT1+NFUNCC[CONTJ]) DO
234900                                             IF SOLUCI.CUBIERTO[CONT8,CONT2]='@'
235000                                             THEN BEGIN
235100                                                 SOLUCI.CUBIERTO[CONIMPLIC+3,CONT2]:='@';
235200                                                 BANDERA:=FALSE;
235300
235301 (* *)
235302 (* *)

```

```

235303 (* *)
235304 (* *)
235400      END;
235500      IF NOT BANDERA THEN
235600      BEGIN
235700          BANDERA:=TRUE;
235800          FOR CONT2:=CONT1 TO (CONT1+NFUNCCONTJ) DO
235900              IF (SOLUCI.CUBIERTOENIMPLIC+3,CONT2J = '0') AND
236000                  (SOLUCI.CUBIERTOENIMPLIC+2,CONT2J = '0') THEN
236100                  IF SOLUCI.CUBIERTOENIMPLIC+2,CONT2J = ' ' THEN
236200                      BANDERA:=FALSE;
236300              IF NOT BANDERA THEN
236400              BEGIN
236500                  FOR CONT2:=CONT1 TO (CONT1+NFUNCCONTJ) DO
236600                      IF SOLUCI.CUBIERTOENIMPLIC+3,CONT2J = '0' THEN
236700                          SOLUCI.CUBIERTOENIMPLIC+2,CONT2J:='0';
236800          IF CONT6 > CONT5 THEN
236900              WRITE('+');
237000              WRITE(CUBOS(SOLUCI.MATSOLC[CONT3,CONT4J];1);
237100                  SOLUMAQ(CONT6,SOLUCI.NTABSECC[SOLUCI.MATSOLC[CONT3,CONT4J],
237200                      SOLUCI.MATSOLC[CONT3,CONT4J],SOLUCI.IMPSEC]);
237300                  FOR CONT2:=0 TO SOLUCI.NSOLTOTALC[CONT6-1J]-1 DO
237400                      BEGIN
237500                          VARELIM(CUBOS(SOLUCI.MATSOLC[CONT3,CONT4J],CONT2,CONT6-1);
237600                              CONT2:=CONT2-1;
237700                      END;
237800              END;
237900          END;
238000      END;
238100      FOR CONT4:=1 TO NIMPLIC DO
238200          IF SOLUCI.MARCC[CONT4] = '*' THEN BEGIN
238300              BANDERA:=TRUE;
238400              FOR CONT2:=CONT1 TO (CONT1+NFUNCCONTJ) DO
238500                  IF SOLUCI.CUBIERTOENIMPLIC+3,CONT2J = '0' THEN
238600                  BEGIN
238700                      SOLUCI.CUBIERTOENIMPLIC+3,CONT2J:='0';
238800                      BANDERA:=FALSE;
238900                  END;
239000              IF NOT BANDERA THEN
239100              BEGIN
239200                  BANDERA:=TRUE;
239300                  FOR CONT2:=CONT1 TO (CONT1+NFUNCCONTJ) DO
239400                      IF (SOLUCI.CUBIERTOENIMPLIC+3,CONT2J = '0')AND
239500                          (SOLUCI.CUBIERTOENIMPLIC+2,CONT2J = '0') THEN
239600                          IF SOLUCI.CUBIERTOENIMPLIC+2,CONT2J = ' ' THEN
239700                              BANDERA:=FALSE;
239800                          IF NOT BANDERA THEN BEGIN
239900                              FOR CONT2:=CONT1 TO (CONT1+NFUNCCONTJ) DO
240000                                  IF SOLUCI.CUBIERTOENIMPLIC+3,CONT2J = '0' THEN
240100                                      SOLUCI.CUBIERTOENIMPLIC+2,CONT2J:='0';
240200                                  IF CONT6 > CONT5 THEN
240300                                      WRITE('+');
240400                                      WRITE(CUBO(C[CONT4];1);
240500                                          SOLUMAQ(CONT6,SOLUCI.NTABIMP[CONT4],CONT4,SOLUCI.TABIMP);
240600                                          FOR CONT2:=0 TO SOLUCI.NSOLTOTALC[CONT6-1J]-1 DO
240700                                              BEGIN
240800                                                  VARELIM(CUBO(C[CONT4],CONT2,CONT6-1);
240900                                                      CONT2:=CONT2-1;
241000                                                  END;
241100                                          END;
241200                                          END;
241300                                  END;
241400                                  WRITELN;
241500                                  WRITELN('CADA UNO DE ESTOS CUBOS ELIMINO LAS VARIABLES QUE ',
241501 (* *)
241502 (* *)

```

```

241503 (* *)
241504 (* *)
241600 'TE MARCO A CONTINUACION');
241700 WRITELN('CON 'X', AQUELLAS QUE SE MANTIENEN CONSTANTES, NOS',
241800 ' DAN LA SOLUCION');
241900 CONT1:=CONT1+NFUNCC[CONTJ+1];
242000 FOR CONT4:=0 TO CONT10-1 DO
242100 BEGIN
242200 FOR CONT1:=0 TO (NVARIA+3) DO
242300 WRITE(SOLUCIONES[CONT4,CONT1J:1]);
242400 WRITELN;
242500 END;
242600 WRITELN('POR LO CUAL, LA SOLUCION EN FUNCION DE LAS VARIABLES ',
242700 'PROPUESTAS ES:');
242800 FOR CONT4:=0 TO (CONT6-2) DO
242900 BEGIN
243000 FOR CONT2:=0 TO (SOLUCI.NSOLTOTAL[CONT4J]-1) DO
243100 WRITE(SOLUCI.SOLTOTAL[CONT4,CONT2J:1]);
243200 WRITE('+');
243300 END;
243400 FOR CONT4:=0 TO SOLUCI.NSOLTOTAL[CONT6-1J]-1 DO
243500 WRITE(SOLUCI.SOLTOTAL[CONT6-1,CONT4J:1]);
243600 WRITELN;
243700 END;
243800 END;
243900 END.
#

```

r quine/multiple

#RUNNING 5256

ESTE PROGRAMA TE GUIARA EN LA MINIMIZACION DE FUNCIONES A TRAVES DEL METODO DE QUINE-McCLUSKEY CON LA OPCION A MULTIPLES SALIDAS.

DESEAS CONOCER UNA PEQUEÑA INTRODUCCION A ESTE METODO?

#?

SI

LOS MINTERMINOS DE UNA FUNCION SON LLAMADOS CUBOS-0.

EL METODO CONSISTE EN UNA BUSQUEDA EXHAUSTIVA DE TODAS LAS COMBINACIONES POSIBLES DE LOS CUBOS-N PARA FORMAR LOS CUBOS-(N+1) Y LUEGO SELECCIONAR LA COMBINACION MINIMA DE CUBOS PARA REALIZAR LA(S) FUNCION(ES).

SE ANALIZA LA INTERSECCION DE LAS FUNCIONES Y CADA UNA DE ELLAS INDEPENDIENTEMENTE.

SE INICIA OBTENIENDO LA REPRESENTACION BINARIA DE LOS CUBOS-0 Y SE AGRUPAN EN FUNCION DEL NUMERO DE "1'S".

LOS CUBOS-1 SE OBTIENEN COMPARANDO GRUPOS CONSECUTIVOS POR PARES. EJEMPLO:

GRUPO 1	GRUPO 2
001 1	011 3
010 2	110 6

SI COMPARAMOS UN CUBO DEL GRUPO 2 CON OTRO DEL GRUPO 1 Y LA DIFERENCIA ES POTENCIA DE 2 HEMOS OBTENIDO UN CUBO-1.

$$\text{CUBO-0(GRUPO 2)} - \text{CUBO-0(GRUPO 1)} = 3 - 2 = 1$$

OBSERVA QUE SE HA HECHO LA DIFERENCIA DEL GRUPO MAYOR - GRUPO MENOR, LO CONTRARIO NO ES CIERTO.

ESTE PROCEDIMIENTO SE CONTINUA HASTA CURRIR TODOS LOS CUBOS DE TODOS LOS GRUPOS.

PARA OBTENER LOS CUBOS-2 HASTA CUBOS-N DEBES OBSERVAR LA MISMA CONDICION DE DIFERENCIA DE POTENCIA DE 2 ENTRE LOS CUBOS DE CADA GRUPO CONSECUTIVO EN SUS ELEMENTOS CORRESPONDIENTES Y ADEMAS QUE LA(S) DIFERENCIA(S) ANTERIORES, LA CUAL TE PRESENTARE EN ESTE PROGRAMA ENTRE PARENTESIS SEA LA MISMA.

CONTINUA...

SI ALGUN CUBO DE ALGUN GRUPO NO SE COMBINA, LO MARCARAS, ESTE FORMARA PARTE DE LOS IMPLICANTES PRIMOS.

LOS IMPLICANTES PRIMOS SON TODOS AQUELLOS CUBOS QUE DURANTE EL PROCESO DE COMPARACION NO LOGRARON COMBINARSE, DE ENTRE ELLOS SE SELECCIONARAN AQUELLOS QUE NOS DARAN LA SOLUCION MINIMA PARA CADA UNA DE LAS FUNCIONES. ESTOS IMPLICANTES SE ORDENAN EN UNA TABLA EN FUNCION DE LA CANTIDAD DE CUBOS-0 QUE CONTENGAN.

ESTA TABLA TENDRA EN SUS RENGLONES LOS IMPLICANTES ORDENADOS Y EN SUS COLUMNAS TODOS LOS CUBOS-0 DE TODAS LAS FUNCIONES.

DE CADA IMPLICANTE SE BUSCARAN EN LAS COLUMNAS, LOS CUBOS-0 QUE LO FORMAN Y SE MARCARA LA COLUMNA CORRESPONDIENTE.

QUE LE PERTENEZCAN.

EL PASO SIGUIENTE ES VERIFICAR CADA COLUMNA QUE CONTenga SOLO UNA MARCA, CUANDO SE ENCUENTREN, SE MARCARAN EN UN RENGLON ADICIONAL AL IGUAL QUE TODAS AQUELLAS QUE ESTEN MARCADAS EN EL RENGLON QUE SE ENCONTRO SOLO UNA MARCA, ESTE IMPLICANTE SE CONVIERTE EN ESCENCIAL Y FORMARA PARTE DE LA SOLUCION.

SI AL TERMINAR DE REVISAR TODAS LAS COLUMNAS Y MARCADO LOS IMPLICANTES ESCENCIALES HAY COLUMNAS QUE NO SE MARCARON, PASAREMOS ESTAS COLUMNAS A OTRA TABLA ASI COMO LOS IMPLICANTES QUE LAS CONTENGAN.

CONTINUA...

EXISTEN DIVERSOS METODOS PARA RESOLVER ESTA ULTIMA TABLA, EN ESTE PROGRAMA LO HAREMOS A TRAVES DE UN METODO LLAMADO DE PETRICK, EL CUAL CONSISTE EN TOMAR LOS IMPLICANTES QUE FORMAN ESTA TABLA E INTERPRETARLAS COMO VARIABLES BOOLEANAS, ESTO ES TOMARAN VALORES CIERTO O FALSO, DEPENDIENDO DE SI SE ESCOGE O NO PARA EL ANALISIS.

A PARTIR DE LA PRIMERA COLUMNA SE DEBEN SELECCIONAR CUALQUIERA DE LOS IMPLICANTES QUE TENGAN MARCA EN ELLA PARA CUBRIRLO DE MANERA QUE SI REPRESENTAMOS CADA IMPLICANTE POR UNA LETRA Y *a* ES EL PRIMER IMPLICANTE Y *d* EL TERCERO DE LA TABLA SECUNDARIA, Y AMBOS TIENEN MARCADOS EN ALGUNA DE LAS COLUMNAS, SE EXPRESARAN POR:

$$(a+d)=1$$

YA QUE CUALQUIERA DE LOS DOS CUBRE ESE CUBO, DEBIDO A TODOS ELLOS DEBEN CUBRIRSE, SE UNIRAN POR EL CONECTOR "Y", ESTO ES, SI OTRA COLUMNA CONTIENE POR EJEMPLO LOS CUBOS *c* y *f* MARCADOS, SE TENDRA LA EXPRESION:

$$(a+b)(c+f)=1$$

ESTE PROCESO SE SIGUE HASTA INVOLUCRAR TODAS LAS COLUMNAS.

ENSEGUIDA SE REALIZARAN LOS PRODUCTOS INDICADOS Y SE SIMPLIFICARA LA EXPRESION RESULTANTE POR MEDIO DE LOS TEOREMAS FUNDAMENTALES DEL ALGEBRA DE BOOLE, POR LO MENOS ALGUNO DE LOS PRODUCTOS DEBE SER CIERTO (1), DE ESTOS SE SELECCIONA ALGUNO CUYO COSTO SEA MENOR, ESTO ES, EL NUMERO DE ENTRADAS ES MAS REDUCIDO.

LOS CUBOS QUE CONTENGA ESTE PRODUCTO MAS LOS IMPLICANTES DE LA PRIMERA TABLA, DAN UNA SOLUCION MINIMA.

CONTINUA...

ANALIZANDO CADA IMPLICANTE ESCENCIAL Y SECUNDARIO EN FORMA INDEPENDIENTE COMPARAMOS LAS POSICIONES CORRESPONDIENTES DE TODAS LAS REPRESENTACIONES BINARIAS DE LOS CUBOS-0 QUE LO COMPONEN.

LOS VALORES DE ESTAS POSICIONES QUE SE MANTENGAN CONSTANTES SE ASOCIAN A LA VARIABLE QUE LE CORRESPONDA POR SU POSICION Y ESTO NOS DA LA SOLUCION DE LA FUNCION CON LAS VARIABLES DE QUE ESTA DEFINIDA.

EJEMPLO:

SUPONIENDO QUE LA SOLUCION ESTA EXPRESADA POR LOS IMPLICANTES:

$$F(A,B,C) = a+b$$

SIENDO  $a = 0,1,2,3$

$b = 0,1,4,5$

ANALIZANDO  $a$ :

CBA  
 $0 = 000$   
 $1 = 001$   
 $2 = 010$   
 $3 = 011$

ANALIZANDO  $b$ :

CBA  
 $0 = 000$   
 $1 = 001$   
 $4 = 100$   
 $5 = 101$

OBSERVAMOS QUE C SE MANTIENE CONSTANTE EN LOS 4 CUBOS-0 DE  $a$  y B en  $b$  Y EL VALOR ASOCIADO ES 0 (EN ESTE CASO) PARA AMBOS

$\Rightarrow a=C0$  Y  $b=B0$

Y LA SOLUCION DE LA FUNCION EXPRESADA CON SUS VARIABLES ES:

$$f(A,B,C) = a+b = C0 + B0$$

CONTINUA...

¿HABRÁ BIEN CUANTAS SALIDAS DESEAS USAR?  
2

¿CUANTAS ENTRADAS TIENEN TUS FUNCIONES?  
3

LOS MINTERMINOS DE CADA UNA DE LAS FUNCIONES DEBERAS PROPORCIONARMELOS EN FORMA DECIMAL SEPARADOS POR COMAS "," Y TERMINAR CON ";"

DAME LOS MINTERMINOS DE LA FUNCION 1  
0,2,6,7;

DAME LOS MINTERMINOS DE LA FUNCION 2  
4,5,6,7;

TUS FUNCIONES SON:

F1= Sm( 0, 2, 6, 7)  
F2= Sm( 4, 5, 6, 7)

¿DESEAS HACERLES ALGUN CAMBIO?  
NO

ANALISIS DE LA INTERSECCION DE LAS FUNCIONES 1 2

TUS MINTERMINOS (CUBOS-0) ORDENADOS EN FUNCION DEL NUMERO DE UNOS CON SU REPRESENTACION BINARIA Y DECIMAL SON:

GRUPO 2	GRUPO 3
110 6	111 7

DE AQUI EN ADELANTE, USAREMOS LOS MINTERMINOS EN SU REPRESENTACION DECIMAL

ANALISIS DE LOS CUBOS 0 PARA OBTENER LOS CUBOS 1

GRUPO 2	GRUPO 3
6	7

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA POTENCIA DE 2?  
SI

DAME UNA PAREJA DE ELLOS  
6,7;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 2	GRUPO 3
6C	7C

AUN HAY CUBOS POR COMBINAR?  
NO



BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS

ANALISIS DE LOS CUBOS 1 PARA OBTENER LOS CUBOS 2

PARA QUE SE PUEDAN COMBINAR CUBOS DE LOS SIGUIENTES GRUPOS, SE DEBE CUMPLIR CON:

- A) LA(S) DIFERENCIA(S) INDICADA(S) EN EL PARENTESIS DEBE(N) SER LA(S) MISMA(S), YA QUE CORRESPONDE A LA MISMA VARIABLE ELIMINADA
- B) LA(S) DIFERENCIA(S) ENTRE LOS ELEMENTOS CORRESPONDIENTES ENTRE LOS CUBOS DE DIFERENTES GRUPOS DEBE SER IGUAL Y POTENCIA DE 2

GRUPO 3

6, 7 (1)

POR NO HABER UN GRUPO ANTERIOR NI UNO POSTERIOR CON EL CUAL ESTE SE PUEDA COMBINAR, SE MARCARAN TODOS SUS CUBOS COMO IMPLICANTES PRIMOS

GRUPO 3

6, 7 (1)\*

CONTINUA...

LOS IMPLICANTES GENERADOS POR ESTA FUNCION SON:

6, 7

LOS IMPLICANTES ACUMULADOS HASTA EL MOMENTO SON:

6, 7

CONTINUA...

ANALISIS DE LA FUNCION 2

TUS MINTERMINOS (CUBOS-0) ORDENADOS EN FUNCION DEL NUMERO DE UNOS CON SU REPRESENTACION BINARIA Y DECIMAL SON:

GRUPO 1	GRUPO 2	GRUPO 3
100 4	101 5	111 7
	110 6	

DE AQUI EN ADELANTE, USAREMOS LOS MINTERMINOS EN SU REPRESENTACION DECIMAL

ANALISIS DE LOS CUBOS 0 PARA OBTENER LOS CUBOS 1

GRUPO 1	GRUPO 2
4	5
	6

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA POTENCIA DE 2?  
SI

DAME UNA PAREJA DE ELLOS

4,5

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 1	GRUPO 2
4C	5C
	6

AUN HAY CUBOS POR COMBINAR?

SI

DAME UNA PAREJA DE ELLOS  
4,6;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 1	GRUPO 2
4C	5C
	6C

AUN HAY CUBOS POR COMBINAR?

NO

BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS  
TE PRESENTARE LOS SIGUIENTES

GRUPO 2	GRUPO 3
5C	7
6C	

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA  
POTENCIA DE 2?

SI

DAME UNA PAREJA DE ELLOS  
5,7;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 2	GRUPO 3
5C	7C
6C	

AUN HAY CUBOS POR COMBINAR?

NO

BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS

ANALISIS DE LOS CUBOS 1 PARA OBTENER LOS CUBOS 2

PARA QUE SE PUEBAN COMBINAR CUBOS DE LOS SIGUIENTES GRUPOS, SE DEBE  
CUMPLIR CON:

- A) LA(S) DIFERENCIA(S) INDICADA(S) EN EL PARENTESIS DEBE(N) SER LA(S)  
MISMA(S), YA QUE CORRESPONDE A LA MISMA VARIABLE ELIMINADA
- B) LA(S) DIFERENCIA(S) ENTRE LOS ELEMENTOS CORRESPONDIENTES ENTRE LOS  
CUBOS DE DIFERENTES GRUPOS DEBE SER IGUAL Y POTENCIA DE 2

GRUPO 2	GRUPO 3
4, 5 (1)	5, 7 (2)
4, 6 (2)	6, 7 (1)

EXISTEN EN ESTOS GRUPOS CUBOS QUE SE COMBINEN?

SI

DAME UNA PAREJA DE ELLOS  
4,5,6,7;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 2	GRUPO 3
4, 5 (1)C	5, 7 (2)
4, 6 (2)	6, 7 (1)C

AUN HAY CUBOS POR COMBINAR?

SI

DAME UNA PAREJA DE ELLOS  
4,6,5,7;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 2	GRUPO 3
4, 5 (1)C	5, 7 (2)C
4, 6 (2)C	6, 7 (1)C

AUN HAY CUBOS POR COMBINAR?

SI

YA CUBRISTE TODOS LOS CUBOS DE ESTOS GRUPOS, SOLO GENERARIAS COMBINACIONES DE LOS QUE YA TIENES, ASI QUE SEGUIREMOS CON LOS SIGUIENTES GRUPOS

ANALISIS DE LOS CUBOS 2 PARA OBTENER LOS CUBOS 3

PARA QUE SE PUEDAN COMBINAR CUBOS DE LOS SIGUIENTES GRUPOS, SE DEBE CUMPLIR CON:

A) LA(S) DIFERENCIA(S) INDICADA(S) EN EL PARENTESIS DEBE(N) SER LA(S) MISMA(S), YA QUE CORRESPONDE A LA MISMA VARIABLE ELIMINADA  
B) LA(S) DIFERENCIA(S) ENTRE LOS ELEMENTOS CORRESPONDIENTES ENTRE LOS CUBOS DE DIFERENTES GRUPOS DEBE SER IGUAL Y POTENCIA DE 2

GRUPO 2

4, 5, 6, 7 (1,2)

POR NO HABER UN GRUPO ANTERIOR NI UNO POSTERIOR CON EL CUAL ESTE SE PUEDA COMBINAR, SE MARCARAN TODOS SUS CUBOS COMO IMPLICANTES PRIMOS

GRUPO 2

4, 5, 6, 7 (1,2)\*

CONTINUA...

LOS IMPLICANTES GENERADOS POR ESTA FUNCION SON:

4, 5, 6, 7

LOS IMPLICANTES ACUMULADOS HASTA EL MOMENTO SON:

4, 5, 6, 7

6, 7

CONTINUA...

ANALISIS DE LA FUNCION 1

TUS MINTERMINOS (CUBOS-0) ORDENADOS EN FUNCION DEL NUMERO DE UNOS CON SU REPRESENTACION BINARIA Y DECIMAL SON:

GRUPO 0	GRUPO 1	GRUPO 2	GRUPO 3
000 0	010 2	110 6	111 7

DE AQUI EN ADELANTE, USAREMOS LOS MINTERMINOS EN SU REPRESENTACION DECIMAL

ANALISIS DE LOS CUBOS 0 PARA OBTENER LOS CUBOS 1

GRUPO 0      GRUPO 1

0                      2

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA POTENCIA DE 2?

SI

DAME UNA PAREJA DE ELLOS

0,2;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 0      GRUPO 1  
0C            2C

AUN HAY CUBOS POR COMBINAR?

NO

BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS  
TE PRESENTARE LOS SIGUIENTES

GRUPO 1      GRUPO 2  
2C            6

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA  
POTENCIA DE 2?

SI

DAME UNA PAREJA DE ELLOS

2,6;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 1      GRUPO 2  
2C            6C

AUN HAY CUBOS POR COMBINAR?

NO

BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS  
TE PRESENTARE LOS SIGUIENTES

GRUPO 2      GRUPO 3  
6C            7

EXISTEN EN ESTOS GRUPOS UN CUBO DE CADA UNO, CUYA DIFERENCIA SEA UNA  
POTENCIA DE 2?

SI

DAME UNA PAREJA DE ELLOS

6,7;

BIEN, ESTOS CUBOS SI SE COMBINAN

GRUPO 2      GRUPO 3  
6C            7C

AUN HAY CUBOS POR COMBINAR?

NO

BIEN, COMPLETASTE CORRECTAMENTE TU ANALISIS DE ESTOS GRUPOS

ANALISIS DE LOS CUBOS 1 PARA OBTENER LOS CUBOS 2

PARA QUE SE PUEDAN COMBINAR CUBOS DE LOS SIGUIENTES GRUPOS, SE DEBE  
CUMPLIR CON:

A) LA(S) DIFERENCIA(S) INDICADA(S) EN EL PARENTESIS DEBE(N) SER LA(S)  
MISMA(S); YA QUE CORRESPONDE A LA MISMA VARIABLE ELIMINADA

B) LA(S) DIFERENCIA(S) ENTRE LOS ELEMENTOS CORRESPONDIENTES ENTRE LOS  
CUBOS DE DIFERENTES GRUPOS DEBE SER IGUAL Y POTENCIA DE 2

GRUPO 1                      GRUPO 2  
0, 2 (2)                      2, 6 (4)

EXISTEN EN ESTOS GRUPOS CUBOS QUE SE COMBINEN?

NO

HUBO CUBOS QUE NO SE PUDIERON COMBINAR?  
SI

DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA  
0,2;

BIEN, ESTE CUBO NO SE COMBINA  
GRUPO 1 GRUPO 2  
0, 2 (2)\* 2, 6 (4)\*

AUN HAY CUBOS QUE NO SE COMBINAN?  
SI

DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA  
2,6;

BIEN, ESTE CUBO NO SE PUEDE COMBINAR (HASTA EL MOMENTO)  
GRUPO 1 GRUPO 2  
0, 2 (2)\* 2, 6 (4)\*

AUN HAY CUBOS QUE NO SE COMBINAN?  
NO

BIEN, COMPLETASTE CORRECTAMENTE EL ANALISIS DE ESTOS GRUPOS  
TE PRESENTARE LOS SIGUIENTES

GRUPO 2 GRUPO 3  
2, 6 (4)\* 6, 7 (1)

EXISTEN EN ESTOS GRUPOS CUBOS QUE SE COMBINEN?

NO

HUBO CUBOS QUE NO SE PUDIERON COMBINAR?  
SI

DAME ALGUNO DE LOS CUBOS QUE NO SE COMBINA  
6,7;

BIEN, ESTE CUBO NO SE PUEDE COMBINAR (HASTA EL MOMENTO)  
GRUPO 2 GRUPO 3  
2, 6 (4)\* 6, 7 (1)\*

AUN HAY CUBOS QUE NO SE COMBINAN?  
NO

BIEN, COMPLETASTE CORRECTAMENTE EL ANALISIS DE ESTOS GRUPOS  
LOS IMPLICANTES GENERADOS POR ESTA FUNCION SON:

6, 7  
2, 6  
0, 2

COMO PODRAS OBSERVAR, EXISTE(N) ALGUN(OS) CUBO(S) QUE YA SE HABIA(N)  
GENERADO DEL ANALISIS DE LA(S) FUNCION(ES) ANTERIOR(ES)  
LOS IMPLICANTES ACUMULADOS HASTA EL MOMENTO SON:

0, 2  
2, 6  
4, 5, 6, 7  
6, 7

CONTINUA...

A CONTINUACION TE PRESENTO LA TABLA DE IMPLICANTES PRIMOS, LLENA EL ULTIMO RENGLON, SI EXISTE SOLO UNA MARCA EN UNA COLUMNA, MARCA ESTA Y TODAS AQUELLAS QUE CONTENGAN A LOS MINTERMINOS DE ESE IMPLICANTE, SI NO SE CUMPLE LO ANTERIOR MARCALA CON EL CARACTER "B"

M: MARCA DE IMPLICANTES PRIMOS ESCENCIALES  
 NF: FUNCIONES INVOLUCRADAS  
 IM: NOMBRE ASIGNADO AL IMPLICANTE

M	NF	IM	0	2	6	7	4	5	6	7
1	2	a					0	0	0	0
1	1	b	0	0						
1	1	c		0	0					
1	2	d			0	0			0	0
			0	0	0	0	0	0	0	0

BIEN, TU TABLA CON EL ULTIMO RENGLON ES:

M: MARCA DE IMPLICANTES PRIMOS ESCENCIALES  
 NF: FUNCIONES INVOLUCRADAS  
 IM: NOMBRE ASIGNADO AL IMPLICANTE

M	NF	IM	0	2	6	7	4	5	6	7
1	2	a					0	0	0	0
1	1	b	0	0						
1	1	c		0	0					
1	2	d			0	0			0	0
			0	0	0	0	0	0	0	0

AHORA DAME LOS IMPLICANTES PRIMOS ESCENCIALES, SEPARADOS POR UN ESPACIO  
 a b d

BIEN, MARCASTE CORRECTAMENTE LOS IMPLICANTES PRIMOS ESCENCIALES

LA TABLA COMPLETA ES:

M: MARCA DE IMPLICANTES PRIMOS ESCENCIALES  
 NF: FUNCIONES INVOLUCRADAS  
 IM: NOMBRE ASIGNADO AL IMPLICANTE

M	NF	IM	0	2	6	7	4	5	6	7
*1	2	a					0	0	0	0
*1	1	b	0	0						
*1	1	c		0	0					
*1	2	d			0	0			0	0
			0	0	0	0	0	0	0	0

LA SOLUCION PARA LA FUNCION 1 ES

b+d  
 CADA UNO DE ESTOS CUBOS ELIMINO LAS VARIABLES QUE TE MARCO A CONTINUACION CON "X", AQUELLAS QUE SE MANTIENEN CONSTANTES, NOS DAN LA SOLUCION

b = 0X0

d = 11X

POR LO CUAL, LA SOLUCION EN FUNCION DE LAS VARIABLES PROPUESTAS ES:

COA0+C1B1

LA SOLUCION PARA LA FUNCION 2 ES

a  
CADA UNO DE ESTOS CUBOS ELIMINO LAS VARIABLES QUE TE MARCO A CONTINUACION  
CON 'X', AQUELLAS QUE SE MANTIENEN CONSTANTES, NOS DAN LA SOLUCION

a = 1X0

POR LO CUAL, LA SOLUCION EN FUNCION DE LAS VARIABLES PROPUESTAS ES:

C1

#ET=17:43.7 PT=4.7 IO=0.2

### VI.3 MANUAL DE USUARIO

El comando necesario para correr este programa dependera de la maquina que se utilice, por lo que debera investigarse en el lugar correspondiente.

Siempre que se conteste algo a la maquina, debe terminarse presionando la tecla <RETURN>. Si se trata de una decision (SI o NO) deberan proporcionarse los dos caracteres.

De entrada el programa dara la oportunidad de conocer una pequena introduccion al metodo.

A continuacion solicita el numero de funciones a minimizar, este debe ser un entero no mayor de 3 ni menor de 1.

Posteriormente solicita el numero de entradas (variables) de la(s) funcion(es).

El producto del numero de entradas y de funciones no debe ser mayor de 16 ni menor de 1.

Enseguida pedira los minterminos de cada funcion, estos deberan darse en forma decimal, separados por comas "," y terminar con el caracter "!" para cada una de ellas.

Cuando el usuario ha proporcionado todas las funciones, la maquina se las presentara con el objeto de que pueda verificarlas y hacer algun cambio si asi lo desea.

Si hay correcciones, pedira el numero de la funcion que se cambiara y enseguida todos los minterminos correctos de ella.

Con esta informacion se inicia el proceso de minimizacion.

Si existe interseccion entre los cubos-0 de las funciones, se empezara resolviendo cada una de estas intersecciones como una funcion mas y posteriormente cada funcion independientemente.



## SOLUCION DE CADA FUNCION

El programa le indicara al usuario el numero de funcion o interseccion de ellas que le presentara para su analisis, enseguida los cubos-0 correspondientes ordenados en grupos en funcion del numero de '1's de su representacion binaria. A partir de este momento, se presentaran los grupos por parejas para que el usuario revise la posible combinacion de los cubos que los integran.

El analisis de los cubos se divide propiamente en dos pasos:

- 1) Combinar todos los cubos posibles
- 2) Indicar aquellos que no se combinaron.

Para lo cual se preguntara al usuario si hay cubos que se combinen en esos grupos.

- 1) Si hay cubos que se combinen. Debera darse una pareja de ellos. Ejemplo:

GRUPO 1	GRUPO 2
0,1 (1)	4,6(2)
0,2 (2)	
0,4 (4)	

como a continuacion se indica:

0,2,4,6;

Notese que se proporciono primero el cubo del grupo menor y despues el del grupo mayor; todos los elementos se separaron por comas y se termino con el caracter ';'.

- 2) Existen cubos que no se combinan. Terminados de combinar los cubos que son posibles, el programa preguntara si existen cubos que no se combinan, habiendo contestado afirmativamente, pedira estos cubos, del ejemplo anterior, la forma en que se proporcionara cada uno sera:

0,1;

Se observa que se expresa un cubo por vez.

Todos los cubos que se van combinado son marcados con el caracter 'C' y aquellos que no pueden serlo con '\*'.

Este proceso se repite hasta encontrar todos los cubos(n+1) posibles para cada funcion.

Terminado el analisis de las funciones e intersecciones, se presentan al usuario los implicantes que se acaban de generar y los que se van acumulando de los anteriores.

Cuando se han obtenido todos los implicantes de todas las funciones, el programa arma para el usuario la tabla correspondiente con ellos. Ejemplo:

De una funcion dada por  $F(C,B,A) = S(0,1,3,4,5)$ , su tabla de implicantes es:

	0	1	4	6	7
a				@	@
b			@	@	
c	@		@		
d	@	@			

Donde:

- a = {6,7}
- b = {4,6}
- c = {0,4}
- d = {0,1}

pidiendole llenar siempre el ultimo renglon.

El usuario se posicionara con el cursor en cada columna y la marcara con los caracteres '@' o 'B' segun sea el caso. El caracter 'b' indica que esa columna se mantiene en blanco.

Verificada la respuesta, se piden los implicantes esenciales los cuales se daran con las letras que los identifican separados por un espacio. Del ejemplo anterior, se daria:

a d

Si existen implicantes primos secundarios, se armara la tabla correspondiente, mostrandola al usuario, procediendo a presentarle los pasos que se realizan para resolverla a traves del metodo de Petrick (Anexo A).

Por ultimo se daran a conocer los implicantes esenciales y secundarios que forman la solucion de cada funcion, asi como la expresion eb funcion de las variables A,B,C,.. donde el peso de las variables es A<B<C.

Si la variable se acompaña de "1", se considerara su valor cierto, si se acompaña de "0", se considerara su valor nesado.

Con esto concluye este programa.

## VII. SECUENCIALES

### VII.1 ACCIONES

Este programa tiene como función guiar al usuario en la reducción de una tabla de estados secuenciales e implementarla con Flip Flop; su desarrollo se puede dividir en dos aspectos:

- Reducción de una tabla de estados secuenciales (con uso de la técnica de PARTICION).
- Con la tabla reducida implementar el circuito con los Flip Flop JK, SR, D o T usando Mapas de Karnaugh para construir la lógica combinacional correspondiente.

El Diagrama Jerárquico de este programa se muestra en la Fig. VII.1.

#### PROGRAMA PRINCIPAL

Inicialmente este programa principal da la oportunidad al usuario de conocer una introducción acerca de la forma en que se desarrollara el método en el programa.

Otra de sus funciones es acotar la tabla de estados pidiendo el número de estados y entradas.

El función del número de estados pide el nombre con que se identificara cada uno de ellos. Posteriormente pide el nombre de los estados futuros y salidas correspondientes para cada entrada de cada estado presente, para lo cual se auxilia de la subrutina EDOSFUTUROS.

La tabla construida se presenta al usuario, dándole oportunidad de hacer cambios en ella si así lo requiere.

Con la tabla definitiva procede a reducir los estados indistinguibles.

Con esta información pregunta al usuario si hay estados que se puedan reducir, si no los hay, da oportunidad a plantear otro

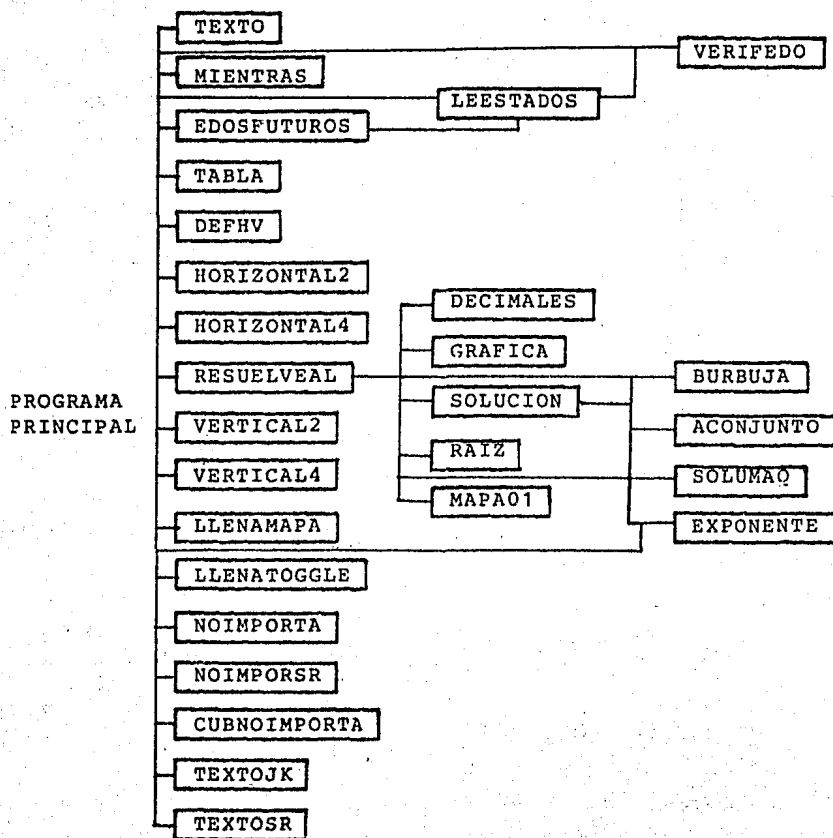


FIG. VII.1 DIAGRAMA JERARQUICO

problema en el que si existen estos, o bien, concluir el proceso.

Si existen estados que se deban reducir, los pedira por parejas y validara; para realizar lo anterior, se auxilia de la subrutina VERIFEDO.

Al finalizar esta fase presentara nuevamente la tabla presentada anteriormente pero ahora reducida a manera de informacion para el usuario.

Con el nuevo numero de estados y el numero original de entradas, acota el Mapa de Karnaugh necesario para hacer el analisis de implementacion.

A continuacion presenta los posibles Flip Flop con los que puede implementar su circuito para que elija alguno de ellos.

Los algoritmos usados en la modificacion del Mapa de Karnaugh para el analisis de JK y SR son muy practicos ya que no usa sus tablas de transicion, aunque se deriva de ellas como a continuacion se justifica:

1) Para el Flip Flop JK el programa plantea al usuario que:

Q(t)		Q(t+1)	J	K
0	->	0	0	*
0	->	1	1	*
1	->	0	*	1
1	->	1	*	0

Tabla de Transicion

$$Q(t+1) = Q(t)J + Q(t)K \quad \text{ec. caracteristica}$$

Interpretacion de la tabla de transicion para efectos de simplificacion.

1.1) Para el analisis de J(X) se hacen '\*' (no importa) todos los estados donde la variable es cierta ('1') y se analiza el mapa con los '1' restantes en el; lo cual se deduce del analisis siguiente:

Para el dise&o de J se tienen las siguientes caracteristicas:

Donde Q(t) es verdadera, J = \*

Y de la ecuacion se tiene que J es verdadera por lo que se diseña con '1'.

1.2) Para el analisis de K(X) se hacen '\*' (no importa) todos los estados donde la variable es negada ('0') y se analiza el mapa con los '0' restantes en el; lo cual se deduce del analisis siguiente: De la tabla se observa que la ecuacion característica es: Para el diseño de K se tienen las siguientes características:

Donde Q(t) es negada,  $K = *$

De la ecuacion se tiene que K es negada por lo tanto se diseña con '0'.

2) Para el Flip Flop SR el programa plantea al usuario que:

Q(t)	:	Q(t+1)	:	S	:	R
0	->	0	:	0	:	*
0	->	1	:	1	:	0
1	->	0	:	0	:	1
1	->	1	:	*	:	0

Tabla de Transicion

$$Q(t+1) = S(t) + Q(t)R \quad \text{ec. característica}$$

Interpretacion de la tabla de transicion para efectos de simplificacion

2.1) Para el analisis de S(X) se hacen '\*' (no importa) todos los minterminos de la funcion donde la variable es cierta ('1') y se analiza el mapa con los '1' restantes en el; lo cual se deduce del analisis siguiente: Donde Q(t) es Q(t+1) son verdaderas,  $S = *$

y de la ecuacion, cuando Q(t+1) es verdadero, S es verdadera por lo que se diseña con '1'.

2.2) Para el analisis de R(X) se hacen '\*' (no importa) todos los maxterminos de la funcion donde la variable es negada ('0') y se analiza el mapa con los '0' restantes en el; lo cual se deduce del analisis siguiente:

Donde Q(t) y Q(t+1) son falsas,  $R = *$

y de la ecuacion, cuando Q(t+1) es verdadero, R es falsa, por lo que se diseña con '0'.

## SUBROUTINAS

En la siguiente descripción de las subrutinas usadas por este programa, serán obviadas aquellas que fueron descritas en los capítulos IV, V y VI.

### - TEXTO (Cap. IV)

### - TEXTOJK

Contiene una breve descripción acerca del algoritmo usado por el programa para implementar el circuito con Flip Flop JK.

### - TEXTOSR

Contiene una breve descripción acerca del algoritmo usado por el programa para implementar el circuito con Flip Flop SR.

### - TABLA

Grafica la tabla de estados.

### - VERIFEDO

Variables de entrada

ESTADO: caracter vectorial <estado que se verifica>

CONT1: entero escalar <renslon en la tabla del estado verificado>

CONT2: entero escalar <entrada del estado que se verifica>

Variables de salida

BANDERA: booleano <indica si el estado se encontro o no>

LUGAR: entero escalar <renslon en la tabla donde fue encontrado el estado>

Verifica si existe el estado enviado a través de ESTADO en la tabla de estados, si es así, regresa el renslon donde se encontro y el valor de BANDERA es cierto.

### - MIENTRAS

Esta subrutina verifica si la respuesta del usuario es SI o NO.



## - LEESTADOS

## Variables de entrada

EDO: entero escalar &lt;renslon de la tabla del estado&gt;

ENTRA: entero escalar &lt;entrada del estado analizado&gt;

## Variables de salida

ESTADO: caracter vectorial &lt;nombre del estado enviado&gt;

LUGAR: entero escalar &lt;regresa el renslon de la tabla de estados donde se ubica ESTADO&gt;

Lee estados y manda a verificar si existen en la tabla de estados o no.

## - EDOSFUTUROS

Indica al usuario el estado futuro que debe proporcionar y lo verifica ; tambien pide y verifica la salida correspondiente.

## - EXPONENTE (Cap. IV)

## - LLENAMAPA

## Variables de entrada

BINA: caracter escalar &lt;valor binario (0 o 1)&gt;

CONT: entero escalar &lt;posicion de la representacion binaria que se analiza&gt;

Obtiene como minterminos aquellos estados donde la posicion de la representacion binaria (variable) analizada, sea igual a BINA. Esta subrutina es usada cuando se implementa el circuito con Flip Flop JK o SR.

## - LLENATOGGLE

Obtiene como minterminos aquellos en que exista un cambio (0 a 1) o (1 a 0) del estado presente al estado futuro en la representacion binaria (variable) que se analiza. Esta subrutina se usa cuando se implementa el circuito con Flip Flop Toggle.

## - GRAFICA (Cap. IV)

## - NOIMPORTA

Variables de entrada

BINA: caracter escalar &lt;valor binario (0 o 1)&gt;

CONT: entero escalar &lt;posicion de la representacion binaria que se analiza&gt;

Hace no importa '\*' todos los estados donde la representacion binaria (variable) analizada es igual a BINA.

## - NOIMPORSR

Variables de entrada

BINA: caracter escalar &lt;valor binario (0 o 1)&gt;

CONT: entero escalar &lt;posicion de la representacion binaria que se analiza&gt;

Hace no importa '\*' los estados del mapa de Karnaugh cuya posicion de la representacion binaria (variable) analizada y estado del mapa sean iguales a BINA.

## - CUBNOIMPORTA

Variable de entrada

CONT: entero escalar &lt;posicion de la representacion binaria que se analiza&gt;

Incluye como minterminos los estados no importa para obtener soluciones optimas (manejo interno del programa).

- DECIMALES (Cap. IV)

- HORIZONTAL2 (Cap. IV)

- HORIZONTAL4 (Cap. IV)

- VERTICAL2 (Cap. IV)

- VERTICAL4 (Cap. IV)

- RAIZ (Cap. IV)

- DEFHU (Cap. IV)

- ACONJUNTO (Cap. IV)
- SECUNDARIOS (Cap. IV)
- SOLUNAQ (Cap. IV)
- BURBUJA (Cap. IV)
- SOLUCION (Cap. IV)
- MAPABINARIO (Cap. IV)
- MAPAACTUAL (Cap. IV)
- MAPAS01 (Cap. IV)
- MAPAORIG

Inicializa el mapa de Karnaugh para poder llenarlo con los minterminos de la funcion que se vaya a analizar. En caso de que no se tengan definidos todos los estados, completa el mapa con no importa '\*'.

- RESUELVEAL

Variable de entrada

CONTA: entero escalar <indicador de la variable del mapa que se esta analizando>

En esta subrutina se lleva a cabo la minimizacion de las funciones generadas para implementar el circuito secuencial, usando el metodo de Mapas de Karnaugh; es de hecho el programa principal del subsistema descrito en el Capitulo IV y se auxilia de las subrutinas del mismo, para relacionarse con su funcion, habra que referirse totalmente al capitulo citado.

SECUENCIALES

FILE (VR83)SECUENCIALES ON UNAM2

100 PROGRAM SECUENCIALES;

200 (% ESTE PROGRAMA SIRVE PARA MINIMIZAR TABLAS DE ESTADOS SECUENCIALES  
 300 Y OBTENER LA IMPLEMENTACION DEL CIRCUITO EN FLIP-FLOPS "JK" O "SR"  
 400 O "D" O "T", PARA LO CUAL SE AUXILIA DE LAS SIGUIENTES SUBROUTINAS:

- 500 - TEXTO: CONTIENE UNA BREVE EXPLICACION ACERCA DE COMO SE MINIMI  
 600 ZA UNA TABLA DE ESTADOS SECUENCIALES POR EL METODO DE  
 700 PARTICION. E INDICA LA FORMA DE LLENAR UN MAPA DE  
 800 KARNAUGH CON ESOS ESTADOS, YA QUE ES ESTE EL MEDIO DEL  
 900 QUE SE UTILIZA EN ESTE PROGRAMA PARA MINIMIZAR FUNCIONES.
- 1000 - TEXTOSR: CONTIENE UNA INTRODUCCION ACERCA DE COMO MANEJAR EL  
 1100 MAPA GENERAL PARA RESOLVER EL CIRCUITO CON SR.
- 1200 - TEXTOSJK: CONTIENE UNA INTRODUCCION ACERCA DE COMO MANEJAR EL  
 1300 MAPA GENERAL PARA RESOLVER EL CIRCUITO CON JK.
- 1400 - TABLA: GRAFICA LA TABLA DE ESTADOS SECUENCIALES.
- 1500 - VERIFEDO: VERIFICA LA EXISTENCIA DE UN ESTADO.
- 1600 - MIENTRAS: VERIFICA DECISIONES DEL USUARIO (SI O NO).
- 1700 - LEESTADOS: LEE ESTADOS, LOS CUALES MANDA VERIFICAR CON VERIFEDO.
- 1800 - EDOSFUTUROS: LEE LOS ESTADOS FUTUROS, VERIFICA QUE CORRESPON  
 1900 DAN A ALGUN ESTADO PRESENTE, LEE LAS SALIDAS DE  
 2000 CADA UNO DE ELLOS Y SI ESTAN EN EL RANGO INICIAL  
 2100 MENTE PROPUESTO POR EL USUARIO.
- 2200 - EXPONENTE: OBTIENE LA POTENCIA DE UN NUMERO EN BASE 2.
- 2300 - LLENAMAPA: OBTIENE LOS MINTERMINOS DE LA FUNCION QUE ORIGINA  
 2400 EL ANALISIS DE CIERTA POSICION DE LA REPRESENTACION  
 2500 BINARIA DE CADA ESTADO FUTURO PARA TODAS  
 2600 LAS ENTRADAS PROPUESTAS Y LLENA EL MAPA DE K.  
 2700 CORRESPONDIENTE (USADO CON JK, SR Y D).
- 2800 - GRAFICA: GRAFICA 1 O 2 MAPAS DE KARNAUGH, UNO DE ELLOS CON  
 2900 TODAS LAS POSICIONES POSIBLES PARA N ENTRADAS, EL  
 3000 SEGUNDO INDICA LOS MINTERMINOS DE LA FUNCION ANA  
 3100 LIZADA.
- 3200 - NOIMPORTA: SUSTITUYE POR \* (NO IMPORTA) LAS POSICIONES  
 3300 DONDE LA VARIABLE QUE SE ESTA ANALIZANDO ES (1)  
 3400 CIERTA O (0) FALSA, DEPENDIENDO SI SE RESUELVE  
 3500 CON 1 O 0. EL MAPA.
- 3600 - NOIMPORSR: SUSTITUYE POR \* (NO IMPORTA) LOS MINTERMINOS  
 3700 DONDE LA VARIABLE QUE SE ESTA ANALIZANDO ES (1)  
 3800 CIERTA O (0) FALSA, DEPENDIENDO SI SE RESUELVE  
 3900 CON 1 O 0 EL MAPA.
- 4000 - CUBNOIMPORTA: ASUME TODOS LOS ESTADOS \* COMO 1 O 0 DEPEN  
 4100 DIENDO SI SE RESUELVE CON 1 O 0 EL MAPA.
- 4200 - DECIMALES: LEE Y VERIFICA LOS MINTERMINOS QUE PROPONE EL  
 4300 USUARIO PARA MINIMIZARSE.
- 4400 - HORIZONTAL2,HORIZONTAL4: INTERCAMBIA RENGLONES DE UN MAPA  
 4500 QUE ORIGINALMENTE SE ENCUENTRA EN FORMA SECUEN  
 4600 CIAL.
- 4700 - VERTICAL2,VERTICAL4: INTERCAMBIA COLUMNAS DE UN MAPA QUE  
 4800 ORIGINALMENTE SE ENCUENTRA EN FORMA SECUENCIAL.
- 4900 - RAIZ: OBTIENE LA RAIZ N(ESIMA) DE UN NUMERO.
- 5000 - DEFHV: DEFINE EL NUMERO DE RENGLONES Y COLUMNAS QUE TIENE  
 5100 EL MAPA DE KARNAUGH PARA EL NUMERO DE ESTADOS Y  
 5200 ENTRADAS.
- 5300 - ACONJUNTO: SI HAY IMPLICANTES SECUNDARIOS, ESTA SUBROUTINA  
 5400 ES LLAMADA Y ENCUENTRA LOS QUE DAN LA SOLUCION  
 5500 OPTIMA.
- 5600 - SECUNDARIOS: DEFINE LA TABLA DE IMPLICANTES SECUNDARIOS.
- 5700 - SOLUMAR: OBTIENE LA(S) SOLUCION(ES) FINAL(ES) CON LAS  
 5800 VARIABLES QUE DEFINE EL PROGRAMA.
- 5900
- 5901
- 5902

```

5903
5904
6000 - BURBUJA: ORDENA LOS MINTERMINOS POR EL METODO DE BURBUJA.
6100 - SOLUCION: SUBROUTINA ESENCIAL DEL PROGRAMA YA QUE ES EN
6200   ESTA DONDE SE RESUELVE LA FUNCION A TRAVES DEL
6300   METODO DE QUINE MC.CLUSKEY,
6400 - MAPABINARIO: OBTIENE EL VALOR BINARIO DE TODAS LAS POSI
6500   CIONES POSIBLES DEL MAPA DE K.
6600 - MAPACTUAL: OBTIENE UN MAPA ORDENANDO LOS MINTERMINOS
6700   SECUENCIALMENTE, EL CUAL POSTERIORMENTE SERA
6800   ORDENADO DE LA FORMA APROPIADA PARA MINIMIZAR
6900   LOS MINTERMINOS INTERCAMBIANDO RENGLONES Y
7000   COLUMNAS.
7100 - MAPA01: DEFINE EL MAPA UBICANDO LOS MINTERMINOS DE LA
7200   FUNCION Y MARCA AQUELLOS QUE YA FUERON MINIMIZADOS
7300   POR EL USUARIO DURANTE SU PROCESO DE REDUCCION.
7400 - MAPADRIG: LIMPIA Y LLENA EL MAPA QUE ORIGINALMENTE SE CREA
7500   ANALIZANDO LA TABLA DE ESTADOS.
7600 - RESUELVEAL: LEE LOS MINTERMINOS QUE PROPONE EL USUARIO
7700   PARA REDUCIRSE, VERIFICA Y DA LA SOLUCION DE
7800   EL MAPA QUE EN ESE MOMENTO SE ESTA ANALIZANDO.
7900   ESTE PROGRAMA RESTRINGE EL NUMERO DE ENTRADAS Y ESTADOS DEBIDO
8000   A QUE ES UN PROGRAMA DE TIPO INTERACTIVO Y EDUCATIVO Y EL ESPA
8100   CIO DE LA PANTALLA NO PERMITE AMPLIAR ESTAS RESTRICCIONES. *)
8200 LABEL 1;
8300 TYPE
8400   NOMENTRADAS = PACKED ARRAY(0..65,0..7) OF CHAR;
8500   NOMENFUT = PACKED ARRAY(0..7,0..20,0..10) OF CHAR;
8600   SALIDAS = PACKED ARRAY(0..7,0..20) OF INTEGER;
8700   MAPA = ARRAY(0..8,0..8) OF INTEGER;
8800   CHAPA = ARRAY(0..8,0..8) OF CHAR;
8900   CARAC = ARRAY(0..32) OF CHAR;
9000   MATRICES = ARRAY(0..70,0..65) OF INTEGER;
9100   MATRICESC = ARRAY(0..70,0..64) OF CHAR;
9200   VECTOR = ARRAY(-1..100) OF INTEGER;
9300   MATENT = ARRAY(0..30,0..30) OF INTEGER;
9400   MATCARAC = ARRAY(0..64,0..79) OF CHAR;
9500   NUMEROS = SET OF INTEGER;
9600   STRING = ARRAY(0..6) OF CHAR;
9700   LETRAS = RECORD
9800     LETRA : STRING; (* VARIABLES DEL MAPA *)
9900     NOASOC : VECTOR; (* POTENCIAS DE 2 ASOCIADAS CON
10000     LA POSICION DE LAS VARIABLES *)
10100   BINARIO : ARRAY(0..8,0..8,0..5) OF INTEGER;
10200   END;
10300   SOLUC = RECORD
10400     ARREGLO : MATRICES;
10500     ARRMARCA : MATRICESC;
10600     NOCUBIERTO : MATCARAC;
10700     HATSOL,IMPSEC,TABIMP: MATENT;
10800     SOLTOTAL,CUBIERTO : MATCARAC;
10900     MARC : ARRAY(1..30) OF CHAR;
11000     NELEM,NNATSOL,NSOLTOTAL,NTARSEC,NTABIMP : VECTOR;
11100   END;
11200 VAR
11300   BANDJS,BANDEX,BANDERA,BANDERA1,BANDERA2 : BOOLEAN;
11400   EPRESENTE,EPRESBIN : NOMENTRADAS; (* ESTADOS PRESENTES, NOMBRE Y
11500   REPRESENTACION BINARIA *)
11600   EFUTURO,EFUT,EFUTBIN, (* ESTADOS FUTUROS, NOMBRE Y REPRESENTACION
11700   BINARIA *)
11800   EDOAUX,ESTADO (* ARREGLOS AUXILIARES PARA LECTURA DE ESTADOS *)
11900   : NOMENFUT;
12000   SALIDA : SALIDAS; (* ARREGLO PARA GUARDAR LAS SALIDAS PARA CADA
12100   ESTADO FUTURO DE TODAS LAS ENTRADAS *)
12101
12102

```

12103  
12104  
12200 MARCAALUM, MARCA : ARRAY[0..20] OF NUMERDS;  
12300 NUMERO : PACKED ARRAY[0..1] OF INTEGER;  
12400 CONT, CONT1, CONT2, CONT3, CONT4, CONT5, CONT6, I, J, II, K, (\*CONTADORES \*)  
12500 NI, NJ, (\* DIMENSIONAMIENTO DEL MAPA DE KARNAUGH \*)  
12600 NVARIA, (\* NUMERO DE VARIABLES DEL MAPA DE KARNAUGH \*)  
12700 NESTADOS, (\* NUMERO DE ESTADOS DE LA TABLA DE SECUENCIALES \*)  
12800 NENTRADAS, (\* NUMERO DE ENTRADAS DE LA TABLA DE SECUENCIALES \*)  
12900 NENTRA, (\* NUMERO DE VARIABLES NECESARIAS PARA CUBRIR  
13000 NENTRADAS (POTENCIA DE 2) \*)  
13100 LUGAR,  
13200 NDIGITOS (\* TOTAL DE MINTERMINOS \*),  
13300 NIMPSEC (\* TOTAL DE IMPLICANTES SECUNDARIOS \*),  
13400 NIMPLIC (\* TOTAL DE IMPLICANTES PRIMOS \*),  
13500 NDIGSEC (\* NUMERO DE MINTERMINOS NO CUBIERTOS EN LA TABLA DE  
13600 IMPLICANTES PRIMOS \*),  
13700 NCASILLAS (\* NUMERO DE MINTERMINOS PARA SOLUCIONES PARCIALES \*),  
13800 NCASI (\* CONTIENE EL NUMERO DE MINTERM. MINIMOS DE C/FUNC. \*),  
13900 TOTSOL (\* TOTAL DE SOLUCIONES FINALES \*),  
14000 POTENCIA (\* NUMERO MAXIMO DE MINTERMINOS PARA NVARIA \*),  
14100 T1 (\* NUMERO DE IS DE LOS MINTERMINOS \*),  
14200 T (\* DIRECCIONAMIENTO DE RENGLONES \*),  
14300 NSECUN,  
14400 DECIMAL (\* MINTERMINO EN FORMA DECIMAL \*)  
14500 : INTEGER;  
14600 MAPAS (\* POSICIONES DE TODOS LOS MINTERMINOS POSIBLES  
14700 PARA EL NUMERO DE VARIABLES EN EL MAPA \*)  
14800 : MAPA;  
14900 MAPASMIN (\* MAPA CON IS EN LOS MINTERMINOS DE LA FUNCION \*),  
15000 CMAPASHIN (\* MARCA EN EL MAPA DE IS (MAPASHIN), LOS MINTER-  
15100 MINOS QUE VA CUBRIENDO EL USUARIO \*)  
15200 : CMAPA;  
15300 ARRVAR1, ARRVAR2 (\* POSICION DE LAS VARIABLES POR PESO \*),  
15400 CAMBIO, METODO, OPCION (\* LEE DECISIONES DEL USUARIO DE SI O NO \*)  
15500 : STRING;  
15600 SOLUCIONES : MATCARAC; (\* TODAS LAS SOLUCIONES DE LA  
15700 MAQUINA \*)  
15800 TABALUM : MATENT;  
15900 SIMBOLO : LETRAS;  
16000 SOLUCI : SOLUC;  
16100 NSOLUCIONES (\* NUMERO TOTAL DE SOLUCIONES \*),  
16200 NCONJEXTEN (\* NUMERO DE POSIBLES SOLUCIONES PARA IMPL. SEC \*),  
16300 NCONJUNTO,  
16400 PEURD (\* CUBOS ENVIADOS A ORDENARSE \*),  
16500 DIGITOS (\* MINTERMINOS \*),  
16600 DIGSEC (\* MINTERMINOS DE LOS IMPLICANTES SECUNDARIOS \*),  
16700 NTABALUM (\* NUMERO DE REDUCCIONES DEL ALUMNO \*),  
16800 CASI (\* CONTIENE LOS MINTERMINOS MINIMOS DE CADA FUNC. \*),  
16900 CASILLAS (\* CUBOS MANEJADOS POR EL ALUMNO \*)  
17000 : VECTOR;  
17100 CASDIB,  
17200 CDIGITOS (\* REFLEJO DE DIGITOS DONDE SE MARCAN LOS MINTERMINOS  
17300 QUE EL USUARIO VA CUBRIENDO \*) : CARAC;  
17400 UNO (\* MINTERMINOS CON 1 UNO EN SU REPRESENTACION BINARIA \*),  
17500 DOS (\* MINTERMINOS CON 2 UNOS EN SU REPRESENTACION BINARIA \*),  
17600 TRES (\* MINTERMINOS CON 3 UNOS EN SU REPRESENTACION BINARIA \*),  
17700 CUATRO (\* MINTERMINOS CON 4 UNOS EN SU REPRESENTACION BINARIA \*),  
17800 CINCO (\* MINTERMINOS CON 5 UNOS EN SU REPRESENTACION BINARIA \*),  
17900 SEIS (\* MINTERMINOS CON 6 UNOS EN SU REPRESENTACION BINARIA \*),  
18000 CONJ1, CONJ2 (\* CONJUNTOS AUXILIARES PARA VERIFICACION DE INTERSEC  
18100 CIONES O IGUALDAD DE CUBOS \*)  
18200 : NUMEROS;  
18300 CONJUNTO (\* ARREGLO DE CONJUNTOS DE CUBOS \*)  
18301  
18302

```

18303
18304
18400                : ARRAY [0..50,0..150] OF NUMEROS;
18500      SOLALUM : CARAC;
18600      MARCAMIN: ARRAY[0..30] OF CHAR;
18700  PROCEDURE TEXTO;
18800  BEGIN
18900  WRITELN('LA MINIMIZACION QUE PLANTEA ES LA MAS SENCILLA (METODO ',
19000  'DE PARTICION).');
19100  WRITELN('UN ESTADO ES ABSORVIDO POR OTRO SI TODOS SUS ESTADOS ',
19200  'FUTUROS Y SUS SALIDAS');
19300  WRITELN('CORRESPONDIENTES SON IGUALES. ');
19400  WRITELN('PARA UNA MINIMIZACION MAS FORNAL, EXISTEN OTROS ',
19500  'METODOS, UNO DE ELLOS ');
19600  WRITELN('CONOCIDO COMO METODO DE TABLA DE IMPLICACION, EL CUAL ',
19700  'PLANTEA UNA SERIE DE');
19800  WRITELN('TABLAS Y BUSQUEDAS QUE ES PREFERIBLE DESARROLLES EN ',
19900  'PAPEL DAÑO LO LABORIOSO');
20000  WRITELN('QUE RESULTA. ');
20100  WRITELN('PARA CONSULTAR ESTE METODO PUEDES CONSULTAR LOS ',
20200  'SIGUIENTES TEXTOS:');
20300  WRITELN('      1) TEORIA DE CONMUTACION Y DISEÑO LOGICO');
20400  WRITELN('      FREDERICK J. HILL');
20500  WRITELN('      2) LOGICA DIGITAL Y DISEÑO DE COMPUTADORES');
20600  WRITELN('      M. MORRIS MAND');
20700  WRITELN('PARA OBTENER LA SOLUCION EN FUNCION DE CUALQUIER FLIP ',
20800  'FLOP (JK,SR,D,T) SE');
20900  WRITELN('REQUIERE');
21000  WRITELN('LA REPRESENTACION BINARIA DE LOS ESTADOS PRESENTES ',
21100  'REDUCIDOS Y LOS FUTUROS');
21200  WRITELN('PARA TODAS LAS SALIDAS. ');
21300  WRITELN('SE IRAN OBTENIENDO LOS MAPAS DE KARNAUGH PARA CADA ',
21400  'UNA DE LAS <POSICIONES>');
21500  WRITELN('DE LA REPRESENTACION BINARIA. ');
21600  WRITELN('EJEMPLO:');
21700  WRITELN('CONTINUA...'); READLN;
21800  WRITELN('      EPRESENTE      EFUT E=0      EFUT E=1');
21900  WRITELN('      ML              ML              ML');
22000  WRITELN('      A 00            A 00            B 01');
22100  WRITELN('      B 01            A 00            C 10');
22200  WRITELN('      C 10            B 01            B 01');
22300  WRITELN('      DONDE:');
22400  WRITELN('      M,L: <POSICIONES BINARIAS DE LOS ESTADOS>');
22500  WRITELN('      A,B,C: NOMBRES ASIGNADOS A LOS ESTADOS');
22600  WRITELN;
22700  WRITELN('REACOMODANDO');
22800  WRITELN('      Q(T)          Q(T+1)');
22900  WRITELN('      E  M  L      M  L');
23000  WRITELN('      0  0  0  0    0  0');
23100  WRITELN('      1  0  0  1    0  0');
23200  WRITELN('      2  0  1  0    0  1');
23300  WRITELN('      3  0  1  1    *  *');
23400  WRITELN('      4  1  0  0    0  1');
23500  WRITELN('      5  1  0  1    1  0');
23600  WRITELN('      6  1  1  0    0  1');
23700  WRITELN('      7  1  1  1    *  *');
23800  WRITELN('CONTINUA...'); READLN;
23900  WRITELN('LLENANDO EL MAPA PARA L(T+1)');
24000  WRITELN;
24100  WRITELN('      E\ML 00 01 11 10');
24200  WRITELN('      -----');
24300  WRITELN('      | 0 | 0 | * | 1 |');
24400  WRITELN('      -----');
24500  WRITELN('      | 1 | 0 | * | 1 |');
24501
24502

```

```

24503
24504
24600 WRITELN('-----');
24700 WRITELN('SERA NUESTRA HERRAMIENTA A LA CUAL APLICAREMOS ',
24800 'REGLAS ESPECIALES PARA');
24900 WRITELN('IMPLEMENTAR EL CIRCUITO DEPENDIENDO DEL FLIP FLOP QUE ',
25000 'USEMOS.');
```

```

25100 END; (* TEXTO *)
25200
25300 PROCEDURE TEXTOJK;
25400 BEGIN
25500 WRITELN('PARA IMPLEMENTAR UN CIRCUITO CON FLIP FLOP JK, YA CON ',
25600 'EL MAPA DEFINIDO ');
25700 WRITELN('PARA LOS ESTADOS FUTUROS DE CADA FUNCION, USAREMOS UN',
25800 'METODO MUY ');
25900 WRITELN('SENCILLO:');
26000 WRITELN('PARA J:');
26100 WRITELN('SE HACEN NO IMPORTA TODAS LAS POSICIONES DONDE LA <VA',
26200 'RIABLE> ANALIZADA');
26300 WRITELN('ES 1 Y SE RESUELVE EL MAPA CON LOS 1 RESTANTES.');
```

```

26400 WRITELN('PARA K:');
26500 WRITELN('SE HACEN NO IMPORTA TODAS LAS POSICIONES DONDE LA <VA',
26600 'RIABLE> ANALIZADA');
26700 WRITELN('Y SE RESUELVE EL MAPA CON LOS 0 RESTANTES.');
```

```

26800 WRITELN('SIGUIENDO ESTAS SENCILLAS REGLAS, RESUELVE TU CIRCUITO');
26900 WRITELN('CONTINUA...'); READLN;
27000 END; (* TEXTOJK *)
27100
27200
27300 PROCEDURE TEXTOSR;
27400 BEGIN
27500 WRITELN('PARA IMPLEMENTAR UN CIRCUITO CON FLIP FLOP SR, YA CON ',
27600 'EL MAPA DEFINIDO ');
27700 WRITELN('PARA LOS ESTADOS FUTUROS DE CADA FUNCION, USAREMOS UN',
27800 'METODO MUY ');
27900 WRITELN('SENCILLO:');
28000 WRITELN('PARA S:');
28100 WRITELN('SE HACEN NO IMPORTA TODOS LOS 1 DONDE LA VARIABLE ANALI',
28200 'ZADA ES CIERTA (1)');
28300 WRITELN('Y SE RESUELVE EL MAPA CON LOS 1 RESTANTES.');
```

```

28400 WRITELN('PARA R:');
28500 WRITELN('SE HACEN NO IMPORTA TODOS LOS 0 DONDE LA VARIABLE ANALI',
28600 'ZADA ES FALSA (0)');
28700 WRITELN('Y SE RESUELVE EL MAPA CON LOS 0 RESTANTES.');
```

```

28800 WRITELN('SIGUIENDO ESTAS SENCILLAS REGLAS, RESUELVE TU CIRCUITO');
28900 WRITELN('CONTINUA...'); READLN;
29000 END; (* TEXTOSR *)
29100
29200 PROCEDURE TABLA;
29300 BEGIN
29400 (* FOR CONT:=1 TO 20 DO
29500 WRITELN(CHR(10)); *)
29600 WRITELN(' SIGNIFICADO DE VARIABLES:');
29700 WRITELN(' QP : ESTADO PRESENTE');
29800 WRITELN(' QF : ESTADO FUTURO PARA EL NUMERO DE ENTRADAS');
29900 WRITELN(' E : ENTRADA');
30000 WRITELN(' S : SALIDA');
30100 WRITELN;
30200 WRITE('!----');
30300 FOR CONT:=0 TO NENTRADAS DO
30400 WRITE('!-----');
30500 WRITELN('');
30600 WRITE('! ');
30700 FOR CONT:=0 TO NENTRADAS DO
30701
30702
```



```

30703
30704
30800     WRITE('E=';6,CONT;1, ' ');
30900     WRITELN('');
31000     WRITE('!-----');
31100     FOR CONT:=0 TO NENTRADAS DO
31200         WRITE('!-----');
31300         WRITELN('');
31400     WRITE('! QP :;5);
31500     FOR CONT:=0 TO NENTRADAS DO
31600         WRITE('! QP : S ');
31700         WRITELN('');
31800     WRITE('!-----');
31900     FOR CONT:=0 TO NENTRADAS DO
32000         WRITE('!-----');
32100         WRITELN('');
32200     FOR CONT:=1 TO NESTADOS DO
32300     BEGIN
32400         WRITE('! ');
32500         FOR CONT1:=1 TO 3 DO
32600             WRITE(EPRESENTE[CONT,CONT1];1);
32700         FOR CONT2:=0 TO NENTRADAS DO
32800         BEGIN
32900             WRITE('! ');
33000             FOR CONT1:=1 TO 3 DO
33100                 WRITE(EFUTURO[CONT2,CONT,CONT1];1);
33200             WRITE('! ',SALIDA[CONT2,CONT];1, ' ');
33300         END;
33400         WRITELN('!');
33500     END;
33600     WRITE('!-----');
33700     FOR CONT:=0 TO NENTRADAS DO
33800         WRITE('!-----');
33900         WRITELN('!');
34000 END;(* TABLA *)
34100
34200 PROCEDURE VERIFEDO(ESTADO;NOMENFUT; CONT2,CONT1;INTEGER;
34300     VAR LUGAR; INTEGER; VAR BANDERA;BOOLEAN);
34400 VAR
34500     CONT3,CONT4; INTEGER;
34600 BEGIN
34700     FOR CONT3:=1 TO NESTADOS DO
34800     BEGIN
34900         BANDERA:=TRUE;
35000         FOR CONT4:=1 TO 3 DO
35100             IF EPRESENTE[CONT3,CONT4] <> ESTADO[CONT2,CONT1,CONT4] THEN
35200                 BANDERA:=FALSE;
35300         IF BANDERA THEN
35400         BEGIN
35500             LUGAR:=CONT3;
35600             CONT3:=NESTADOS;
35700         END;
35800     END;
35900 END;(* VERIFEDO *)
36000
36100 PROCEDURE MIENTRAS;
36200 BEGIN
36300     WHILE NOT((OPCION[1] = 'S') AND (OPCION[2] = 'I') OR
36400         (OPCION[1] = 'N') AND (OPCION[2] = 'O')) DO
36500     BEGIN
36600         WRITELN('RESPONDE (SI) O (NO) SEGUN SEA EL CASO');
36700         READLN;
36800         READ(OPCION[1],OPCION[2]);
36900     END;
36901
36902

```

```

36903
36904
37000 END; (* MIENTRAS *)
37100
37200 PROCEDURE LEESTADOS(VAR ESTADO:NOMENFUT;ENTRA,EDO:INTEGER;
37300 VAR LUGAR:INTEGER);
37400 BEGIN
37500 BANDERA:=FALSE;
37600 WHILE NOT BANDERA DO
37700 BEGIN
37800 READLN;
37900 CONT:=1;
38000 LUGAR:=1;
38100 WHILE (NOT EOLN) AND (CONT < 4) DO
38200 BEGIN
38300 READ(ESTADOENTRA,EDO,CONT);
38400 CONT:=CONT+1;
38500 END;
38600 VERIFEDO(ESTADO,ENTRA,EDO,LUGAR,BANDERA);
38700 IF NOT BANDERA THEN
38800 BEGIN
38900 FOR CONT:=1 TO 3 DO
39000 ESTADOENTRA,EDO,CONT:= ' ';
39100 WRITELN('EL ESTADO NO EXISTE, VERIFICA');
39200 END;
39300 END;
39400 END; (* LEESTADOS *)
39500
39600 PROCEDURE EDOSFUTUROS;
39700 BEGIN
39800 WRITELN;
39900 WRITELN('TUS ENTRADAS ESTAN NUMERADAS DEL 0 AL ',NENTRADAS:1);
40000 FOR CONT1:=1 TO NESTADOS DO
40100 FOR CONT2:=0 TO NENTRADAS DO
40200 BEGIN
40300 WRITE('CUAL ES EL ESTADO EN T+1 DEL ESTADO ');
40400 FOR CONT:=1 TO 3 DO
40500 WRITE(EPRESENTE[CONT1,CONT]:1);
40600 WRITELN(' PARA UNA ENTRADA =',CONT2:1);
40700 LEESTADOS(EFUTURO,CONT2,CONT1,LUGAR);
40800 WRITELN('CUAL ES LA SALIDA CORRESPONDIENTE');
40900 READLN;
41000 READ(SALIDA[CONT2,CONT1]);
41100 WHILE (SALIDA[CONT2,CONT1] < 0) OR
41200 (SALIDA[CONT2,CONT1] > NENTRADAS) DO
41300 BEGIN
41400 WRITELN('LA SALIDA DEBE ESTAR ENTRE 0 Y ',NENTRADAS:1,
41500 ' VERIFICA');
41600 READLN;
41700 READ(SALIDA[CONT2,CONT1]);
41800 END;
41900 END;
42000 END; (* EDOSFUTUROS *)
42100
42200 FUNCTION EXPONENTE(N:INTEGER) : INTEGER;
42300 BEGIN
42400 EXPONENTE:=ROUND(EXP(N* LN(2)));
42500 END;
42600
42700 PROCEDURE LLENAMAPA(CONT:INTEGER; BINA:CHAR);
42800 BEGIN
42900 CONT3:=0;
43000 NDIGITOS:=-1;
43100 (* SE OBTIENEN LOS MINTERMINOS DE LA FUNCION *)
43101
43102

```

```

43103
43104
43200 FOR CONT1:=1 TO NESTADOS DO
43300 FOR CONT2:=0 TO EXPONENTE(NENTRA)-1 DO
43400 BEGIN
43500     IF BANDEX THEN BEGIN
43600         IF (EFUTBINCONT2,CONT1,NVARIA+1-CONTJ = BINA) AND (CONT2 <=
43700             NENTRADAS) THEN BEGIN
43800             NDIGITOS:=NDIGITOS+1;
43900             DIGITOS(NDIGITOS):=CONT3;
44000         END END
44100         ELSE
44200             IF SALIDACONT2,CONT1J = 1 THEN
44300                 BEGIN
44400                     NDIGITOS:=NDIGITOS+1;
44500                     DIGITOS(NDIGITOS):=CONT3;
44600                 END;
44700                 CONT3:=CONT3+1;
44800             END;
44900 END; (* LLENAMAPA *)
45000
45100 PROCEDURE LLENATOGGLE;
45200 BEGIN CONT3:=0; NDIGITOS:=-1;
45300     FOR CONT1:=1 TO NESTADOS DO
45400     FOR CONT2:=0 TO NENTRADAS DO
45500     BEGIN
45600         IF (EPRESBINCONT1,NVARIA+1-CONTJ <> EFUTBINCONT2,CONT1,NVARIA+
45700             1-CONTJ) AND (CONT2 <= NENTRADAS) THEN
45800             BEGIN
45900                 NDIGITOS:=NDIGITOS+1;
46000                 DIGITOS(NDIGITOS):=CONT3;
46100             END;
46200             CONT3:=CONT3+1;
46300         END;
46400     END; (* LLENATOGGLE *)
46500
46600 PROCEDURE GRAFICA(MAPEO:MAPA; MAPEOMIN,CMAPEOMIN:CMAPA);
46700 VAR I1:INTEGER;
46800 BEGIN
46900     IF BANDERA THEN
47000         I1:=2
47100     ELSE
47200         I1:=1;
47300     FOR J:=1 TO I1 DO
47400     BEGIN
47500         FOR I:=2 DOWNT0 0 DO
47600             WRITE(ARRVAR2(IJ));
47700             WRITE('\');
47800         FOR I:=2 DOWNT0 0 DO
47900             WRITE(ARRVAR1(IJ));
48000         FOR I:=1 TO NJ DO
48100             WRITE(' ');
48200     END;
48300     WRITELN;
48400     WRITE(' ');
48500     FOR J:=1 TO NJ DO
48600         WRITE('----');
48700     IF BANDERA THEN
48800     BEGIN
48900         WRITE(' ');
49000         FOR J:=1 TO NJ DO
49100             WRITE('----');
49200     END;
49300     WRITELN;
49301
49700

```

```

49303
49304
49400 FOR I:=0 TO NI-1 DO
49500 BEGIN
49600 WRITE('  ');
49700   FOR J:=0 TO NJ-1 DO
49800     WRITE(MAPEOCI,JJ:2,' ');
49900   IF BANDERA THEN
50000     BEGIN
50100       WRITE('  ');
50200     IF BANDERA2 THEN
50300       FOR J:=0 TO NJ-1 DO
50400         WRITE(MAPEOMINCI,JJ:2,CHAPEOMINCI,JJ:1,' ');
50500     ELSE
50600       FOR J:=0 TO NJ-1 DO
50700         WRITE(MAPEOMINI,JJ:2,' ');
50800     END;
50900     WRITELN;
51000   WRITE(' - ');
51100   FOR J:=1 TO NJ DO
51200     WRITE('----');
51300   IF BANDERA THEN
51400     BEGIN
51500       WRITE(' - ');
51600     FOR J:=1 TO NJ DO
51700       WRITE('----');
51800     END;
51900     WRITELN;
52000   END;
52100   WRITELN('CONTINUA...'); READLN;
52200 END; (* GRAFICA *)
52300
52400 PROCEDURE NOIMPORTA(CONT,BINA;INTEGER);
52500 BEGIN
52600   FOR CONT1:=0 TO NI-1 DO
52700     FOR CONT2:=0 TO NJ-1 DO
52800       IF SIMBOLO.BINARIO[CONT1,CONT2,NVARIA+NENTRA-CONT1]=BINA THEN
52900         BEGIN
53000           FOR CONT3:=0 TO NDIGITOS DO
53100             IF DIGITOS[CONT3] = MAPAS[CONT1,CONT2] THEN
53200               BEGIN
53300                 FOR CONT4:=CONT3 TO NDIGITOS-1 DO
53400                   DIGITOS[CONT4]:=DIGITOS[CONT4+1];
53500                 CONT3:=NDIGITOS;
53600                 NDIGITOS:=NDIGITOS-1;
53700               END;
53800               MAPASMIN[CONT1,CONT2]:='*';
53900             END;
54000           FOR CONT1:=0 TO NDIGITOS DO
54100             CASI[CONT1]:=DIGITOS[CONT1];
54200           NCASI:=NDIGITOS;
54300         END; (* NOIMPORTA *)
54400
54500 PROCEDURE NOIMPORSR(CONT,BINA;INTEGER);
54600 BEGIN
54700   FOR CONT1:=0 TO NI-1 DO
54800     FOR CONT2:=0 TO NJ-1 DO
54900       IF SIMBOLO.BINARIO[CONT1,CONT2,NVARIA+NENTRA-CONT1] = BINA THEN
55000         FOR CONT3:=0 TO NDIGITOS DO
55100           IF DIGITOS[CONT3] = MAPAS[CONT1,CONT2] THEN
55200             BEGIN
55300               FOR CONT4:=CONT3 TO NDIGITOS-1 DO
55400                 DIGITOS[CONT4]:=DIGITOS[CONT4+1];
55500               CONT3:=NDIGITOS;
55501
55502

```



```

61703
61704
61800     FOR I2:=J-1 DOWNT0 0 DO
61900         IF DIGSCI2J = AUX1 THEN
62000             BEGIN
62100                 WRITELN('HAY UN MINTERMINO REPETIDO,VUELVE A DARLOS');
62200                 GOTO 60;
62300             END;
62400         DIGSCJ:=AUX1;
62500         J:=J+1;
62600         WHILE MINTER = ' ' DO
62700             READ(MINTER);
62800         IF MINTER = ',' THEN
62900             READ (MINTER);
63000     END
63100 ELSE
63200     BEGIN
63300         WRITELN('HAY UN CARACTER NO VALIDO, VUELVE A DARMELOS');
63400         GOTO 60;
63500     END;
63600 END;
63700     NDIGS:=J-1;
63800 END;
63900
64000 PROCEDURE HORIZONTAL2(VAR MAPEO:MAPA);
64100 VAR AUXILIAR : ARRAY[0..8] OF INTEGER;
64200 BEGIN
64300     FOR I:=0 TO (NI-1) DO
64400         BEGIN
64500             AUXILIAR[I]:=MAPEO[I,2];
64600             MAPEO[I,2]:=MAPEO[I,3];
64700             MAPEO[I,3]:=AUXILIAR[I];
64800         END;
64900     END;
65000
65100 PROCEDURE HORIZONTAL4(VAR MAPEO:MAPA);
65200 VAR AUXILIAR : ARRAY[0..8] OF INTEGER;
65300 BEGIN
65400     FOR I:=0 TO NI-1 DO
65500         BEGIN
65600             AUXILIAR[I]:=MAPEO[I,6];
65700             MAPEO[I,6]:=MAPEO[I,7];
65800             MAPEO[I,7]:=AUXILIAR[I];
65900         END;
66000     END;
66100
66200 PROCEDURE VERTICAL2(VAR MAPEO:MAPA);
66300 VAR AUXILIAR : ARRAY[0..8] OF INTEGER;
66400 BEGIN
66500     FOR J:=0 TO NJ-1 DO
66600         BEGIN
66700             AUXILIAR[J]:=MAPEO[2,J];
66800             MAPEO[2,J]:=MAPEO[3,J];
66900             MAPEO[3,J]:=AUXILIAR[J];
67000         END;
67100     END;
67200
67300 PROCEDURE VERTICAL4(VAR MAPEO:MAPA);
67400 VAR AUXILIAR : ARRAY[0..8] OF INTEGER;
67500 BEGIN
67600     FOR J:=0 TO NJ-1 DO
67700         BEGIN
67800             AUXILIAR[J]:=MAPEO[6,J];
67900             MAPEO[6,J]:=MAPEO[7,J];
67901
67902

```

```

67903
67904
68000
68100
68200
68300
68400 FUNCTION RAIZ(NUMERO:INTEGER) : INTEGER;
68500 VAR I,COCIENTE : INTEGER;
68600 BEGIN
68700   COCIENTE:=0;
68800   I:=NUMERO;
68900   WHILE I > 1 DO
69000     BEGIN
69100       I:=I DIV 2;
69200       COCIENTE:=COCIENTE+1;
69300     END;
69400   RAIZ:=COCIENTE;
69500 END; (* RAIZ *)
69600
69700 PROCEDURE DEFHV(NVARIA:INTEGER;VAR NJ,NI:INTEGER);
69800 BEGIN
69900   FOR I:=0 TO 2 DO
70000     BEGIN
70100       ARRVAR1[I]:=' ';
70200       ARRVAR2[I]:=' ';
70300     END;
70400     CASE NVARIA OF
70500       1: BEGIN
70600         WRITELN('PARA UNA SOLA VARIABLE, LA SOLUCION ES: ');
70700         WRITE(' SI LA VARIABLE TOMA EL VALOR DE '1', SE ');
70800         WRITE('TIENE UNA TAUTOLOGIA');
70900         WRITELN;
71000         WRITE(' SI LA VARIABLE TOMA EL VALOR DE '0', SE ');
71100         WRITE('TIENE UNA ANTITAUTOLOGIA ');
71200         WRITELN;
71300         .ABORT;
71400         END;
71500       2: BEGIN
71600         NI:=2;
71700         NJ:=2;
71800         ARRVAR1[0]:=SIMBOLO.LETRAC1;
71900         ARRVAR2[0]:=SIMBOLO.LETRAC2;
72000       END;
72100       3: BEGIN
72200         NI:=2;
72300         NJ:=4;
72400         ARRVAR1[0]:=SIMBOLO.LETRAC1;
72500         ARRVAR1[1]:=SIMBOLO.LETRAC2;
72600         ARRVAR2[0]:=SIMBOLO.LETRAC3;
72700       END;
72800       4: BEGIN
72900         NI:=4;
73000         NJ:=4;
73100         ARRVAR1[0]:=SIMBOLO.LETRAC1;
73200         ARRVAR1[1]:=SIMBOLO.LETRAC2;
73300         ARRVAR2[0]:=SIMBOLO.LETRAC3;
73400         ARRVAR2[1]:=SIMBOLO.LETRAC4;
73500       END;
73600       5: BEGIN
73700         NI:=4;
73800         NJ:=8;
73900         ARRVAR1[0]:=SIMBOLO.LETRAC1;
74000         ARRVAR1[1]:=SIMBOLO.LETRAC2;
74100         ARRVAR1[2]:=SIMBOLO.LETRAC3;
74101
74102

```





```

80303
80304
80400      JJ:=0;
80500      FOR I:=0 TO (NCONJEXTENCNDIGS) DO
80600      BEGIN
80700          BANDERA:=FALSE;
80800          K1:=I;
80900          IF CONJUNTOCNDIGS,IJ <> CELIMINAJ THEN
81000          BEGIN
81100              FOR J:=0 TO (NCONJEXTENCNDIGS)-1 DO
81200                  IF (J <> I) AND (CONJUNTOCNDIGS,JJ <> CELIMINAJ) AND
81300                      (CONJUNTOCNDIGS,IJ)*CONJUNTOCNDIGS,JJ = CONJUNTOCNDIGS,IJ) TH
81400              BEGIN
81500                  BANDERA:=TRUE;
81600                  CONJUNTOCNDIGS,JJ:=CELIMINAJ;
81700              END;
81800              FOR J:=0 TO NCONJEXTENCNDIGS-1 DO
81900                  IF (J <> I) AND (CONJUNTOCNDIGS,JJ <> CELIMINAJ) AND
82000                      (CONJUNTOCNDIGS,JJ)*CONJUNTOCNDIGS,IJ = CONJUNTOCNDIGS,JJ) THE
82100              BEGIN
82200                  BANDERA:=TRUE;
82300                  CONJUNTOCNDIGS,IJ:=CELIMINAJ;
82400                  K1:=J;
82500                  FOR J1:=0 TO NCONJEXTENCNDIGS-1 DO
82600                      IF (J1 <> K1) AND (CONJUNTOCNDIGS,J1J <> CELIMINAJ) AND
82700                          (CONJUNTOCNDIGS,K1J)*CONJUNTOCNDIGS,J1J =CONJUNTOCNDIGS,K1J) THE
82800                          CONJUNTOCNDIGS,J1J:=CELIMINAJ;
82900                  END; BANDERA:=TRUE; WHILE BANDERA DO BEGIN BANDERA:=FALSE;
83000                  FOR J:=0 TO (JJ-1) DO
83100                      IF CONJUNTOCNDIGS+1,JJ*CONJUNTOCNDIGS,K1J = CONJUNTOCNDIGS,K
83200                      THEN BEGIN
83300                          FOR J1:=J TO JJ-2 DO
83400                              CONJUNTOCNDIGS+1,J1J:=CONJUNTOCNDIGS+1,J1+1J;
83500                          JJ:=JJ-1; BANDERA:=TRUE; J:=JJ;
83600                      END;
83700                  END;
83800                  CONJUNTOCNDIGS+1,JJJ:=CONJUNTOCNDIGS,K1J;
83900                  JJ:=JJ+1;
84000                  IF K1 <> I THEN
84100                      CONJUNTOCNDIGS,K1J:=CELIMINAJ;
84200                  END;
84300          END;
84400          K2:=JJ-1;
84500          FOR I:=0 TO K2 DO
84600          BEGIN
84700              J:=1;
84800              K1:=0;
84900              TOPECIJ:=0;
85000              CONJ2:=CONJUNTOCNDIGS+1,IJ;
85100              WHILE J<=NIMP DO
85200              BEGIN
85300                  IF (J IN CONJ2) THEN
85400                  BEGIN
85500                      TOPECIJ:=TOPECIJ+SOLUCI.NTABSECCJJ;
85600                      SOLUCI.MATSOLCI,K1J:=J;
85700                      K1:=K1+1;
85800                  END;
85900                  J:=J+1;
86000              END;
86100              SOLUCI.NMATSOLCIJ:=K1;
86200          END;
86300      (* SE BUSCA EL CONJUNTO CON COSTO MINIMO *)
86400      FOR I:=0 TO K2 DO
86500          MARCAHINCIIJ:= ' '
86501
94507

```

```

86503
86504
86600 MINIMO:=TOPECOJ
86700 POSICION:=0
86800 FOR I:=1 TO K2 DO
86900 IF TOPECIJ<MINIMO THEN
87000 BEGIN
87100 MINIMO:=TOPECIJ
87200 POSICION:=I
87300 END
87400 MARCAMIN[POSICION]:='*'
87500 (* BUSCA TODOS LOS CONJUNTOS CON EL MISMO COSTO MINIMO *)
87600 FOR I:=0 TO K2 DO
87700 IF I<>POSICION THEN
87800 IF TOPECIJ=MINIMO THEN
87900 MARCAMIN[I]:='*'
88000 CONT:=K2
88100 END (* ACONJUNTO *)
88200
88300 PROCEDURE SECUNDARIOS(J:INTEGER;VAR II:INTEGER);
88400 VAR
88500 JJ,CONT,K : INTEGER;
88600 BEGIN
88700 (* SE OBTIENE LA TABLA DE IMPLICANTES SECUNDARIOS *)
88800 FOR K:=0 TO (SOLUCI.NTABIMPLJJ-1) DO
88900 SOLUCI.IMPSECCII,KJ:=SOLUCI.TABIMPLJ,KJ
89000 FOR I:=0 TO NDIGITOS DO
89100 IF (SOLUCI.CUBIERTOELJ,I)='0') AND (SOLUCI.CUBIERTOENIMPLICI,I)
89200 = ' ') THEN
89300 BEGIN
89400 JJ:=0;
89500 WHILE DIGITOSECIJ<>DIGSECCJJ DO
89600 JJ:=JJ+1;
89700 SOLUCI.NOCUBIERTOELJ,JJ:='0';
89800 SOLUCI.NTABSECCIIJ:=SOLUCI.NTABIMPLJ;
89900 END;
90000 II:=II+1;
90100 END (* SECUNDARIOS *)
90200
90300 PROCEDURE SOLUMAR(VAR CONT:INTEGER;NELEMEN,I:INTEGER;MATRIZ:MATENT);
90400 VAR
90500 BINARIO:ARRAY[0..30,0..30] OF CHAR;
90600 LIM2,K,CONT1,CONT2,CONT3,DIR,DIR1,DECIMAL,DELTA,LIM:INTEGER;
90700 AUXILIAR: STRING;
90800 AUX: CHAR;
90900 BEGIN
91000 DIR:=0;
91100 FOR CONT1:=0 TO (NELEMEN-1) DO
91200 BEGIN
91300 DECIMAL:=MATRIZ[I,CONT1];
91400 FOR CONT2:=0 TO (NVARIA+NENTRA-1) DO
91500 BEGIN
91600 IF (DECIMAL MOD 2) = 1 THEN
91700 BINARIO[DIR,CONT2]:='1'
91800 ELSE
91900 BINARIO[DIR,CONT2]:='0';
92000 DECIMAL:=DECIMAL DIV 2;
92100 END;
92200 DIR:=DIR+1;
92300 END;
92400 DELTA:=2;
92500 LIM2:=NELEMEN DIV 2;
92600 LIM:=LIM2;
92700 IF NELEMEN > 1 THEN
92701
92702

```

```

92703
92704
92800 FOR CONT3:=1 TO LIM2 DO
92900 BEGIN
93000 WHILE NOT(CONT3 IN UNO) DO
93100 CONT3:=CONT3+1;
93200 FOR CONT1:=0 TO (LIM-1) DO
93300 BEGIN
93400 DIR1:=CONT1*DELTA;
93500 DIR:=DIR1+CONT3;
93600 FOR CONT2:=0 TO (NVARIA+NENTRA-1) DO
93700 IF (BINARYOCDIR,CONT2) <> (BINARYOCDIR1,CONT2) THEN
93800 BINARYOCDIR1,CONT2:='X';
93900 END;
94000 LIM:=LIM DIV 2;
94100 DELTA:=DELTA*2;
94200 END;
94300 FOR CONT1:=1 TO NVARIA+NENTRA DO
94400 AUXILIAR[CONT1]:=SIMBOLO.LETRA[CONT1];
94500 FOR CONT1:=1 TO (NVARIA+NENTRA-1) DO
94600 FOR CONT2:=(CONT1+1) TO NVARIA+NENTRA DO
94700 IF AUXILIAR[CONT2] < AUXILIAR[CONT1] THEN
94800 BEGIN
94900 AUX:=AUXILIAR[CONT1];
95000 AUXILIAR[CONT1]:=AUXILIAR[CONT2];
95100 AUXILIAR[CONT2]:=AUX;
95200 END;
95300 CONT1:=0;
95400 FOR K:=(NVARIA+NENTRA-1) DOWNT0 0 DO
95500 IF BINARYO[C,K] <> 'X' THEN
95600 BEGIN
95700 SOLUCI.SOLTOTAL[CONT,CONT1]:=AUXILIAR[K+1];
95800 SOLUCI.SOLTOTAL[CONT,CONT1+1]:=BINARYO[C,K];
95900 CONT1:=CONT1+2;
96000 SOLUCI.NSOLTOTAL[CONT1]:=CONT1;
96100 END;
96200 CONT:=CONT+1;
96300 END; (* SOLUMAO *)
96400
96500 PROCEDURE BURBUJA(VAR DIGITOS:VECTOR;NDIGITOS:INTEGER);
96600 VAR
96700 CONT1,AUX,
96800 CONT2 : INTEGER;
96900 BEGIN
97000 FOR CONT2:=0 TO (NDIGITOS-1) DO
97100 FOR CONT1:=(CONT2+1) TO NDIGITOS DO
97200 IF DIGITOS[CONT1] < DIGITOS[CONT2] THEN
97300 BEGIN
97400 AUX:=DIGITOS[CONT2];
97500 DIGITOS[CONT2]:=DIGITOS[CONT1];
97600 DIGITOS[CONT1]:=AUX;
97700 END;
97800 END; (* DE BURBUJA *)
97900
98000 PROCEDURE SOLUCION(NVARIA:INTEGER);
98100 VAR
98200 MODUL,AUX,RED,DIFERENCIA,J1,K2,L,T,I,J,K,I1 : INTEGER;
98300 CONTCUBRE,ESTADO,I3,I2,K1,CONT,CONT1,DIR,
98400 T1:INTEGER;
98500 AUXILIAR : VECTOR;
98600 BEGIN
98700 BANDERA:=TRUE;
98800 FOR I:=0 TO 64 DO
98900 BEGIN
98901
98902

```

```

98903
98904
99000   FOR J:=0 TO 50 DO
99100   BEGIN
99200     SOLUCI.ARREGLOC1,JJ:=0;
99300     SOLUCI.ARRMARCAC1,JJ:='0';
99400   END;
99500   SOLUCI.NELEMCI1:=0;
99600   SOLUCI.NSOL_TOTALLCI1:=0;
99700 END;
99800   FOR I:=1 TO 30 DO
99900   BEGIN
100000   FOR J:=0 TO 50 DO
100100   BEGIN
100200     SOLUCI.NOCUDIERTOCI,JJ:=' ';
100300     SOLUCI.CUBIERTOCI,JJ:=' ';
100400   END;
100500   SOLUCI.MARCCI1:=' ';
100600 END;
100700   FOR I:=0 TO NDIGITOS DO
100800   CASE DIGITOSCI1 OF
100900     0: SOLUCI.NELEMCI0:=1;
101000     1,2,4,8,16,32,64: BEGIN
101100       SOLUCI.ARREGLOC1,SOLUCI.NELEMCI1J:=DIGITOSCI1;
101200       SOLUCI.NELEMCI1:=SOLUCI.NELEMCI1+1;
101300     END;
101400     3,5,6,9,10,12,17,18,20,24,33,34,36,40,48: BEGIN
101500       SOLUCI.ARREGLOC2,SOLUCI.NELEMCI2J:=DIGITOSCI1;
101600       SOLUCI.NELEMCI2:=SOLUCI.NELEMCI2+1;
101700     END;
101800     7,11,13,14,19,21,22,25,26,28,35,37,38,41,42,44,49,50,52,56: BEGIN
101900       SOLUCI.ARREGLOC3,SOLUCI.NELEMCI3J:=DIGITOSCI1;
102000       SOLUCI.NELEMCI3:=SOLUCI.NELEMCI3+1;
102100     END;
102200     15,23,27,29,30,39,43,45,46,51,53,54,57,58,60: BEGIN
102300       SOLUCI.ARREGLOC4,SOLUCI.NELEMCI4J:=DIGITOSCI1;
102400       SOLUCI.NELEMCI4:=SOLUCI.NELEMCI4+1;
102500     END;
102600     31,47,55,59,61,62: BEGIN
102700       SOLUCI.ARREGLOC5,SOLUCI.NELEMCI5J:=DIGITOSCI1;
102800       SOLUCI.NELEMCI5:=SOLUCI.NELEMCI5+1;
102900     END;
103000     63: BEGIN
103100       SOLUCI.ARREGLOC6,SOLUCI.NELEMCI6J:=DIGITOSCI1;
103200       SOLUCI.NELEMCI6:=SOLUCI.NELEMCI6+1;
103300     END;
103400   END; (* CASE *)
103500   IF SOLUCI.NELEMCI0=1 THEN
103600   IF SOLUCI.NELEMCI1>0 THEN
103700   BEGIN
103800     SOLUCI.ARRMARCAC0,0J:='C';
103900     FOR I:=0 TO (SOLUCI.NELEMCI1*2-1) DO
104000     BEGIN
104100       SOLUCI.ARREGLOC11,IJ:=0;
104200       I:=I+1;
104300       MED:=I DIV 2;
104400       SOLUCI.ARREGLOC11,IJ:=SOLUCI.ARREGLOC1,MEDJ;
104500       SOLUCI.ARRMARCAC1,MEDJ:='C';
104600     END;
104700     SOLUCI.NELEMCI1J:=SOLUCI.NELEMCI1*2;
104800   END
104900   ELSE
105000     SOLUCI.ARRMARCAC0,0J:='*';
105100   K:=11;
105101
105102

```

```

105103
105104
105200 FOR T:=1 TO (NVARIA-1) DO
105300 BEGIN
105400 K:=K+1;
105500 L:=0;
105600 IF ((SOLUCI.NELEMCTJ>0) AND (SOLUCI.NELEMCT+1J>0)) THEN
105700 BEGIN
105800 FOR I:=0 TO (SOLUCI.NELEMCTJ-1) DO
105900 FOR J:=0 TO (SOLUCI.NELEMCT+1J-1) DO
106000 IF ((SOLUCI.ARREGLOCT+1,JJ - SOLUCI.ARREGLOCT,IJ) IN UND)
106100 AND (SOLUCI.ARREGLOCT+1,JJ-SOLUCI.ARREGLOCT,IJ > 0) THEN
106200 BEGIN
106300 SOLUCI.ARREGLOCK,LJ:=SOLUCI.ARREGLOCT,IJ;
106400 SOLUCI.ARREGLOCK,L+1J:=SOLUCI.ARREGLOCT+1,JJ;
106500 L:=L+2;
106600 SOLUCI.ARRMARCACT,IJ:='C';
106700 SOLUCI.ARRMARCACT+1,JJ:='C';
106800 END
106900 ELSE
107000 IF SOLUCI.ARRMARCACT,IJ <> 'C' THEN
107100 SOLUCI.ARRMARCACT,IJ:='*';
107200 END
107300 ELSE
107400 FOR I:=0 TO SOLUCI.NELEMCTJ-1 DO
107500 IF SOLUCI.ARRMARCACT,IJ <> 'C' THEN
107600 SOLUCI.ARRMARCACT,IJ:='*';
107700 SOLUCI.NELEMCKJ:=L;
107800 END;
107900 IF SOLUCI.NELEMENVARIAJ > 0 THEN
108000 FOR I:=0 TO SOLUCI.NELEMENVARIAJ-1 DO
108100 IF SOLUCI.ARRMARCACTNVARIA,IJ <> 'C' THEN
108200 SOLUCI.ARRMARCACTNVARIA,IJ:='*';
108300 K:=21;
108400 T:=11;
108500 K2:=1;
108600 T1:=EXPONENTE(K DIV 10);
108700 WHILE (SOLUCI.NELEMCTJ=0) AND (T<70) DO
108800 BEGIN
108900 T:=T+1;
109000 K:=K+1;
109100 END;
109200 WHILE SOLUCI.ARREGLOCT,IJ <> 0 DO
109300 BEGIN
109400 WHILE (SOLUCI.ARREGLOCT+1,IJ <> 0) OR (SOLUCI.ARREGLOCT,IJ <> 0) DO
109500 IF SOLUCI.ARREGLOCT+1,IJ <> 0 THEN
109600 BEGIN
109700 K1:=0;
109800 BANDERA:=TRUE;
109900 T1:=EXPONENTE(K DIV 10);
110000 FOR I:=0 TO (SOLUCI.NELEMCTJ-1) DO
110100 BEGIN
110200 FOR J:=0 TO (SOLUCI.NELEMCT+1J-1) DO
110300 BEGIN
110400 DIFERENCIA:=SOLUCI.ARREGLOCT+1,JJ-SOLUCI.ARREGLOCT,IJ;
110500 FOR J1:=1 TO K2 DO
110600 BEGIN
110700 J:=J+1;
110800 I:=I+1;
110900 AUX:=SOLUCI.ARREGLOCT+1,JJ-SOLUCI.ARREGLOCT,IJ;
111000 IF (AUX=DIFERENCIA) AND (AUX>0) AND (AUX IN UND) THEN
111100 BANDERA:=BANDERA AND TRUE
111200 ELSE
111300 BANDERA:=FALSE;
111301
111302
111301
111302

```

```

111303
111304
111400      END;
111500      I:=I-K2;
111600      IF BANDERA THEN
111700      BEGIN
111800          J:=J-K2;
111900          FOR I1:=I TO I+K2 DO
112000              SOLUCI.ARRMARCACT,I1J:='C';
112100          FOR I1:=J TO J+K2 DO
112200              SOLUCI.ARRMARCACT+1,I1J:='C';
112300          I1:=I;
112400          I2:=J;
112500          FOR I3:=K1 TO (K1+K2) DO
112600          BEGIN
112700              PBURBCI3-K1J:=SOLUCI.ARREGLOCT,I1J;
112800              PBURBCI3-K1+K2+1J:=SOLUCI.ARREGLOCT+1,I2J;
112900              I1:=I1+1;
113000              I2:=I2+1;
113100          END;
113200          BURBUJA(PBURB,T1-1);
113300          IF K1>0 THEN
113400          BEGIN
113500              FOR I3:=0 TO ((K1 DIV T1) -1) DO
113600              BEGIN
113700                  BANDERA:=TRUE;
113800                  CONT:=0;
113900                  FOR I1:=T1*I3 TO T1*(I3+1)-1 DO
114000                  BEGIN
114100                      IF PBURBCONTJ = SOLUCI.ARREGLOCK,I1J THEN
114200                          BANDERA:=BANDERA AND TRUE
114300                          ELSE
114400                              BANDERA:=FALSE;
114500                          CONT:=CONT+1;
114600                      END;
114700                      IF BANDERA THEN
114800                          I3:=K1;
114900                      END;
115000                      IF NOT BANDERA THEN
115100                      BEGIN
115200                          FOR I3:=K1 TO (K1+T1-1) DO
115300                              SOLUCI.ARREGLOCK,I3J:=PBURBCI3-K1J;
115400                              SOLUCI.NELEMCKJ:=SOLUCI.NELEMCKJ+T1;
115500                              K1:=SOLUCI.NELEMCKJ;
115600                          END;
115700                      END
115800                      ELSE
115900                      BEGIN
116000                          FOR I3:=0 TO (T1-1) DO
116100                              SOLUCI.ARREGLOCK,I3J:=PBURBCI3J;
116200                              SOLUCI.NELEMCKJ:=T1;
116300                              K1:=T1;
116400                          END;
116500                          J:=J+K2;
116600                      END
116700                      ELSE
116800                      FOR I1:=I TO I+K2 DO
116900                          IF SOLUCI.ARRMARCACT,I1J <> 'C' THEN
117000                              SOLUCI.ARRMARCACT,I1J:='*';
117100                          BANDERA:=TRUE;
117200                      END;
117300                      I:=I+K2;
117400                      END;
117500                      T:=T+1;
117501
117502

```

```

117503
117504
117600      K:=K+1;
117700      END
117800      ELSE
117900      BEGIN
118000          FOR I:=0 TO SOLUCI.NELEMCTJ-1 DO
118100              IF SOLUCI.ARRMARCACT,IJ <> 'C' THEN
118200                  SOLUCI.ARRMARCACT,IJ:='*';
118300                  T:=T+2; K:=K+2;
118400      END;
118500      K2:=T1-1;
118600      T:=((T+10) DIV 10)*10+1;
118700      K:=((K+10) DIV 10)*10+1;
118800      WHILE (SOLUCI.NELEMCTJ=0) AND (T<70) DO
118900          BEGIN
119000              T:=T+1;
119100              K:=K+1;
119200          END;
119300      END;
119400      BURBUJA(DIGITOS,NDIGITOS);
119500      NIMPLIC:=1;
119600      T:=70;
119700      WHILE (SOLUCI.NELEMCTJ = 0) AND (T>0)-DO
119800          T:=T-1;
119900      WHILE (T >= 0) AND (SOLUCI.NELEMCTJ <> 0) DO
120000      BEGIN
120100          MODUL:=(T DIV 10)* 10;
120200          WHILE (T > MODUL) OR (T = 0) DO.
120300      BEGIN
120400          J:=0;
120500          T1:=EXPONENTE(T DIV 10);
120600          WHILE J < SOLUCI.NELEMCTJ DO
120700      BEGIN
120800          WHILE (SOLUCI.ARRMARCACT,JJ <> '*') AND (J<SOLUCI.NELEMCTJ) DO
120900              J:=J+1;
121000          IF SOLUCI.ARRMARCACT,JJ = '*' THEN
121100      BEGIN
121200          FOR I:=0 TO (T1-1) DO
121300      BEGIN
121400          SOLUCI.TABIMPCNIMPLIC,IJ:=SOLUCI.ARREGLOCT,JJ;
121500          J:=J+1;
121600          END;
121700          SOLUCI.NTABIMPCNIMPLICJ:=T1;
121800          NIMPLIC:=NIMPLIC+1;
121900      END;
122000      END;
122100      IF T>=0 THEN
122200          T:=T-1;
122300      WHILE (SOLUCI.NELEMCTJ=0) AND (T>MODUL) DO
122400          T:=T-1;
122500      END;
122600      IF T >= 0 THEN
122700          T:=T-1;
122800      WHILE (SOLUCI.NELEMCTJ=0) AND (T>0) DO
122900          T:=T-1;
123000      END;
123100      NIMPLIC:=NIMPLIC-1;
123200      CONJ1:=CJ;
123300      FOR I:=0 TO NCASI DO
123400          CONJ1:=CONJ1+ICASICIJ;
123500      REPEAT
123600          BANDERA:=FALSE;
123700          FOR I:=1 TO NIMPLIC DO
123701
123702

```

```

123703
123704
123800 BEGIN
123900 CONJ2:=[];
124000 FOR J:=0 TO SOLUCI.NTABIMP[C]J-1 DO
124100 CONJ2:=CONJ2+[SOLUCI.TABIMP[C]J,J];
124200 IF CONJ1*CONJ2 = [] THEN
124300 BEGIN
124400 BANDERA:=TRUE;
124500 FOR K:=1 TO NIMPLIC-1 DO
124600 BEGIN
124700 FOR CONT:=0 TO SOLUCI.NTABIMP[K+1]J-1 DO
124800 SOLUCI.TABIMP[K,CONTJ]:=SOLUCI.TABIMP[K+1,CONTJ];
124900 SOLUCI.NTABIMP[KJ]:=SOLUCI.NTABIMP[K+1J];
125000 END;
125100 I:=NIMPLIC;
125200 NIMPLIC:=NIMPLIC-1;
125300 END;
125400 END;
125500 UNTIL NOT BANDERA;
125600 CONJ1:=[];
125700 FOR I:=1 TO NIMPLIC DO
125800 FOR J:=0 TO SOLUCI.NTABIMP[C]J-1 DO
125900 CONJ1:=CONJ1+[SOLUCI.TABIMP[C]J,J];
126000 CONT:=-1;
126100 FOR I:=0 TO NDIGITOS DO
126200 IF DIGITOS[C]I IN CONJ1 THEN
126300 BEGIN
126400 CONT:=CONT+1;
126500 AUXILIAR[CONTJ]:=DIGITOS[C]I;
126600 END;
126700 FOR I:=0 TO CONT DO
126800 DIGITOS[C]I:=AUXILIAR[C]I;
126900 NDIGITOS:=CONT;
127000 FOR I:=1 TO NIMPLIC DO
127100 FOR J:=0 TO NDIGITOS DO
127200 FOR K:=0 TO (SOLUCI.NTABIMP[C]J-1) DO
127300 IF SOLUCI.TABIMP[C]KJ = DIGITOS[C]J THEN
127400 SOLUCI.CUBIERTO[C]J,JJ:='0';
127500 FOR I:=0 TO NDIGITOS DO
127600 BEGIN
127700 CONTCUBRE:=0;
127800 FOR J:=1 TO NIMPLIC DO
127900 IF SOLUCI.CUBIERTO[C]J,IJ = '0' THEN
128000 BEGIN
128100 CONTCUBRE:=CONTCUBRE+1;
128200 ESTADO:=J;
128300 END;
128400 IF CONTCUBRE = 1 THEN
128500 BEGIN
128600 SOLUCI.MARCC[ESTADOJ]:='*';
128700 FOR K:=0 TO NDIGITOS DO
128800 IF SOLUCI.CUBIERTO[C]ESTADO,KJ = '0' THEN
128900 SOLUCI.CUBIERTO[C]NIMPLIC+1,KJ:='0';
129000 END; (* IF *)
129100 END; (* FOR *)
129200 CONTCUBRE:=0;
129300 FOR I:=0 TO NDIGITOS DO
129400 IF SOLUCI.CUBIERTO[C]NIMPLIC+1,IJ = ' ' THEN
129500 CONTCUBRE:=CONTCUBRE+1;
129600 II:=1;
129700 NSECUN:=0;
129800 IF CONTCUBRE > 0 THEN
129900 BEGIN
129901
129902

```



```

129903
129904
130000      I:=0;
130100      FOR J:=0 TO NDIGITOS DO
130200          IF SOLUCI.CUBIERTO<NIMPLIC+1,JJ = ' ' THEN
130300              BEGIN
130400                  DIGSECCIJ:=DIGITOS<JJ;
130500                  I:=I+1;
130600              END;
130700          NDIGSEC:=I-1;
130800          FOR J:=1 TO NIMPLIC DO
130900              IF SOLUCI.MARCIJJ <> '*' THEN
131000                  SECUNDARIOS(J,I);
131100          NIMPSEC:=II-1;
131200          ACONJUNTO(NDIGSEC,NIMPSEC,SOLUCI.NTABSEC,SOLUCI.NOCUBIERTO,NSECUN)
131300      END; (* IF *)
131400      CONT:=1;
131500      FOR I:=1 TO (NIMPLIC) DO
131600          IF SOLUCI.MARCIJ = '*' THEN
131700              SOLUMAQ(CONT,SOLUCI.NTABIMP<I,I,SOLUCI.TABIMP);
131800          FOR I:=0 TO NSECUN DO
131900              BEGIN
132000                  K:=0;
132100                  FOR J:=1 TO CONT-1 DO
132200                      BEGIN
132300                          J1:=0;
132400                          WHILE J1<SOLUCI.NSOLTOTAL<JJ DO
132500                              BEGIN
132600                                  SOLUCIONES<I,KJ:=SOLUCI.SOLTOTAL<EJ,J1;
132700                                  K:=K+1;
132800                                  J1:=J1+1;
132900                              END;
133000                                  SOLUCIONES<I,KJ:='+';
133100                                  K:=K+1;
133200                              END;
133300                                  IF K < 2 THEN
133400                                      NSOLUCIONES<I:=0;
133500                                  ELSE
133600                                      NSOLUCIONES<I:=K-1;
133700                                  END;
133800          IF CONTCUBRE > 0 THEN
133900              BEGIN
134000                  FOR I:=0 TO NSECUN DO
134100                      FOR J:=0 TO SOLUCI.NMATSOL<I-1 DO
134200                          BEGIN
134300                              SOLUMAQ(CONT,SOLUCI.NTABSECC<SOLUCI.MATSOL<I,JJJ,SOLUCI.MATSOL<I,J
134400                                  SOLUCI.IMPSEC);
134500                              IF NSOLUCIONES<I> > 0 THEN
134600                                  BEGIN
134700                                      SOLUCIONES<I;NSOLUCIONES<IJJ:='+';
134800                                      J1:=NSOLUCIONES<IJJ+2;
134900                                  END
135000                                  ELSE
135100                                      J1:=0;
135200                                  K:=0;
135300                                  WHILE K<SOLUCI.NSOLTOTAL<CONT-1 JJ DO
135400                                      BEGIN
135500                                          SOLUCIONES<I,J1:=SOLUCI.SOLTOTAL<CONT-1,KJ;
135600                                          K:=K+1;
135700                                          J1:=J1+1;
135800                                      END;
135900                                          NSOLUCIONES<I:=J1;
136000                                  END;
136100          END;
136101
136102

```

```

136103
136104
136200 TOTSOL:=CONT;
136300 END; (* DE SOLUCION *)
136400
136500 PROCEDURE MAPABINARIO(NVARIA:INTEGER);
136600 VAR I,J,K,L: INTEGER;
136700 BEGIN
136800   FOR I:=0 TO NI-1 DO
136900     FOR J:=0 TO NJ-1 DO
137000       BEGIN
137100         L:=MAPASCI,JJ;
137200         FOR K:=0 TO NVARIA+NENTRA-1 DO
137300           BEGIN
137400             SIMBOLO.BINARIOCI,J,KJ:=L MOD 2;
137500             L:=L DIV 2;
137600           END;
137700         END;
137800       END; (* MAPABINARIO *)
137900
138000 PROCEDURE MAPAACTUAL(NVARIA:INTEGER);
138100 VAR PARAMAPA,I,J,I1,J1 : INTEGER;
138200     AUXILIAR: STRING;
138300     AUX1 : CHAR;
138400 BEGIN
138500   FOR I:=1 TO NVARIA DO
138600     AUXILIARCIJ:=SIMBOLO.LETRACIJ;
138700     FOR I:=1 TO NVARIA-1 DO
138800       FOR J:=I+1 TO NVARIA DO
138900         IF AUXILIARCIJ > AUXILIARCJ THEN
139000           BEGIN
139100             AUX1:=AUXILIARCIJ;
139200             AUXILIARCIJ:=AUXILIARCJ;
139300             AUXILIARCJ:=AUX1;
139400           END;
139500           I:=I;
139600           FOR J:=0 TO (NVARIA-1) DO
139700             BEGIN
139800               WHILE SIMBOLO.LETRACJ+1J <> AUXILIARCIJ DO
139900                 I:=I+1;
140000                 CASE I OF
140100                   1 : SIMBOLO.NOASOCCJJ:=1;
140200                   2 : SIMBOLO.NOASOCCJJ:=2;
140300                   3 : SIMBOLO.NOASOCCJJ:=4;
140400                   4 : SIMBOLO.NOASOCCJJ:=8;
140500                   5 : SIMBOLO.NOASOCCJJ:=16;
140600                   6 : SIMBOLO.NOASOCCJJ:=32;
140700                 END;
140800                 I:=I;
140900             END;
141000             I:=0;
141100           FOR I1:=0 TO NI-1 DO
141200             FOR J1:=0 TO NJ-1 DO
141300               BEGIN
141400                 PARAMAPA:=0;
141500                 FOR J:=0 TO (NVARIA-1) DO
141600                   PARAMAPA:=PARAMAPA+SIMBOLO.NOASOCCJJ*SIMBOLO.
141700                     BINARIOCI1,J1,JJ;
141800                   MAPASCI1,J1J:=PARAMAPA;
141900                   I:=I+1;
142000                 END;
142100             END; (* MAPAACTUAL *)
142200
142300 PROCEDURE MAPASO1;
142301
142302

```

```

142303
142304
142400 BEGIN
142500 IF NOT BANDERA2 THEN BEGIN
142600 FOR I:=0 TO NI-1 DO
142700 FOR J:=0 TO NJ-1 DO
142800 FOR K:=0 TO NDIGITOS DO
142900 IF DIGITOSEKJ = MAPASCI,JJ THEN
143000 MAPASMINCI,JJ:='1' END
143100 ELSE
143200 FOR I:=0 TO NI-1 DO
143300 FOR J:=0 TO NJ-1 DO
143400 FOR K:=0 TO NDIGITOS DO
143500 IF (DIGITOSEKJ = MAPASCI,JJ) AND (CDIGITOSEKJ = 'X') THEN
143600 CMAPASMINCI,JJ:='X';
143700 END; (* MAPAS01 *)
143800
143900 PROCEDURE MAPAORIG;
144000 VAR CONT,CONT1,CONT2:INTEGER;
144100 BEGIN
144200 NDIGITOS:=-1;
144300 FOR CONT:=0 TO 8 DO
144400 FOR CONT1:=0 TO 8 DO
144500 BEGIN
144600 CMAPASMINCCONT,CONT1J:=' '; MAPASMINCONT,CONT1J:='0';
144700 END;
144800 MAPABINARIO(NVARIA+NENTRA);
144900 MAPAACTUAL(NVARIA+NENTRA);
145000 CONT2:=EXPONENTE(NENTRA)*NESTADOS;
145100 FOR CONT:=0 TO NI-1 DO
145200 FOR CONT1:=0 TO NJ-1 DO
145300 IF MAPASCONT,CONT1J.>= CONT2 THEN
145400 MAPASMINCCONT,CONT1J:='*'; CONT2:=0;
145500 IF NENTRADAS < EXPONENTE(NENTRA)-1 THEN BEGIN
145600 FOR CONT:=1 TO NESTADOS DO
145700 FOR CONT1:=0 TO EXPONENTE(NENTRA)-1 DO
145800 BEGIN
145900 IF CONT1 > NENTRADAS THEN
146000 BEGIN
146100 NDIGITOS:=NDIGITOS+1;
146200 DIGITOSCNDIGITOSJ:=CONT2;
146300 END;
146400 CONT2:=CONT2+1;
146500 END;
146600 FOR CONT:=0 TO NI-1 DO
146700 FOR CONT1:=0 TO NJ-1 DO
146800 FOR CONT2:=0 TO NDIGITOS DO
146900 IF DIGITOSCCONT2J = MAPASCONT,CONT1J THEN
147000 MAPASMINCCONT,CONT1J:='*';
147100 END;
147200 END; (* MAPAORIG *)
147300
147400 PROCEDURE RESUELVEAL(CONTA:INTEGER);
147500 LABEL 3; VAR CONT,CONTAB:INTEGER;
147600 BEGIN 3; WRITELN;
147700 IF OPCIONC1J <> '3' THEN
147800 WRITELN('PARA ',OPCIONC2J:1,(' SIMBOL',LETRAENVARIA+1+
147900 NENTRA-CONTA:1,') APLICANDO EL METODO, SE GENERA ',
148000 'EL MAPA SIGUIENTE');
148100 FOR CONT:=0 TO 32 DO
148200 BEGIN
148300 CASDIBCONTJ:=' '; CDIGITOSCCONTJ:=' ';
148400 END;
148500 FOR CONT:=0 TO 8 DO
148501
148502

```

```

148503
148504
148600     FOR CONT1:=0 TO 8 DO
148700         CHAPASMINCONT,CONT1:= ' ';
148800         BANDERA2:=TRUE;
148900         BANDERA:=TRUE;
149000         GRAFICA(MAPAS,MAPASMIN,CHAPASMIN);
149100         SOLUCION(NVARIA+NENTRA);
149200         IF NDIGITOS = -1 THEN
149300             BEGIN
149400                 WRITELN;
149500                 WRITELN('COMO PODRAS OBSERVAR, NO EXISTEN ELEMENTOS ',
149600                     'CON LOS CUALES MINIMIZAR DE');
149700                 WRITE('DE ACUERDO AL METODO, POR LO CUAL, TU SOLUCION ',
149800                     'ES O ');
149900                 EXIT(RESUELVEAL);
150000             END;
150100             WRITELN;
150200     WRITELN('DE ACUERDO AL MAPA ANTERIOR, PROPORCIONA LOS MINTERMINOS
150300         'QUE SE PUEDEN');
150400     WRITELN('COMBINAR, CADA UNO SEPARADO POR (,) Y TERMINAR CON (;)');
150500     WRITELN;
150600     WRITELN('EXISTEN MINTERMINOS QUE SE COMBINEN?');
150700     READLN;
150800     READ(CAMBIOC1J,CAMBIOC2J);
150900     WHILE NOT (((CAMBIOC1J = 'S') AND (CAMBIOC2J = 'I')) OR
151000         ((CAMBIOC1J = 'N') AND (CAMBIOC2J = 'O'))) DO
151100     BEGIN
151200         WRITELN('REPITE SI HAY MINTERMINOS QUE SE COMBINEN');
151300         READLN;
151400         READ(CAMBIOC1J,CAMBIOC2J);
151500     END;
151600     CONTAB:=1;
151700     CONT2:=0;
151800     WHILE ((CAMBIOC1J = 'S') OR (CAMBIOC1J='N')) DO
151900     IF CAMBIOC1J = 'S' THEN
152000     WHILE CAMBIOC1J='S' DO
152100     BEGIN
152200         WRITELN('PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA',
152300             'FORMA INDICADA');
152400         CASILLASC0J:=0;
152500         NCASILLAS:=0;
152600         DECIMALES(CASILLAS,NCASILLAS);
152700         IF NCASILLAS > 0 THEN
152800             IF (NCASILLAS+1) IN UNO THEN
152900                 BEGIN
153000                     BURBUJA(CASILLAS,NCASILLAS);
153100                     T1:=RAIZ(NCASILLAS+1);
153200                     T:=T1*10 + 10;
153300                     WHILE (SOLUCI.NELEMETJ = 0) AND (T > (T1*10)) DO
153400                         T:=T-1;
153500                     BANDERA:=FALSE;
153600                     IF T > T1*10 THEN
153700                         WHILE (NOT BANDERA) AND (T>T1*10) DO
153800                             BEGIN
153900                                 FOR CONT:=0 TO SOLUCI.NELEMETJ-1 DO
154000                                     BEGIN
154100                                         IF SOLUCI.ARREGLOIT,CONTJ = CASILLASC0J THEN
154200                                             BEGIN
154300                                                 BANDERA:=TRUE;
154400                                                 FOR CONT1:=0 TO NCASILLAS DO
154500                                                     BEGIN
154600                                                         IF SOLUCI.ARREGLOIT,CONTJ=CASILLASC0J THEN
154700                                                             BANDERA:=BANDERA AND TRUE
154701
154702

```



```

160903
160904
161000 BANDERA1:=TRUE;
161100 FOR CONT:=0 TO NCASILLAS DO
161200 BEGIN
161300 BANDERA:=TRUE;
161400 FOR CONT1:=0 TO NDIGITOS DO
161500 IF CASILLAS[CONTJ] = DIGITOS[CONT1] THEN
161600 BANDERA:=FALSE;
161700 IF BANDERA THEN
161800 BANDERA1:=FALSE;
161900 END;
162000 IF NOT BANDERA1 THEN
162100 WRITELN('UNO DE LOS NUMEROS QUE PROPORCIONASTE NO ES',
162200 ' MINTERMINO');
162300 ELSE
162400 WRITELN('LOS MINTERMINOS QUE PROPORCIONASTE NO SE PUEDEN',
162500 ' COMBINAR');
162600 BANDERA:=FALSE;
162700 END;
162800 IF CAMBIO[1] = 'N' THEN
162900 BEGIN
163000 FOR CONT:=0 TO NCASI DO
163100 IF CASDIG[CONTJ] = 'C' THEN
163200 BANDERA:=BANDERA AND TRUE
163300 ELSE
163400 BANDERA:=FALSE;
163500 IF NOT BANDERA THEN
163600 BEGIN
163700 WRITELN('AUN HAY MINTERMINOS POR CUBRIR, INTENTALO');
163800 WRITELN('SE PUEDEN COMBINAR ENTRE ELLOS',
163900 ' LOS MINTERMINOS QUE NO SE HAN CUBIERTO?');
164000 READLN;
164100 READ(CAMBIO[1],CAMBIO[2]);
164200 WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
164300 ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
164400 BEGIN
164500 WRITELN('SE PUEDEN COMBINAR ENTRE ELLOS O NO?');
164600 READLN;
164700 READ(CAMBIO[1],CAMBIO[2]);
164800 END;
164900 END
165000 ELSE
165100 CAMBIO[1]:='F';
165200 END
165300 ELSE
165400 BEGIN
165500 BANDERA1:=TRUE;
165600 FOR CONT:=0 TO NCASI DO
165700 IF CASDIG[CONTJ] <> 'C' THEN
165800 BANDERA1:=FALSE;
165900 IF BANDERA1 THEN
166000 BEGIN
166100 WRITE('TODOS LOS MINTERMINOS YA ESTAN CUBIERTOS, POR ',
166200 ' LO CUAL ');
166300 CAMBIO[1]:='F';
166400 END;
166500 END;
166600 END
166700 ELSE
166800 BEGIN
166900 WRITELN('EL NUMERO DE MINTERMINOS NO ES POTENCIA DE 2,',
167000 ' POR LO CUAL NO SE PUEDEN ');
167100 WRITELN('COMBINAR');
167101
167102

```

```

167103
167104
167200 END
167300 ELSE
167400 BEGIN
167500 WRITELN('SI SOLO SE TRATA DE UN MINTERMINO, QUIERE DECIR QUE NO',
167600 ' SE PUEDE COMBINAR, ');
167700 WRITELN('REPITELO');
167800 CAMBIO[1]='N';
167900 END
168000 END (* WHILE DE SI *)
168100 ELSE
168200 BEGIN
168300 WRITELN('PROPORCIONA CADA MINTERMINO QUE NO SE COMBINO, ',
168400 ' 1 A LA VEZ');
168500 WHILE CAMBIO[1]='N' DO
168600 BEGIN
168700 DECIMALES(CASILLAS,NCASILLAS);
168800 WHILE NCASILLAS <> 0 DO
168900 BEGIN
169000 WRITELN('SOLO 1 A LA VEZ, REPITE');
169100 DECIMALES(CASILLAS,NCASILLAS);
169200 END;
169300 T1:=0;
169400 T:=10;
169500 WHILE ((SOLUCI.NELEM[T]=0) AND (T>=0)) DO
169600 T:=T-1;
169700 BANDERA:=FALSE;
169800 WHILE (NOT BANDERA) AND (T>=0) DO
169900 BEGIN
170000 FOR CONT:=0 TO SOLUCI.NELEM[T]-1 DO
170100 IF SOLUCI.ARREGLO[CONT]=CASILLAS[0] THEN
170200
170300 IF SOLUCI.ARRMARCA[CONT]='*' THEN
170400 BEGIN
170500 CONT:=SOLUCI.NELEM[T];
170600 BANDERA:=TRUE;
170700 END
170800 ELSE
170900 BEGIN
171000 WRITELN('EL MINTERMINO, 'CASILLAS[0][2], ' SI SE PUEDE ',
171100 'COMBINAR, INTENTALO');
171200 CAMBIO[1]='S';
171300 T:=0;
171400 END;
171500 T:=T-1;
171600 END;
171700 IF BANDERA THEN
171800 BEGIN
171900 FOR CONT1:=0 TO NCASILLAS DO
172000 FOR CONT2:=0 TO NCASI DO
172100 IF CASILLAS[CONT1] = CASI[CONT2] THEN
172200 IF CASDIG[CONT2] <> 'C' THEN
172300 BANDERA:=FALSE;
172400 IF NOT BANDERA THEN
172500 BEGIN
172600 FOR CONT:=0 TO NCASILLAS DO
172700 TABALUM[CONTAB,CONT]:=CASILLAS[CONT];
172800 NTABALUM[CONTAB]:=NCASILLAS+1;
172900 CONTAB:=CONTAB+1;
173000 FOR CONT1:=0 TO NCASILLAS DO
173100 FOR CONT:=0 TO NCASI DO
173200 IF CASILLAS[CONT1] = CASI[CONT] THEN
173300 CASDIG[CONT]='C';
173301
173302

```

```

173303
173304
173400 WRITELN('BIEN, EL MINTERMINO NO SE COMBINA, POR LO CUAL NO SE REDUCE');
173500   BANDERA:=TRUE;
173600
173700   FOR CONT:=0 TO NCASILLAS DO
173800     FOR CONTI:=0 TO NDIGITOS DO
173900       IF CASILLAS[CONT] = DIGITOS[CONTI] THEN
174000         CDIGITOS[CONTI]:='X';
174100   BANDERA2:=TRUE;
174200   MAPAS01;
174300   GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
174400 END
174500 ELSE
174600 BEGIN
174700   WRITELN('TODOS LOS MINTERMINOS QUE PROPONES YA ESTAN CUBIERTOS,',
174800     'NO LOGRAS MINIMIZAR');
174900   WRITELN('MAS TU FUNCION');
175000 END;
175100 IF CAMBIO[1] = 'N' THEN
175200 BEGIN
175300   WRITELN('AUN HAY MINTERMINOS QUE NO SE COMBINEN?');
175400   READLN; READ(METODO[1],METODO[2]);
175500   WHILE NOT (((METODO[1] = 'S') AND (METODO[2] = 'I')) OR
175600     ((METODO[1] = 'N') AND (METODO[2] = 'O'))) DO
175700     BEGIN
175800       WRITELN('REPITE SI AUN HAY MINTERMINOS QUE NO SE COMBINEN');
175900       READLN; READ(METODO[1],METODO[2]);
176000     END;
176100   END;
176200 IF METODO[1]='S' THEN
176300 BEGIN
176400   BANDERA1:=TRUE;
176500   FOR CONT:=0 TO NDIGITOS DO
176600     IF CDIGITOS[CONT] <> 'C' THEN
176700       BANDERA1:=FALSE;
176800   IF BANDERA1 THEN
176900     BEGIN
177000       WRITE('TODOS LOS MINTERMINOS YA ESTAN CUBIERTOS, POR LO CUAL');
177100       CAMBIO[1]:='F';
177200     END
177300     ELSE
177400     BEGIN
177500       WRITELN('PROPORCIONAME EL SIGUIENTE');
177600       CAMBIO[1]:='N'
177700     END;
177800   END
177900 ELSE
178000 BEGIN
178100   FOR CONT:=0 TO NDIGITOS DO
178200     IF CASDIG[CONT] = 'C' THEN
178300       BANDERA:=BANDERA AND TRUE
178400     ELSE
178500       BANDERA:=FALSE;
178600   IF NOT BANDERA THEN
178700     BEGIN
178800       WRITELN('AUN HAY MINTERMINOS POR CUBRIR, INTENTALO');
178900       WRITELN('SE PUEDEN COMBINAR ENTRE ELLOS LOS MINTERMINOS ',
179000         'QUE FALTAN POR CUBRIR?');
179100       READLN; READ(CAMBIO[1],CAMBIO[2]);
179200       WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
179300         ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
179400         BEGIN
179500           WRITELN('SE PUEDEN COMBINAR ENTRE ELLOS O NO?');
179501
179502

```



```

179503
179504
179600 READLN; READ(CAMBIOC1],CAMBIOC2]);
179700 END;
179800 END
179900 ELSE CAMBIOC1]:= 'F';
180000 END;
180100 END;
180200 IF (NOT BANDERA) AND (T<0) AND (CAMBIOC1]='N') THEN
180300 WRITELN('EL PROPUESTO NO ES MINTERMINO, DAME OTRO QUE SI LO SEA
180400 END
180500 END;
180600 IF BANDERA THEN
180700 BEGIN
180800 CONT:=TOTSOL;
180900 CONT2:=0;
181000 FOR CONT1:=1 TO CONTAB-1 DO
181100 BEGIN
181200 SOLUMAQ(CONT,NTABALUM[CONT1],CONT1,TABALUM);
181300 CONT4:=0;
181400 FOR CONT3:=CONT2 TO (SOLUCI.NSOLTOTAL[CONT-1]+CONT2) DO
181500 BEGIN
181600 SOLALUM[CONT3]:=SOLUCI.SOLTOTAL[CONT-1,CONT4];
181700 CONT4:=CONT4+1;
181800 END;
181900 SOLALUM[CONT3]:= '+';
182000 CONT2:=CONT3+1;
182100 END;
182200 WRITELN('TU SOLUCION ES:');
182300 CONT1:=0;
182400 CONT3:=0;
182500 IF (CONTAB=2) AND (SOLUCI.NSOLTOTAL[CONT-1]=0) THEN
182600 WRITELN('UNA TAUTOLOGIA, ES DECIR, 1')
182700 ELSE
182800 WHILE (CONT1<(CONT2-1)) DO
182900 BEGIN
183000 WHILE (CONT1 < (CONT2-1)) AND (CONT3 < 75) DO
183100 BEGIN
183200 WRITE(SOLALUM[CONT1]:1);
183300 CONT1:=CONT1+1;
183400 CONT3:=CONT3+1;
183500 END;
183600 WRITELN;
183700 CONT3:=0;
183800 END;
183900 WRITELN;
184000 END;
184100 IF NSECUN = 0 THEN
184200 IF NSOLUCIONESCOJ>=(CONT2-1) THEN
184300 WRITELN('PERFECTO, OBTUVISTE LA SOLUCION OPTIMA')
184400 ELSE
184500 BEGIN
184600 WRITELN('TU SOLUCION ES CORRECTA PERO NO LA OPTIMA ',
184700 'DESEAS INTENTAR LLEGAR A ELLA?');
184800 READLN;
184900 READ(CAMBIOC1],CAMBIOC2]);
185000 WHILE NOT ((CAMBIOC1]='S') AND (CAMBIOC2]='I') OR
185100 ((CAMBIOC1]='N') AND (CAMBIOC2]='O')) DO
185200 BEGIN
185300 WRITELN('REPITE SI DESEAS INTENTAR LLEGAR A LA SOLUCION ',
185400 'OPTIMA O NO');
185500 READLN;
185600 READ(CAMBIOC1],CAMBIOC2]);
185700 END;
185701
185702

```

```

185703
185704
185800 IF CAMBIO[1]='N' THEN
185900 BEGIN
186000   WRITELN('DESEAS CONOCER ANTES LA SOLUCION OPTIMA?');
186100   READLN;
186200   READ(CAMBIO[1],CAMBIO[2]);
186300   WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
186400             ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
186500     BEGIN
186600       WRITELN('REPITE SI DESEAS CONOCER LA SOLUCION OPTIMA O NO');
186700       READLN;
186800       READ(CAMBIO[1],CAMBIO[2]);
186900     END;
187000     IF CAMBIO[1] = 'S' THEN
187100       BEGIN
187200         CONT1:=0;
187300         CONT3:=0;
187400         IF NSOLUCIONES[0] = 0 THEN
187500           WRITELN('UNA TAUTOLOGIA, ES DECIR, 1')
187600         ELSE
187700           WHILE (CONT1<NSOLUCIONES[0]) DO
187800             BEGIN
187900               WHILE (CONT1 < NSOLUCIONES[0]) AND (CONT3 < 75) DO
188000                 BEGIN
188100                   WRITE(SOLUCIONES[0,CONT1]:1);
188200                   CONT1:=CONT1+1;
188300                   CONT3:=CONT3+1;
188400                 END;
188500                 WRITELN;
188600                 CONT3:=0;
188700             END;
188800             WRITELN;
188900           END;
189000         END
189100         ELSE
189200           GOTO 3;
189300         END
189400       ELSE
189500         BEGIN
189600           CONT:=0;
189700           BANDERA:=FALSE;
189800           WHILE (CONT <= NSECUN) AND (NOT BANDERA) DO
189900             BEGIN
190000               IF MARCAMIN[CONT] = '*' THEN
190100                 IF (CONT-1)<=NSOLUCIONES[CONT] THEN
190200                   BEGIN
190300                     WRITELN('PERFECTO, OBTUVISTE LA SOLUCION OPTIMA');
190400                     BANDERA:=TRUE;
190500                   END;
190600                   CONT:=CONT+1;
190700                 END;
190800               IF NOT BANDERA THEN
190900                 BEGIN
191000                   WRITELN('TU SOLUCION ES CORRECTA PERO NO OPTIMA, DESEAS ',
191100                     'LLEGAR A ELLA?');
191200                   READLN;
191300                   READ(CAMBIO[1],CAMBIO[2]);
191400                   WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
191500                             ((CAMBIO[1]='N') AND (CAMBIO[2]='O'))) DO
191600                     BEGIN
191700                       WRITELN('REPITE SI DESEAS INTENTAR LLEGAR A LA SOLUCION ',
191800                         'OPTIMA O NO');
191900                       READLN;
191901
191902

```

```

191903
191904
192000      READ(CAMBIO[1],CAMBIO[2]);
192100      END;
192200      IF CAMBIO[1] = 'N' THEN
192300      BEGIN
192400          WRITELN('DESEAS CONOCER LA(S) SOLUCION(ES) OPTIMA(S)?');
192500          READLN;
192600          READ(CAMBIO[1],CAMBIO[2]);
192700          WHILE NOT (((CAMBIO[1]='S') AND (CAMBIO[2]='I')) OR
192800                    ((CAMBIO[1]='N') AND (CAMBIO[2]='O')))) DO
192900          BEGIN
193000              WRITELN('REPITE SI DESEAS CONOCER LA SOLUCION OPTIMA');
193100              READLN;
193200              READ(CAMBIO[1],CAMBIO[2]);
193300          END;
193400          IF CAMBIO[1]='S' THEN
193500          FOR CONT1:=0 TO NSECUN DO BEGIN
193600              IF MARCAMIN[CONT1] = '*' THEN
193700              FOR CONT2:=0 TO NSOLUCIONES[CONT1] DO
193800                  WRITE(SOLUCIONES[CONT1,CONT2]);
193900              WRITELN;
194000          END;
194100      END
194200      ELSE
194300      GOTO 3;
194400      END;
194500      END;
194600      END; (* RESUELEAL *)
194700      (* PROGRAMA PRINCIPAL *)
194800      BEGIN
194900      1: FOR CONT:=0 TO 20 DO
195000          BEGIN
195100              FOR CONT1:=1 TO 3 DO
195200              BEGIN
195300                  EPRESENTE[CONT,CONT1]:=' ';
195400                  FOR CONT2:=0 TO 5 DO
195500                      EFUTURO[CONT2,CONT,CONT1]:=' ';
195600              END;
195700              FOR CONT1:=0 TO 5 DO
195800                  SALIDA[CONT1,CONT]:=-1;
195900                  MARCA[CONT1]:=C;
196000                  MARCAALUM[CONT1]:=C;
196100          END;
196200          WRITELN('ESTE PROGRAMA TE GUIARA EN LA MINIMIZACION DE ESTA',
196300                'DOS PARA CIRCUITOS');
196400          WRITELN('SECUENCIALES, DESEAS CONOCER UNA PEQUESA INTRODUCCION',
196500                'DE COMO LO HACE');
196600          WRITELN('ESTE PROGRAMA?');
196700          READ(OPCION[1],OPCION[2]);
196800          WHILE NOT (((OPCION[1]='S') AND (OPCION[2]='I')) OR
196900                    ((OPCION[1]='N') AND (OPCION[2]='O')))) DO
197000          BEGIN
197100              WRITELN('REPITE SI DESEAS O NO HACERLO');
197200              READLN; READ(OPCION[1],OPCION[2]);
197300          END;
197400          IF OPCION[1] = 'S' THEN
197500          TEXTO;
197600          UNO:=1,2,4,8,16,32,64;
197700          BANDEX:=FALSE;
197800          WRITELN('CUANTOS ESTADOS TIENE TU TABLA?');
197900          READLN; READ(NESTADOS);
198000          WHILE (NESTADOS < 2) OR (NESTADOS > 16) DO
198100          BEGIN
198100

```

```

198103
198104
198200
198300
198400
198500
198600
198700
198800
198900
199000
199100
199200
199300
199400
199500
199600
199700
199800
199900
200000
200100
200200
200300
200400
200500
200600
200700
200800
200900
201000
201100
201200
201300
201400
201500
201600
201700
201800
201900
202000
202100
202200
202300
202400
202500
202600
202700
202800
202900
203000
203100
203200
203300
203400
203500
203600
203700
203800
203900
204000
204100
204200
204300
204301
204302

```

```

IF NESTADOS < 2 THEN
BEGIN
  WRITELN('ESTE PROGRAMA ESTA ENFOCADO A LA ',
          ' REDUCCION DE ESTADOS DE CIRCUITOS');
  WRITELN('SECUENCIALES POR LO CUAL UN CIRCUITO CON MENOS',
          ' DE DOS ESTADOS NO TIENE');
  WRITELN('REDUCCION, VERIFICA');
END
ELSE
BEGIN
  WRITELN('ESTE PROGRAMA TIENE OBJETIVOS DIDACTICOS, ',
          ' Y SUJETO AL TAMAÑO DE LA');
  WRITELN('PANTALLA, POR LO CUAL NO RESUELVO TABLAS CON MAS DE ',
          ' 16 ESTADOS, DESEAS');
  WRITELN('PROPONER ALGUNA CON ESTA CARACTERISTICA?');
  READLN;
  READ(OPCIONC1,OPCIONC2);
  MIENTRAS;
  IF OPCIONC1 = 'S' THEN
    WRITELN('CUANTOS ESTADOS TIENE AHORA TU TABLA?')
  ELSE
    ABORT;
  END;
  READLN; READ(NESTADOS);
END;
WRITELN;
WRITELN('CUANTAS ENTRADAS TIENE TU TABLA?');
READLN;
READ(NENTRADAS);
WHILE (NENTRADAS < 1) OR (NENTRADAS > 5) DO
BEGIN
  IF NENTRADAS < 1 THEN
    WRITELN('VERIFICA EL NUMERO DE ENTRADAS')
  ELSE
    BEGIN
      WRITELN('DADO QUE ESTE PROGRAMA ES DE TIPO DIDACTICO',
              ' Y CONVERSACIONAL, ESTA SUJETO');
      WRITELN('AL TAMAÑO DE LA PANTALLA Y NO PUEDO MANEJAR ',
              ' MAS DE 5 ENTRADAS. ');
      WRITELN('DESEAS PROPONER LA MISMA TABLA CON UN NUMERO',
              ' DE ESTADOS QUE PUEDA RESOLVER?');
      READLN;
      READ(OPCIONC1,OPCIONC2);
      MIENTRAS;
      IF OPCIONC1 = 'S' THEN
        WRITELN('CUANTAS ENTRADAS TIENE AHORA TU FUNCION?')
      ELSE
        ABORT;
      END;
      READLN; READ(NENTRADAS);
    END;
  NENTRADAS:=NENTRADAS-1;
  WRITELN; WRITELN('DEFINE CADA ESTADO CON UN IDENTIFICADOR',
                  ' NO MAYOR DE 3 CARACTERES. ');
  FOR CONT1:=1 TO NESTADOS DO
  BEGIN
    WRITELN('EL ESTADO ',(CONT1-1):1,' SE LLAMA');
    READLN;
    CONT:=1;
    WHILE (NOT EOLN) AND (CONT < 4) DO
    BEGIN
      READ(EPRESENTE[CONT1,CONT]);

```

```

204303
204304
204400          CONT:=CONT+1;
204500      END;
204600      IF CONT1 > 1 THEN
204700          FOR CONT2:=1 TO CONT1-1 DO
204800              BEGIN
204900                  BANDERA:=TRUE;
205000                  FOR CONT3:=1 TO 3 DO
205100                      IF EPRESENTE[CONT2,CONT3]<>EPRESENTE[CONT1,CONT3] THEN
205200                          BANDERA:=FALSE;
205300                  IF BANDERA THEN
205400                      BEGIN
205500                          CONT1:=CONT1-1;
205600                          CONT2:=CONT1;
205700                          WRITELN('ESTE ESTADO YA EXISTE, VERIFICA');
205800                          FOR CONT3:=1 TO 3 DO
205900                              EPRESENTE[CONT1+1,CONT3]:= ' ';
206000                          END;
206100                      END;
206200          END;
206300      EDOSFUTUROS;
206400      WRITELN('LA TABLA QUE SE ORIGINA ES ');
206500      TABLA;
206600
206700      WRITELN('DESEAS HACER ALGUN CAMBIO A LA TABLA?');
206800      READLN;
206900      READ(OPCIONC1,OPCIONC2);
207000      MIENTRAS;
207100      IF OPCIONC1 = 'S' THEN
207200          BEGIN
207300              REPEAT
207400                  WRITELN('CUAL ES EL ESTADO PRESENTE DONDE DESEAS',
207500                      ' HACER LA MODIFICACION?');
207600                  LEESTADOS(EDOAX,0,0,LUGAR);
207700                  CONT3:=LUGAR;
207800                  FOR CONT2:=0 TO NENTRADAS DO
207900                      BEGIN
208000                          WRITELN('CUAL ES EL ESTADO EN T+1 DE ESTE ',
208100                              ' ESTADO PARA ENTRADA =',CONT2:1);
208200                          LEESTADOS(EFUTURO,CONT2,CONT3,LUGAR);
208300                          WRITELN('CUAL ES LA SALIDA CORRESPONDIENTE?');
208400                          READLN; READ(SALIDA[CONT2,CONT3]);
208500                          WHILE (SALIDA[CONT2,CONT3] < 0) OR (SALIDA[CONT2,CONT3]>
208600                              NENTRADAS) DO
208700                              BEGIN
208800                                  WRITELN('LA SALIDA DEBE SER UN 0 Y ',
208900                                      NENTRADAS:1,' VERIFICA');
209000                                  READLN;
209100                                  READ(SALIDA[CONT2,CONT3]);
209200                              END;
209300                          END;
209400                          WRITELN('DESEAS HACER ALGUN OTRO CAMBIO A LA TABLA?');
209500                          READLN; READ(OPCIONC1,OPCIONC2);
209600                          MIENTRAS;
209700                          UNTIL OPCIONC1 = 'N';
209800                          WRITELN('TU TABLA MODIFICADA ES');
209900                          WRITELN;
210000                          TABLA;
210100                      END;
210200                  FOR CONT1:=0 TO NENTRADAS DO
210300                      FOR CONT1:=1 TO NESTADOS DO
210400                          FOR CONT2:=1 TO 3 DO
210500                              EFUT[CONT,CONT1,CONT2]:=EFUTURO[CONT,CONT1,CONT2];
210501
210502

```

```

210503
210504
210600
210700 REPEAT
210800 BANDEX:=FALSE;
210900 FOR CONT:=1 TO NESTADOS DO
211000 BEGIN
211100 BANDERA1:=TRUE;
211200 FOR CONT2:=1 TO NESTADOS DO
211300 IF CONT <> CONT2 THEN
211400 BEGIN
211500 FOR CONT1:=0 TO NENTRADAS DO
211600 IF BANDERA1 THEN
211700 BEGIN
211800 FOR CONT3:=1 TO 3 DO
211900 IF EFUTECONT1,CONT,CONT3 <> EFUTECONT1,CONT2,CONT3
212000 THEN BANDERA1:=FALSE;
212100 IF BANDERA1 THEN
212200 IF SALIDACONT1,CONT <> SALIDACONT1,CONT2 THEN
212300 BANDERA1:=FALSE;
212400 END;
212500 IF BANDERA1 THEN
212600 BEGIN
212700 IF (CONT IN MARCA[CONT2]) AND (CONT2 IN MARCA[CONT2])
212800 THEN BANDERA:=TRUE ELSE BANDERA:=FALSE;
212900 IF NOT BANDERA THEN
213000 BEGIN
213100 IF MARCA[CONT] <> (MARCA[CONT]+[CONT]+[CONT2]) THEN
213200 BANDEX:=TRUE;
213300 MARCA[CONT]:= MARCA[CONT]+[CONT]+[CONT2];
213400 FOR CONT4:=1 TO NESTADOS DO
213500 IF ((MARCA[CONT]+MARCA[CONT4]) = MARCA[CONT]) AND
213600 (CONT <> CONT4) THEN MARCA[CONT]:=MARCA[CONT4];
213700 FOR CONT4:=1 TO NESTADOS DO
213800 FOR CONT5:=0 TO NENTRADAS DO
213900 BEGIN
214000 BANDERA2:=TRUE;
214100 FOR CONT3:=1 TO 3 DO
214200 IF EFUTECONT5,CONT4,CONT3<>EPRESENTE[CONT,CONT3]
214300 THEN BANDERA2:=FALSE;
214400 IF (BANDERA2) AND (MARCA[CONT2] = [ ]) THEN
214500 FOR CONT3:=1 TO 3 DO
214600 EFUTECONT5,CONT4,CONT3:=EPRESENTE[CONT2,CONT3];
214700 END;
214800 END;
214900 BANDERA1:=TRUE;
215000 END;
215100 END;
215200 UNTIL NOT BANDEX;
215300 (* VERIFICA SI HAY REDUCCIONES *)
215400
215500 BANDERA:=TRUE;
215600 FOR CONT:=1 TO NENTRADAS DO
215700 IF MARCA[CONT] <> [ ] THEN
215800 BANDERA:=FALSE;
215900
216000 (* SOLUCION DEL ALUMNO *)
216100
216200 WRITELN('HAY ESTADOS QUE SE PUEDAN REDUCIR?');
216300 READLN;
216400 READ(OPCION[1],OPCION[2]);
216500 MIENTRAS;
216600 IF OPCION[1] = 'N' THEN
216700 IF BANDERA THEN
216701
216702

```



```

222903
222904
223000      ' END;
223100      END;
223200      END;
223300      FOR CONT:=0 TO 1 DO
223400          FOR CONT1:=1 TO NESTADOS DO
223500              BEGIN
223600                  BANDERA:=TRUE;
223700                  FOR CONT2:=1 TO 3 DO
223800                      IF ESTADOC[CONT,CONT2] <> EPRESENTEC[CONT1,CONT2] THEN
223900                          BANDERA:=FALSE;
224000                          IF BANDERA THEN
224100                              NUMEROC[CONT1]=CONT1;
224200              END;
224300          BANDERA:=FALSE;
224400          FOR CONT:=1 TO NESTADOS DO
224500              IF MARCA[CONT] <> [] THEN
224600                  IF ((NUMEROC[] IN MARCA[CONT]) AND (NUMEROC[] IN MARCA[CONT])) THEN
224700                      IF MARCAALUM[CONT] = [] THEN
224800                          BEGIN
224900                              BANDERA:=TRUE;
225000                              MARCAALUM[CONT]=MARCA[CONT];
225100                              CONT1:=CONT;
225200                              CONT:=NESTADOS;
225300                          END
225400                          ELSE
225500                              BEGIN
225600                                  BANDERA:=TRUE;
225700                                  CONT1:=CONT;
225800                              END;
225900                              IF BANDERA THEN
226000                                  WRITELN('OPCION CORRECTA');
226100                              ELSE
226200                                  WRITELN('LA PAREJA QUE PROPONES NO PUEDE REDUCIR ESTADOS, ',
226300                                      'INTENTA NUEVAMENTE');
226400                                  WRITELN('EXISTEN AUN CASOS QUE SE PUE DAN REDUCIR?');
226500                                  READLN; READ(OPCIONC1,OPCIONC2);
226600                                  MIENTRAS;
226700                                  IF OPCIONC1 = 'N' THEN
226800                                      BEGIN
226900                                          BANDERA:=TRUE;
227000                                          FOR CONT:=1 TO NESTADOS DO
227100                                              IF MARCA[CONT] <> [] THEN
227200                                                  IF MARCA[CONT] = MARCAALUM[CONT] THEN
227300                                                      BANDERA:=BANDERA AND TRUE
227400                                                  ELSE
227500                                                      BANDERA:=FALSE;
227600                                                  IF NOT BANDERA THEN
227700                                                      BEGIN
227800                                                          OPCIONC1:='S';
227900                                                          WRITELN('AUN HAY ESTADOS QUE SE REDUCEN, INTENTA HACERLO');
228000                                                      END;
228100                                                  END
228200                                                  ELSE
228300                                                      WRITELN('DAME OTRA PAREJA');
228400                                          UNTIL OPCIONC1 = 'N';
228500                                          BANDEX:=TRUE;
228600                                          BANDERA:=TRUE;
228700                                          (* ELIMINA LOS ESTADOS QUE FUERON ABSORVIDOS POR OTROS *)
228800                                          WHILE BANDERA DO
228900                                              BEGIN
229000                                                  BANDERA:=FALSE;
229100                                                  FOR CONT:=1 TO NESTADOS DO
229101
229102

```



```

229103
229104
229200 IF MARCA[CONT1] <> [] THEN
229300 BEGIN
229400 FOR CONT1:=CONT TO NESTADOS -1 DO
229500 BEGIN
229600 FOR CONT2:=1 TO 3 DO
229700 BEGIN
229800 EPRESENTE[CONT1,CONT2]:=EPRESENTE[CONT1+1,CONT2];
229900 FOR CONT3:=0 TO NENTRADAS DO
230000 EFUT[CONT3,CONT1,CONT2]:=EFUT[CONT3,CONT1+1,CONT2];
230100 END;
230200 MARCA[CONT1]:=MARCA[CONT1+1];
230300 FOR CONT2:=0 TO NENTRADAS DO
230400 SALIDA[CONT2,CONT1]:=SALIDA[CONT2,CONT1+1];
230500 END;
230600 CONT:=NESTADOS;
230700 NESTADOS:=NESTADOS-1;
230800 BANDERA:=TRUE;
230900 END;
231000 END;
231100 (* SE OBTIENE EL NUMERO DE VARIABLES PARA LOS ESTADOS QUE QUEDARON*)
231200 CASE NESTADOS OF
231300 1,2: NVARIA:=1;
231400 3,4: NVARIA:=2;
231500 5,6,7,8: NVARIA:=3;
231600 9,10,11,12,13,14,15,16: NVARIA:=4;
231700 ELSE: NVARIA:=5;
231800 END; (* CASE *)
231900 (* SE OBTIENE EL VALOR BINARIO DE LOS ESTADOS *)
232000 FOR CONT:=1 TO NESTADOS DO
232100 BEGIN
232200 CONT:=CONT-1;
232300 FOR CONT2:=1 TO NVARIA DO
232400 BEGIN
232500 CASE (CONT1 MOD 2) OF
232600 1: EPRESBIN[CONT,CONT2]:='1';
232700 0: EPRESBIN[CONT,CONT2]:='0';
232800 END; (* CASE *)
232900 CONT1:=CONT1 DIV 2
233000 END;
233100 FOR CONT2:=1 TO NESTADOS DO
233200 FOR CONT1:=0 TO NENTRADAS DO
233300 BEGIN
233400 BANDERA:=TRUE;
233500 FOR CONT3:=1 TO 3 DO
233600 IF EFUT[CONT1,CONT2,CONT3] <> EPRESENTE[CONT,CONT3] THEN
233700 BANDERA:=FALSE;
233800 IF BANDERA THEN
233900 FOR CONT3:=1 TO NVARIA DO
234000 EFUTBIN[CONT1,CONT2,CONT3]:=EPRESBIN[CONT,CONT3];
234100 END;
234200 END;
234300 CASE NENTRADAS OF
234400 0,1: NENTRA:=1;
234500 2,3: NENTRA:=2;
234600 ELSE: NENTRA:=3;
234700 END; (* CASE *)
234800 (* LLENA LOS ESTADOS NO DEFINIDOS CON NO IMPORTA *)
234900 FOR CONT:= (NESTADOS+1) TO EXPONENTE(NVARIA) DO
235000 BEGIN
235100 FOR CONT1:=1 TO NVARIA DO
235200 BEGIN
235300 EPRESBIN[CONT,CONT1]:='*';
235301
235302

```

```

235303
235304
235400     FOR CONT2:=0 TO NENTRADAS DO
235500         EFUTBINECONT2,CONT,CONT1J:='*';
235600     END;
235700     FOR CONT1:=1 TO 3 DO
235800         BEGIN
235900             EPRESENTE[CONT,CONT1J:='*';
236000             FOR CONT2:=0 TO NENTRADAS DO
236100                 EFUT[CONT2,CONT,CONT1J:='*';
236200             END;
236300         END;
236400     SIMBOLO.LETRAC1J:='E';
236500     FOR CONT:=2 TO NENTRA DO
236600         SIMBOLO.LETRACONTJ:=SUCC(SIMBOLO.LETRACCONT-1J);
236700     SIMBOLO.LETRACNENTRA1J:='W';
236800     FOR CONT:=NENTRA+2 TO (NVARIA+NENTRA) DO
236900         SIMBOLO.LETRACONTJ:=SUCC(SIMBOLO.LETRACCONT-1J);
237000     WRITELN; WRITELN('A CONTINUACION TE PRESENTO TU TABLA REDUCIDA');
237100     WRITELN;
237200     WRITE('!-----');
237300     FOR CONT:=1 TO NVARIA DO
237400         WRITE('-');
237500     FOR CONT1:=0 TO NENTRADAS DO
237600         BEGIN
237700             WRITE('!-----');
237800             FOR CONT:=1 TO NVARIA DO
237900                 WRITE('-');
238000         END;
238100     WRITELN('!');
238200     WRITE('!'); WRITE('!S');
238300     FOR CONT:=1 TO NVARIA DO WRITE(' ');
238400     FOR CONT1:=0 TO NENTRADAS DO
238500         BEGIN
238600             WRITE('! E=',CONT1:1,' ');
238700             FOR CONT:=1 TO NVARIA DO
238800                 WRITE(' ');
238900         END;
239000     WRITELN('!');
239100     WRITE('! DP ');
239200     FOR CONT1:=1 TO NVARIA DO
239300         WRITE(' ');
239400     FOR CONT1:=0 TO NENTRADAS DO
239500         BEGIN
239600             WRITE('! QF ');
239700             FOR CONT1:=1 TO NVARIA DO
239800                 WRITE(' ');
239900             WRITE('! S ');
240000     END; WRITELN('!');
240100     WRITE('!-----');
240200     FOR CONT:=1 TO NVARIA DO
240300         WRITE('-');
240400     FOR CONT1:=0 TO NENTRADAS DO
240500         BEGIN
240600             WRITE('!-----');
240700             FOR CONT:=1 TO NVARIA DO
240800                 WRITE('-');
240900         END;
241000     WRITELN('!');
241100     WRITE('! ');
241200     FOR CONT:=(NVARIA+NENTRA) DOWNT0 (NENTRA+1) DO
241300         WRITE(SIMBOLO.LETRACONTJ:1);
241400     FOR CONT:=0 TO NENTRADAS DO
241500         BEGIN
241501
241502

```

```

241503
241504
241600 WRITE('! ');
241700 FOR CONT1:=(NVARIA+NENTRA) DOWNT0 (NENTRA+1) DO
241800 WRITE(SIMBOLO.LETRACCONT1);
241900 WRITE('! ');
242000 END;
242100 WRITELN('!');
242200 FOR CONT:=1 TO EXPONENTE(NVARIA) DO
242300 BEGIN WRITE('! ');
242400 FOR CONT2:=1 TO 3 DO
242500 WRITE(EPRESENTECCONT,CONT2); WRITE(' ');
242600 FOR CONT2:=NVARIA DOWNT0 1 DO
242700 WRITE(EPRESBINCCONT,CONT2);
242800 FOR CONT1:=0 TO NENTRADAS DO
242900 BEGIN WRITE('! ');
243000 FOR CONT2:=1 TO 3 DO
243100 WRITE(EFUTCCONT1,CONT,CONT2); WRITE(' ');
243200 FOR CONT2:=NVARIA DOWNT0 1 DO
243300 WRITE(EFUTBINCCONT1,CONT,CONT2);
243400 IF CONT > NENTRADOS THEN
243500 WRITE('! * ');
243600 ELSE
243700 WRITE('! ',SALIDAICONT1,CONT2;1,' ');
243800 END;
243900 WRITELN('!');
244000 END;
244100 WRITE('!-----');
244200 FOR CONT:=1 TO NVARIA DO
244300 WRITE('!-');
244400 FOR CONT1:=0 TO NENTRADAS DO
244500 BEGIN
244600 WRITE('!-----');
244700 FOR CONT:=1 TO NVARIA DO
244800 WRITE('!-');
244900 END;
245000 WRITELN('!');
245100 POTENCIA:=EXPONENTE(NVARIA+NENTRA);
245200 WRITELN('CONTINUA...'); READLN;
245300 WRITELN; WRITELN('PUEDES IMPLEMENTAR TU CIRCUITO CON:');
245400 WRITELN(' 1 : J-K');
245500 WRITELN(' 2 : S-R');
245600 WRITELN(' 3 : D');
245700 WRITELN(' 4 : T');
245800 WRITELN('CON CUAL DE ELLOS DESEAS HACERLO?');
245900 READLN;
246000 READ(OPCIONC1);
246100 WHILE (OPCIONC1 < '1') OR (OPCIONC1 > '4') DO
246200 BEGIN
246300 WRITELN('VERIFICA LA OPCION');
246400 READLN;
246500 READ(OPCIONC1);
246600 END;
246700 DEFHV(NVARIA+NENTRA,NJ,NI);
246800 FOR CONT:=0 TO NI-1 DO
246900 FOR CONT2:=0 TO NJ-1 DO
247000 MAPASICONT,CONT2:=NJ*CONT + CONT2;
247100 IF NJ > 2 THEN
247200 BEGIN
247300 HORIZONTAL2(MAPAS);
247400 IF NJ > 4 THEN
247500 BEGIN
247600 HORIZONTAL4(MAPAS);
247700 VERTICAL2(MAPAS);
247701
247701

```

```

247703
247704
247800 END
247900 ELSE IF NI > 2 THEN
248000     VERTICAL2(MAPAS);
248100 IF NI > 4 THEN
248200     VERTICAL4(MAPAS);
248300 END;
248400 BANDERA:=FALSE;
248500 MAPAORIG;
248600 WRITELN;
248700 WRITELN('EL MAPA CON TODAS LAS POSICIONES POSIBLES EN FUNCION ',
248800     'DEL NUMERO DE');
248900 WRITELN('ESTADOS QUE RESULTARON Y EL NUMERO DE ENTRADAS ES:');
249000 GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
249100 CASE OPCIONC1J OF
249200     '1': BEGIN (* IMPLEMENTACION DEL CIRCUITO CON JK *)
249300         TEXTOK;
249400         FOR CONT:=1 TO NVARIA DO
249500             BEGIN
249600                 WRITELN;
249700                 WRITELN('DE LA TABLA, PARA LA FUNCION ',SIMBOLO.LETRACNVARIA+
249800                     '+NENTRA-CONTJ:1); WRITELN;
249900                 BANDJS:=TRUE;
250000                 MAPAORIG;
250100                 LLENAMAPA(CONT,'1');
250200                 BANDERA2:=FALSE;
250300                 MAPAS01;
250400                 WRITELN('EL MAPA CON LOS MINTERMINOS CORRESPONDIENTES ',
250500                     'SE PRESENTA A LA DERECHA'); BANDERA:=TRUE;
250600                 GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
250700                 NDIMPORTA(CONT,1);
250800                 CUBNOIMPORTA(CONT,'1');
250900                 OPCIONC2J:='J';
251000                 RESUELVEAL(CONT);
251100                 (* SOLUCION DE K*)
251200                 BANDJS:=FALSE;
251300                 WRITELN;
251400                 WRITELN('AHORA ANALIZAREMOS LA SOLUCION PARA K'); WRITELN;
251500                 MAPAORIG;
251600                 LLENAMAPA(CONT,'1');
251700                 BANDERA2:=FALSE;
251800                 MAPAS01;
251900                 WRITELN('EL MAPA CON LOS MINTERMINOS CORRESPONDIENTES ',
252000                     'SE PRESENTA A LA DERECHA');
252100                 BANDERA:=TRUE;
252200                 GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
252300                 LLENAMAPA(CONT,'0');
252400                 NDIMPORTA(CONT,0);
252500                 BANDERA2:=TRUE;
252600                 CUBNOIMPORTA(CONT,'0');
252700                 OPCIONC2J:='K';
252800                 RESUELVEAL(CONT);
252900             END;
253000         END;
253100     '2': BEGIN
253200         TEXTOSR;
253300         FOR CONT:=1 TO NVARIA DO
253400             BEGIN
253500                 WRITELN;
253600                 WRITELN('ANALISIS DE LA FUNCION ',SIMBOLO.LETRACNVARIA+
253700                     '+NENTRA-CONTJ:1); WRITELN;
253800                 BANDJS:=TRUE;
253900                 MAPAORIG;
253901
253902

```

```

253903
253904
254000      LLENAMAPA(CONT,'1');
254100      BANDERA2:=FALSE;
254200      MAPAS01;
254300      WRITELN('EL MAPA CON LOS MINTERMINOS CORRESPONDIENTES ',
254400              'SE PRESENTA A LA DERECHA');
254500      BANDERA1:=TRUE;
254600      GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
254700      BANDERA2:=TRUE;
254800      NOIMPORSR(CONT,1);
254900      CUBNOIHPORTA(CONT,'1');
255000      OPCIONC2J:='S';
255100      RESUELVEAL(CONT);
255200      (* R *) MAPADORIG;
255300      WRITELN;
255400      WRITELN('AHORA ANALIZAREMOS LA SOLUCION PARA R'); WRITELN;
255500      BANDJS:=FALSE;
255600      LLENAMAPA(CONT,'1');
255700      BANDERA2:=FALSE;
255800      MAPAS01;
255900      BANDERA:=TRUE;
256000      WRITELN('EL MAPA CON LOS MINTERMINOS CORRESPONDIENTES ',
256100              'SE PRESENTA A LA DERECHA');
256200      GRAFICA(MAPAS,MAPASMIN,CMAPASMIN);
256300      LLENAMAPA(CONT,'0');
256400      NOIMPORSR(CONT,0);
256500      CUBNOIHPORTA(CONT,'0');
256600      OPCIONC2J:='R';
256700      RESUELVEAL(CONT);
256800      END;
256900      END;
257000      '3': BEGIN
257100          WRITELN('PARA UN FLIP FLOP D, LA SALIDA DESPUES DE UN ',
257200                  'PULSO DE RELOJ ES IGUAL');
257300          WRITELN('A LA ENTRADA D EN EL TIEMPO DE RELOJ. ');
257400          WRITELN('POR LO CUAL SE ANALIZAN LOS MAPAS DE CADA VA',
257500                  'RIABLE DE LA REPRESENTA');
257600          WRITELN('CION BINARIA DE LOS ESTADOS PARA TODAS LAS EN',
257700                  'TRADAS. ');
257800          FOR CONT:=1 TO NVARIA DO
257900              BEGIN
258000                  WRITELN;
258100                  WRITELN('ANALISIS PARA EL D',CONT:1);
258200                  WRITELN;
258300                  BANDJS:=TRUE;
258400                  MAPADORIG;
258500                  LLENAMAPA(CONT,'1');
258600                  FOR CONT1:=0 TO NDIGITOS DO
258700                      CASI[CONT1]:=DIGITOS[CONT1];
258800                  NCASI:=NDIGITOS;
258900                  BANDERA2:=FALSE;
259000                  MAPAS01;
259100                  BANDERA2:=TRUE;
259200                  CUBNOIHPORTA(CONT,'1');
259300                  RESUELVEAL(CONT);
259400              END;
259500          END;
259600      '4': BEGIN
259700          WRITELN('EN UN FLIP FLOP *T*, LA ENTRADA DE RELOJ ES LA ',
259800                  'UNICA EXISTENTE, ASI');
259900          WRITELN('CUANDO HAY UN PULSO, EL FLIP FLOP CAMBIA DE ES',
260000                  'TADO. ');
260100          WRITELN('POR LO CUAL SE ANALIZAN LOS MAPAS DE CADA VARIA',
260101
260102

```

```

-----
260103
260104
260200      'BLE DE LA REPRESENTACION');
260300 WRITELN('BINARIA DE LOS ESTADOS OBSERVANDO SI HUBO UN ',
260400 ' CAMBIO DE ESTADO DE');
260500 WRITELN('VARIABLE(T) A VARIABLE(T+1) PARA CADA ENTRADA, ',
260600 'SI ES ASI, HUBO UN PULSO');
260700 WRITELN('Y SE INDICA "1" EN LA POSICION CORRESPONDIENTE;',
260800 ' EN CASO CONTRARIO CON "0"');
260900     FOR CONT:=1 TO NVARIA DO
261000     BEGIN
261100         WRITELN; WRITELN('ANALISIS PARA EL TOGGLE ',CONT:1);
261200         WRITELN;
261300         OPCIONC2J:='T';
261400         BANDJS:=TRUE;
261500         MAPAORIG;
261600         LLENATOGGLE;
261700         BANDERA2:=FALSE;
261800         MAPASO1;
261900         BANDERA2:=TRUE;
262000         FOR CONT1:=0 TO NDIGITOS DO
262100             CASICCONT1J:=DIGITOSECONT1J;
262200             NCASI:=NDIGITOS;
262300             CUBNOIMPORTA(CONT,'1');
262400             RESUELVEAL(CONT);
262500         END;
262600     END;
262700 END;
262800 WRITELN; WRITELN('MAPA PARA LA FUNCION DE SALIDA'); WRITELN;
262900 OPCIONC1J:='3';
263000 BANDJS:=FALSE;
263100 BANDEX:=FALSE;
263200 MAPAORIG;
263300 LLENAMAPA(CONT,'1');
263400 BANDERA2:=FALSE;
263500 FOR CONT1:=0 TO NDIGITOS DO
263600     CASICCONT1J:=DIGITOSECONT1J;
263700     NCASI:=NDIGITOS;
263800     MAPASO1;
263900     CUBNOIMPORTA(CONT,'1');
264000     RESUELVEAL(CONT);
264100 END.
#

```

R SECUENCIALES

#RUNNING 5617

ESTE PROGRAMA TE GUIARA EN LA MINIMIZACION DE ESTADOS PARA CIRCUITOS SECUENCIALES, DESEAS CONOCER UNA PEQUEÑA INTRODUCCION DE COMO LO HACE ESTE PROGRAMA?

#?

SI

LA MINIMIZACION QUE PLANTEA ES LA MAS SENCILLA (METODO DE PARTICION). UN ESTADO ES ABSORVIDO POR OTRO SI TODOS SUS ESTADOS FUTUROS Y SUS SALIDAS CORRESPONDIENTES SON IGUALES.

PARA UNA MINIMIZACION MAS FORMAL, EXISTEN OTROS METODOS, UNO DE ELLOS CONOCIDO COMO METODO DE TABLA DE IMPLICACION, EL CUAL PLANTEA UNA SERIE DE TABLAS Y BUSQUEDAS QUE ES PREFERIBLE DESARROLLAR EN PAPEL DADO LO LABORIOSO QUE RESULTA.

PARA CONSULTAR ESTE METODO PUEDES CONSULTAR LOS SIGUIENTES TEXTOS:

- 1) TEORIA DE CONMUTACION Y DISEÑO LOGICO  
FREDERICK J. HILL
- 2) LOGICA DIGITAL Y DISEÑO DE COMPUTADORES  
M. MORRIS MAND

PARA OBTENER LA SOLUCION EN FUNCION DE CUALQUIER FLIP FLOP (JK,SR,D,T) REQUIERE:

LA REPRESENTACION BINARIA DE LOS ESTADOS PRESENTES REDUCIDOS Y LOS FUTUROS PARA TODAS LAS SALIDAS.

SE IRAN OBTENIENDO LOS MAPAS DE KARNAUGH PARA CADA UNA DE LAS POSICIONES DE LA REPRESENTACION BINARIA.

EJEMPLO:

CONTINUA...

EPRESENTE	EFUT E=0	EFUT E=1
ML	ML	ML
A 00	A 00	B 01
B 01	A 00	C 10
C 10	B 01	B 01

DONDE:

M,L: POSICIONES BINARIAS DE LOS ESTADOS

A,B,C: NOMBRES ASIGNADOS A LOS ESTADOS

REACOMODANDO

	Q(T)			Q(T+1)	
	E	M	L	M	L
0	0	0	0	0	0
1	0	0	1	0	0
2	0	1	0	0	1
3	0	1	1	*	*
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	0	1
7	1	1	1	*	*

CONTINUA...

LLENANDO EL MAPA PARA L(T+1):

E\ML 00 01 11 10

| 0 | 0 | \* | 1 |

| 1 | 0 | \* | 1 |

SERA NUESTRA HERRAMIENTA A LA CUAL APLICAREMOS REGLAS ESPECIALES PARA IMPLEMENTAR EL CIRCUITO DEPENDIENDO DEL FLIP FLOP QUE USEMOS.

CUANTOS ESTADOS TIENE TU TABLA?

4

CUANTAS ENTRADAS TIENE TU TABLA?

2

DEFINE CADA ESTADO CON UN IDENTIFICADOR NO MAYOR DE 3 CARACTERES.

EL ESTADO 0 SE LLAMA

A

EL ESTADO 1 SE LLAMA

B

EL ESTADO 2 SE LLAMA

C

EL ESTADO 3 SE LLAMA

D

TUS ENTRADAS ESTAN NUMERADAS DEL 0 AL 1

CUAL ES EL ESTADO EN T+1 DEL ESTADO A PARA UNA ENTRADA =0

B

CUAL ES LA SALIDA CORRESPONDIENTE

0

CUAL ES EL ESTADO EN T+1 DEL ESTADO A PARA UNA ENTRADA =1

E

EL ESTADO NO EXISTE, VERIFICA

A

CUAL ES LA SALIDA CORRESPONDIENTE

0

CUAL ES EL ESTADO EN T+1 DEL ESTADO B PARA UNA ENTRADA =0

C

CUAL ES LA SALIDA CORRESPONDIENTE

0

CUAL ES EL ESTADO EN T+1 DEL ESTADO B PARA UNA ENTRADA =1

D

CUAL ES LA SALIDA CORRESPONDIENTE

0

CUAL ES EL ESTADO EN T+1 DEL ESTADO C PARA UNA ENTRADA =0

A

CUAL ES LA SALIDA CORRESPONDIENTE

1

CUAL ES EL ESTADO EN T+1 DEL ESTADO C PARA UNA ENTRADA =1

D

CUAL ES LA SALIDA CORRESPONDIENTE

1

CUAL ES EL ESTADO EN T+1 DEL ESTADO D PARA UNA ENTRADA =0

C



CUAL ES LA SALIDA CORRESPONDIENTE

0

CUAL ES EL ESTADO EN T+1 DEL ESTADO D PARA UNA ENTRADA =1

D

CUAL ES LA SALIDA CORRESPONDIENTE

0

LA TABLA QUE SE ORIGINA ES

SIGNIFICADO DE VARIABLES:

QF : ESTADO PRESENTE

QF : ESTADO FUTURO PARA EL NUMERO DE ENTRADAS

E : ENTRADA

S : SALIDA

		E=0		E=1	
QF	QF	S	QF	S	QF
A	B	0	A	0	
B	C	0	D	0	
C	A	1	D	1	
D	C	0	D	0	

DESEAS HACER ALGUN CAMBIO A LA TABLA?

SI

CUAL ES EL ESTADO PRESENTE DONDE DESEAS HACER LA MODIFICACION?

C

EL ESTADO NO EXISTE, VERIFICA

C

CUAL ES EL ESTADO EN T+1 DE ESTE ESTADO PARA ENTRADA =0

A

CUAL ES LA SALIDA CORRESPONDIENTE?

1

CUAL ES EL ESTADO EN T+1 DE ESTE ESTADO PARA ENTRADA =1

D

CUAL ES LA SALIDA CORRESPONDIENTE?

0

DESEAS HACER ALGUN OTRO CAMBIO A LA TABLA?

NO

TU TABLA MODIFICADA ES

SIGNIFICADO DE VARIABLES:

QF : ESTADO PRESENTE

QF : ESTADO FUTURO PARA EL NUMERO DE ENTRADAS

E : ENTRADA

S : SALIDA

		E=0		E=1	
QF	QF	S	QF	S	QF
A	B	0	A	0	
B	C	0	D	0	
C	A	1	D	1	
D	C	0	D	0	

HAY ESTADOS QUE SE PUEDAN REDUCIR?

SI

DA ME POR PAREJAS LOS ESTADOS QUE SE REDUCEN SEPARADOS POR (,); SI PROPORCIONAS MAS DE DOS ESTADOS, SERAN IGNORADOS

A,B  
LA PAREJA QUE PROPONES NO PUEDE REDUCIR ESTADOS, INTENTA NUEVAMENTE  
EXISTEN AUN CASOS QUE SE PUEDAN REDUCIR?

SI  
DA ME OTRA PAREJA  
B,D  
OPCION CORRECTA  
EXISTEN AUN CASOS QUE SE PUEDAN REDUCIR?  
NO

A CONTINUACION TE PRESENTO TU TABLA REDUCIDA

QP	E=0			E=1		
	QF	S		QF	S	
	XW	XW		XW		
A	00	D 10	0	A 00	0	0
C	01	A 00	1	D 10	0	0
D	10	C 01	0	D 10	0	0
*** **	*** **	*		*** **	*	

CONTINUA...

PUEDES IMPLEMENTAR TU CIRCUITO CON:

- 1 : J-K
- 2 : S-R
- 3 : D
- 4 : T

CON CUAL DE ELLOS DESEAS HACERLO?

1

EL MAPA CON TODAS LAS POSICIONES POSIBLES EN FUNCION DEL NUMERO DE ESTADOS QUE RESULTARON Y EL NUMERO DE ENTRADAS ES:

X\ WE

1 0 1 3 2

4 5 7 6

CONTINUA...

PARA IMPLEMENTAR UN CIRCUITO CON FLIP FLOP JK, YA CON EL MAPA DEFINIDO PARA LOS ESTADOS FUTUROS DE CADA FUNCION, USAREMOS UN METODO MUY SENCILLO:

PARA J:

SE HACEN NO IMPORTA TODAS LAS POSICIONES DONDE LA <VARIABLE> ANALIZADA ES 1 Y SE RESUELVE EL MAPA CON LOS 1 RESTANTES.

PARA K:

SE HACEN NO IMPORTA TODAS LAS POSICIONES DONDE LA <VARIABLE> ANALIZADA Y SE RESUELVE EL MAPA CON LOS 0 RESTANTES.

SIGUIENDO ESTAS SENCILLAS REGLAS, RESUELVE TU CIRCUITO  
CONTINUA...

DE LA TABLA, PARA LA FUNCION X

EL MAPA CON LOS MINTERMINOS CORRESPONDIENTES SE PRESENTA A LA DERECHA

X \ WE	X \ WE
0   1   3   2	1   0   1   0
4   5   7   6	0   1   *   *

CONTINUA...

PARA J(X) APLICANDO EL METODO, SE GENERA EL MAPA SIGUIENTE

X \ WE	X \ WE
0   1   3   2	1   0   1   0
4   5   7   6	*   *   *   *

CONTINUA...

DE ACUERDO AL MAPA ANTERIOR, PROPORCIONA LOS MINTERMINOS QUE SE PUEDEN COMBINAR, CADA UNO SEPARADO POR (,) Y TERMINAR CON (?)

EXISTEN MINTERMINOS QUE SE COMBINEN?

SI

PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA

0,4;

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

X \ WE	X \ WE
0   1   3   2	1X   0   1   0
4   5   7   6	*X   *   *   *

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINEN?

SI

PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA

3,7;

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

X \ WE	X \ WE
0   1   3   2	1X   0   1X   0
4   5   7   6	*X   *   *X   *

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINEN?  
NO  
TU SOLUCION ES:  
W0E0+W1E1

PERFECTO, OBTUVISTE LA SOLUCION OPTIMA

AHORA ANALIZAREMOS LA SOLUCION PARA K

EL MAPA CON LOS MINTERMINOS CORRESPONDIENTES SE PRESENTA A LA DERECHA

X\ WE	X\ WE
-----	-----
0   1   3   2	1   0   1   0
-----	-----
4   5   7   6	0   1   *   *
-----	-----

CONTINUA...

PARA K(X) APLICANDO EL METODO, SE GENERA EL MAPA SIGUIENTE

X\ WE	X\ WE
-----	-----
0   1   3   2	*   *   *   *
-----	-----
4   5   7   6	0   1   *   *
-----	-----

CONTINUA...

DE ACUERDO AL MAPA ANTERIOR, PROPORCIONA LOS MINTERMINOS QUE SE PUEDEN  
COMBINAR, CADA UNO SEPARADO POR (,) Y TERMINAR CON (,)

EXISTEN MINTERMINOS QUE SE COMBINEN?

SI  
PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA  
0,4;

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

X\ WE	X\ WE
-----	-----
0   1   3   2	*X  *   *   *
-----	-----
4   5   7   6	0X  1   *   *
-----	-----

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINEN?

NO

TU SOLUCION ES:  
WEO

TU SOLUCION ES CORRECTA, PERO NO LA OPTIMA DESEAS INTENTAR LLEGAR A ELLA?  
NO  
DESEAS CONOCER ANTES LA SOLUCION OPTIMA?  
SI  
EO

DE LA TABLA, PARA LA FUNCION W

EL MAPA CON LOS MINTERMINOS CORRESPONDIENTES SE PRESENTA A LA DERECHA

X \ WE	X \ WE
0   1   3   2	0   0   0   0
4   5   7   6	1   0   *   *

CONTINUA...

PARA J(W) APLICANDO EL METODO, SE GENERA EL MAPA SIGUIENTE

X \ WE	X \ WE
0   1   3   2	0   0   *   *
4   5   7   6	1   0   *   *

CONTINUA...

DE ACUERDO AL MAPA ANTERIOR, PROPORCIONA LOS MINTERMINOS QUE SE PUEDEN COMBINAR, CADA UNO SEPARADO POR (,) Y TERMINAR CON (;)

EXISTEN MINTERMINOS QUE SE COMBINEN?

SI

PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA 4,6;

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

X \ WE	X \ WE
0   1   3   2	0   0   *   *
4   5   7   6	1X   0   *   *X

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINEN?

NO

TU SOLUCION ES:  
X1E0

PERFECTO, OBTUVISTE LA SOLUCION OPTIMA  
AHORA ANALIZAREMOS LA SOLUCION PARA K

EL MAPA CON LOS MINTERMINOS CORRESPONDIENTES SE PRESENTA A LA DERECHA

X\ WE	X\ WE
1 0 1 1 3 1 2 1	1 0 1 0 1 0 1 0 1
1 4 1 5 1 7 1 6 1	1 1 1 0 1 * 1 * 1

CONTINUA...

PARA K(W) APLICANDO EL METODO, SE GENERA EL MAPA SIGUIENTE

X\ WE	X\ WE
1 0 1 1 3 1 2 1	1 * 1 * 1 0 1 0 1
1 4 1 5 1 7 1 6 1	1 * 1 * 1 * 1 * 1

CONTINUA...

DE ACUERDO AL MAPA ANTERIOR, PROPORCIONA LOS MINTERMINOS QUE SE PUEDEN  
COMBINAR, CADA UNO SEPARADO POR (,) Y TERMINAR CON (;)

EXISTEN MINTERMINOS QUE SE COMBINENT

SI  
PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA  
0;1;2;3;4;5;6;7;

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

X\ WE	X\ WE
1 0 1 1 3 1 2 1	1 *X1 *X1 0X1 0X1
1 4 1 5 1 7 1 6 1	1 *X1 *X1 *X1 *X1

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINENT

NO

TU SOLUCION ES:  
UNA TAUTOLOGIA, ES DECIR, 1

PERFECTO, OBTUVISTE LA SOLUCION OPTIMA

MAPA PARA LA FUNCION DE SALIDA

X\ WE	X\ WE
1 0 1 1 3 1 2 1	1 0 1 0 1 0 1 1 1
1 4 1 5 1 7 1 6 1	1 0 1 0 1 * 1 * 1

CONTINUA...

DE ACUERDO AL MAPA ANTERIOR, PROPORCIONA LOS MINTERMINOS QUE SE PUEDE  
COMBINAR, CADA UNO SEPARADO POR (,) Y TERMINAR CON (,)

EXISTEN MINTERMINOS QUE SE COMBINEN?

SI

PROPORCIONA LOS MINTERMINOS QUE SE COMBINAN EN LA FORMA INDICADA

2,6,

BIEN, LOS MINTERMINOS PROPUESTOS SI SE COMBINAN

X\ WE

X\ WE

-----  
| 0 | 1 | 3 | 2 |  
-----

-----  
| 0 | 0 | 0 | 1X|  
-----

-----  
| 4 | 5 | 7 | 6 |  
-----

-----  
| 0 | 0 | \* | \*X|  
-----

CONTINUA...

AUN HAY MINTERMINOS QUE SE COMBINEN?

NO

TU SOLUCION ES:

W1E0

PERFECTO, OBTUVISTE LA SOLUCION OPTIMA

#ET=25:12.2 PT=7.0 IQ=0.2

### VII.3 MANUAL DE USUARIO

El comando necesario para correr este programa dependera de la maquina que se utilice, por lo que debera investigarse en el lugar correspondiente.

Siempre que se conteste algo a la maquina, debe terminarse presionando la tecla <RETURN>. Si se trata de una decision (SI o NO) deberan proporcionarse los dos caracteres.

Al iniciarse la corrida de este programa, el usuario tendra la oportunidad de informarse.

Enseguida pedira el numero de estados presentes de la tabla, este debe ser un entero no mayor de 16 ni menor de 2. A continuacion, solicitara el numero de entradas de la misma, este debe ser un entero entre 1 y 5.

Cuando el programa requiere el nombre de los estados presentes, se proporcionara un maximo de 3 caracteres alfanumericos para identificar cada uno de ellos.

Los estados futuros se pediran en la misma forma (tres caracteres alfanumericos) indicando a que entrada se referira. Asimismo, se pedira la salida correspondiente.

Cuando se han dado todos estos datos, se tiene construida la tabla de estados y se presenta al usuario, dandole oportunidad de hacer alguna correccion si es necesario, para lo cual pedira el nombre del estado presente que se debe modificar y a continuacion los estados futuros y salidas de ese estado. Terminadas las correcciones, presentara nuevamente la tabla, pero ahora modificada.

Si la tabla de estados propuesta no se pueda reducir, el usuario tendra dos opciones: dar otra tabla con estados que si se reduzcan, o bien, dar por terminado el proceso.

Con una tabla que se pueda reducir, continua el proceso para efectuar esta reduccion, pidiendo por parejas los estados que se reducen separados por una coma (,).



Una vez verificados los estados que se reducen, se muestra el mapa de Karnaugh acotado en función del nuevo número de estados y el número de entradas, mostrando todas las posibles minterminos que se pueden tener para ese mapa.

Las entradas tendrán asociada la variable E para identificarlas, en el mapa de Karnaugh se usan las letras del abecedario que suceden a esta variable si el número de entradas es mayor de 2 debido a que con una sola variable solo es posible definir hasta dos estados.

El usuario podrá elegir entre los Flip Flops JK, SR D y T para implementar su circuito.

Para la minimización de las funciones, se usa el método de Mapas de Karnaugh (cuyo manejo se describe posteriormente), llenando para el usuario los mapas correspondientes dependiendo del Flip Flop elegido.

1.- Para el Flip Flop D:

$$D(t+1) = D(t)$$

Se ubican en el mapa los valores binarios de la variable que se analiza, en las posiciones correspondientes y se resuelve el mapa.

2.- Para el Flip Flop T:

T	T(t+1)
0	1
1	0

Si en la transición del estado presente al estado futuro hay un cambio en la variable que se analiza, se obtiene un mintermino para el mapa y se ubica en la posición correspondiente. Verificados todos los estados, se procede a resolver el mapa.

Para los Flip Flops JK y SR, se evita el manejo de las tablas de transición, traduciéndolas a algoritmos sencillos como se describe a continuación:

- Se obtiene un primer mapa ubicando los valores binarios de la variable que se analiza en las posiciones correspondientes (Flip Flop D), el cual se modificara

dependiendo de cualquiera de los casos siguientes:

Se tomara como ejemplo de este primer mapa:

$$E \setminus YX$$

	0	1	0	1
0	1	1	0	1
1	0	0	1	1

Para JK:

Para analizar J(X), se hacen no importa todos los estados donde la variable es cierta y su analiza el mapa con los "1"s restantes.

Aplicando el algoritmo para el mapa ejemplo, se obtiene:

$$E \setminus YX$$

	0	*	*	1
0	*	*	1	
1	0	*	*	1

considerando "1" los "\*" que convienen para efectos de minimizacion, la solucion seria:

$$J(X) = Y$$

Para analizar K(X), se hacen no importa todos los estados donde la variable es nesada y su analiza el mapa con los "0"s restantes.

Aplicando el algoritmo para el mapa ejemplo, se obtiene:

$$E \setminus YX$$

	*	*	0	1
0	*	*	1	1
1	*	*	1	1

considerando "0" los "\*" que convienen, la solucion seria:

$$K(X) = \overline{X}E$$

Para SR:

Si se analiza  $S(X)$ , se hacen no importa los minterminos del mapa donde la variable es cierta y se analiza con los "1" restantes.

Aplicando este algoritmo para el mapa ejemplo, se obtiene:

E\YX					
	1	0	*	0	1
	1	0	0	*	1

considerando "1" los "\*" que convienen, la solución sería:

$$S(X) = Y\bar{X}$$

Si se analiza  $R(X)$ , se hacen no importa los "0" del mapa en donde la variable es negada y se analiza con los "0" restantes.

aplicando este algoritmo para el mapa ejemplo, se obtiene:

E\YX					
	1	*	1	0	1
	1	*	0	1	1

considerando "0" los "\*" que convienen, la solución sería:

$$R(X) = \bar{E}YX + E\bar{Y}$$

La Justificación de estos algoritmos se encuentra en la Sección VII.1.

Una vez obtenida la lógica necesaria para los Flip Flops, se procederá a obtener la correspondiente para la salida, para lo cual se llena el mapa de Karnaugh con las salidas de los estados futuros.

En la etapa de solución, se presentarán siempre dos mapas, el de la izquierda sigue representando todos los posibles minterminos para la función y el de la derecha, se lleno de

acuerdo al algoritmo del Flip Flop con que se diseñe el circuito, o bien, con las salidas de los estados futuros.

Siempre que se pida una decision al usuario (SI o NO), se deberan dar los dos caracteres.

A continuacion preguntara si existen minterminos que se combinen, en caso afirmativo, los pedira, estos deberan darse en la forma decimal ya indicada.

Validara la respuesta e indicara si es correcta o no.

En el primer caso, ademas presentara los dos mapas, indicando en el de la derecha, los minterminos que combino con el caracter 'X'.

Preguntara si aun existen minterminos por combinar, en caso afirmativo, repetira el proceso anterior; de otra manera pedira saber si hubo minterminos que no se pudieran combinar, si los hay, se debera dar 1 por vez en forma decimal acompañado al final por 'i'.

Al terminar de cubrir todos los minterminos, se presentara la solucion que obtuvo el usuario indicandole si es optima o no. Si no lo es se tiene la opcion de tratar de llegar a ella reinicializando el proceso, o bien, solo de conocer la solucion optima.

La solucion sera mostrada en funcion de las variables del mapa, todas ellas seran acompañadas de '0' si la variable es negada o de '1', si no lo es.

Si no se reinicializa el proceso, continuara con la minimizacion correspondiente.

Las expresiones que minimizan la logica combinacional para los Flip Flops y la salida, se muestran en el momento que se termino el analisis de cada uno de ellos. Estas expresiones se definen en funcion de las variables de los mapas, se les asocia '1' para indicar que la variable es verdadera, o '0' si es falsa.

Quando se concluye el analisis de la funcion de salida:  
Proceso concluye.

### VIII. CONCLUSIONES

La minimización de funciones de conmutación, requiere de un adiestramiento práctico, con objeto de que el diseñador logre visualizar fácilmente la implementación física de dichas funciones.

Para que el usuario logre este objetivo, se creó el sistema que se presenta en esta tesis, en el que tendrá un confiable auxiliar.

El sistema se podrá considerar propiamente como dos sistemas totalmente independientes, en función de los lenguajes que se utilizaron.

Debido a la característica de transportabilidad que tiene PASCAL, ya que puede encontrarse en la mayoría de micro hasta macro computadoras, se consideró adecuado elaborar todo el sistema en este lenguaje.

FORTRAN 77 no tiene la misma versatilidad, por lo que solo dos de los programas (MAPAS DE KARNAUGH y QUINE MC.CUSKEY) que forman el sistema, se implementaron en este lenguaje. No obstante, se enfatizó en el trabajo del manejo de la pantalla, como puede observarse en las corridas ejemplo que acompañan este trabajo. Se consideró práctico establecer el mínimo de modificaciones para el caso de su transportación a otro equipo de cómputo con diferente equipo de entrada/salida (terminal de video).

Para el sistema creado en PASCAL, se tomaron en cuenta las restricciones que se imponen en algunos equipos de cómputo:

- Número de caracteres reconocibles de un identificador.
- Número máximo de dimensiones aceptadas.

**RESTRICCIONES**

Las restricciones que se impusieron son debidas a que el sistema es de tipo interactivo, y siendo en general, 80 el numero maximo de caracteres por renglon en una terminal de video, se restringen tablas y mapas usados por los programas.

Por esta causa, 6 es el numero de variables maximo para el Mapa de Karnaugh; para el programa de Quine Mc.Cluskey, solo se permiten hasta 4 variables de entrada; para Quine Mc.Cluskey Multiples Salidas, el numero de entradas y salidas no puede ser mayor de 3; en Reduccion de Estados secuenciales, hasta 16 estados son permitidos y hasta 5 salidas.

El sistema se encuentra disponible en dos cintas magneticas; en una de ellas, los dos programas escritos en FORTRAN 77 y en la otra, los 4 escritos en PASCAL.

## ANEXO A

La subrutina SOLUCION se encarga de resolver la funcion por medio del metodo de Quine Mc.Cluskey.

Los minterminos(cubos-0) de la funcion se separan de acuerdo al numero de '1' de su representacion binaria y se guardan en la matriz SOLUCI.ARREGLO.

Asi, los cubos-0 que tienen un '1' se almacenan en el renglon 1, los que tienen dos '1', en el renglon 2 y asi sucesivamente hasta analizar todos ellos. Ademas, se va incrementando el contador (SOLUCI.NELEM) de cada renglon en que se guarda algun cubo-0.

El paso siguiente es analizar con estos cubos-0 la posible combinacion de ellos, esto es, si cumplen con:

$$\text{CUBO-0(REGLON (N+1))} - \text{CUBO-0(REGLON N)} = \text{POTENCIA DE 2}$$

y si es asi, almacenarlos en los renglones cuya decena es 1 y unidades son (n+1). Asi se obtienen todos los cubos-1.

Simultaneamente se van marcando las posiciones en SOLUCI.ARRMARCA de los cubos que se van combinando con el caracter 'C' y los que no pueden hacerlo, con el caracter '\*'.

Cuando se ha terminado de analizar el renglon n, se tiene tambien el numero de minterminos que contiene.

Para obtener los cubos-2, se hace el mismo analisis verificando que todos los elementos que componen cada par de cubos que se revisan (uno a uno), tengan la misma diferencia y que sea potencia de 2. Y se en los renglones cuya decena es 2.

Este proceso se realiza sucesivamente hasta obtener el(los) ultimo(s) cubo(s) posible(s).

Como se observa, la busqueda posterior de los cubos que propone el usuario para minimizarse, es casi directo ya que con el



numero de minterminos se sabe en que renglones se debe buscar; de estos se eliminan aquellos que no tengan elementos y se verifica por cubos. Asi, si el primer elemento de un cubo de un renglon no es igual al del que se busca, se pasa directamente a verificar el primer elemento del siguiente cubo.

El siguiente paso en la solucion es sacar de la matriz SOLUCI.ARREGLO aquellos cubos que no se combinaron, esto es, obtenemos los implicantes primos.

En una matriz llamada SOLUCI.CUBIERTO se tendran como columnas todos los minterminos de la funcion y como renglones, los implicantes primos, se revisan los minterminos que contiene cada uno de ellos y se marca con el caracter '@' la columna correspondiente.

Despues de completar el analisis de los N implicantes, se revisan todas las columnas de la matriz y siempre que se encuentre una sola marca en una de ellas, se reconoce en que renglon se localiza, se marca el renglon de SOLUCI.MARC correspondiente con '\*' y se integran sus marcas al renglon (N+1).

Enseguida se revisa si en el renglon (N+1) hay alguna columna en blanco, si es asi, se sabe que existen implicantes primos secundarios. En caso contrario, la solucion optima es unica y esta dada por los implicantes marcados con '\*' en SOLUCI.MARC.

Si hay implicantes secundarios, se toman los cubos-0 e implicantes no cubiertos y se llena la matriz SOLUCI.NOCUBIERTO en la misma forma que se lleno SOLUCI.CUBIERTO.

Para obtener los implicantes primos secundarios se utiliza el metodo de Petrick, el cual consiste en:

Representando los implicantes a traves de letras, estas se interpretan como variables booleanas que adquieren un valor 1 si se selecciona dicho implicante primo, o 0 si no se escoge.

Partiendo de que todos los cubos-0 deben cubrirse, aquellos que aun no lo estan deberan cubrirse de alguna manera, esto nos da la idea de que el cubo-0(i) "y" el cubo-0(j) "y"... deben entrar en la solucion.

Ahora bien, cada uno de ellos puede ser cubierto por cualquiera

de los implicantes en que este contenido, esto nos dice para un cubo-0 en particular que, el implicante(k) "o" el implicante(m) "o" ... hacen ese cubo cierto.

Ejemplo:

Si se tiene la tabla de implicantes primos secundarios siguiente:

	6	7	15
a:	0	1	0
b:	1	1	0
c:	1	0	1
f:	0	1	1

traduciendo la explicacion anterior a una ecuacion se llega a:

$$(a+f)(a+c)(b+c)=1$$

efectuando las operaciones indicadas queda como:

$$ab+ac+cf=1$$

del cual se escoge aquel producto(cubo) que cueste menos.

Cada implicante que forma la(s) solucion(es) optima(s) es enviado a la subrutina SOLUMAR, la cual elimina las variables que varian en este implicante ya que esto quiere decir que lograron minimizarse durante el proceso. Asocia a las que se mantienen constantes su valor binario, obteniendose asi la solucion en funcion de las variables que originalmente asigna el programa.

Con esto se da por concluida esta subrutina. La presentacion de la solucion al usuario la hace el programa principal.

## ANEXO B

Los cubos-0 son almacenados en un arreglo de caracteres (MAT) de dos dimensiones, esto es, MAT(I,J). En la dimension I=1, se almacenan los cubos-0, en la dimension I=2, los cubos-1 y así sucesivamente. La dimension J se utiliza para almacenar los cubos que forman la totalidad de cubos de ese orden (identificado por el valor de I).

## EJEMPLO:

Si existen cinco cubos-0, estos se almacenaran en la dimension I=1 y los cinco cubos en los elementos identificados por J=1, 2, 3, 4, 5 esto es:

```

MAT(1,1) => contiene un cubo-0
MAT(1,2) => contiene un cubo-0
MAT(1,3) => contiene un cubo-0
MAT(1,4) => contiene un cubo-0
MAT(1,5) => contiene un cubo-0

```

Como los cubos pertenecientes a cada orden (cubos-0 por ejemplo), deberan agruparse segun el numero de "1" que contenga en la representacion binaria, para poder comparar grupos y formar cubos de un orden superior, esta agrupacion se realiza por medio de dos arreglos: IN22, ILON2; (de dos dimensiones cada uno de ellos) y contienen informacion que hace referencia o direcciones elementos del arreglo MAT. IN22 proporciona la posicion de inicio de cada grupo e ILON2, el numero de elementos o cubos que forman el grupo.

Para IN22, la dimension I=1 indica la posicion de inicio de cubos-0, I=2 para cubos-1, .. etc. La dimension J indica de que grupo es la posicion de inicio.

Para ILON2, la dimension I=1 indica el tamaño del grupo para cubos-0, I=2 para cubos-1, ... etc. y la dimension J indica de que grupo es el tamaño de I.

## EJEMPLO

```

IN22(1,1) = 1,      ILON2(1,1) = 1
IN22(1,2) = 2,      ILON2(1,2) = 4
IN22(1,3) = 6,      ILON2(1,3) = 1

```

Las variables anteriores indican que el primer grupo de cubos-0 inicia en la posicion 1 (IN22(1,1)=1) del arreglo MAT(1,J) y contiene un cubo-0 (ILON2(1,1)=1). El segundo grupo inicia en la posicion 2 (IN22(1,2)=4) y el tercer grupo inicia en la posicion 6 (IN22(1,3)=6) y contiene un cubo-0 (ILON2(1,3)=1).

El numero de grupos que existe para cada orden de cubos, se almacena en el vector IIM.

**EJEMPLO:**

Si se forman tres grupos de cubos-0:  
 $IIM(1)=3$   
 de la misma forma:  
 $IIM(2)=4$   
 indica que se formaron cuatro grupos cubos-1.

Como se explico anteriormente, los cubos se almacenan en el arreslo de caracteres MAT. Los cubos se almacenan en forma binaria y pueden situarse desde la posicion 1 hasta la posicion 6 (maximo numero de variables de la funcion) de cada elemento del arreslo MAT.

**EJEMPLO:**

un cubo-0 = 0011 (cuatro variables) se almacenara como:  
 $MAT(1,1) = 0011bb$   
 donde el caracter b indica un espacio en blanco.

Un cubo-3 = XX01X1 (seis variables) como:  
 $MAT(4,1) = XX01X1$

Ademas de almacenar cada elemento de MAT un cubo, tambien contiene informacion que indica si el cubo se combino para formar un cubo de orden superior, almacenando el caracter "C" en la posicion 10 de cada elemento que contiene un cubo combinado. Si el cubo no se combino, contendra un caracter blanco en la posicion 10.

Las posiciones 11, 12, 13 de cada elemento de MAT contienen la direccion que apunta a un elemento de un vector (IVA), el cual contiene el o los minterminos que forman el cubo.

**EJEMPLO:**

$MAT(1,1) = 0011 \quad C004$   
 $11\_ \text{posicion } 11-13$   
 $1\_\_\_ \text{posicion } 10$

El 004 indica que a partir de la posicion 4 del vector IVD, estan contenidos los minterminos que forman ese cubo.

Como en la posicion 11-13 se almacena la informacion de inicio de minterminos que forman un determinado cubo, pero no cuantos forman el cubo, se usa la siguiente formula para saberlo:

minterminos que forman un cubo =  $2^{*(I-1)}$

donde I es la dimension de MAT que se accesa para un determinado orden de cubos.

#### EJEMPLO:

Un cubo puede ser:

```

      I J
MAT(1,1) = 0011      C004
                |__Posicion 11-13
  
```

Los minterminos que forman este cubo inician en la posicion 004 del vector IVD y el numero de minterminos esta dado por:

$$2^{**}(I-1)=2^{**}(1-1)=2^{**}(0)=1$$

indica que solo un mintermino forma el cubo.

De la misma forma, para un cubo-3:

```

      I J
MAT(4,1) = X0XX      C018
                |__Posicion 11-13
  
```

la direccion de inicio es 018 y contendra:

$$2^{**}(4-1)=2^{**}3=8$$

8 minterminos almacenados en IVA, esto es:

IVD(18),IVD(19),IVD(20),IVD(21),IVD(22),IVD(23),IVD(23),IVD(25)

cada elemento contiene un mintermino que en conjunto forman un cubo-3.

Ahora bien, como se forma el vector IVD?

1) Al formar los cubos-0 en el arreglo MAT se asigna un numero ascendente a cada mintermino para hacer referencia a el por medio de ese numero, este numero se almacena en el vector IVD y simultaneamente se registra la posicion de almacenamiento en el arreglo MAT.

2) En el proceso de comparacion de los cubos de dos grupos para formar cubos de orden superior, se almacenan en el vector IVD los minterminos que se combinan de la siguiente forma:

Los minterminos que se combinan de cada cubo, se obtienen por informacion de su posicion en el arreglo MAT y se accesan en el vector IVD, en este momento se almacena el caracter "C" en la

posicion 10 de cada elemento de MAT o cubos comparados que se combinan para el arreglo MAT con dimension I+1, donde I identifica el orden de los cubos en proceso de comparacion. En la posicion 11-13 del elemento MAT de dimension I+1 se almacena la informacion de la posicion en el vector IVD, en la que quedaron almacenados los minterminos combinados que dieron lugar a la formacion del nuevo cubo.

A fin de agilizar la busqueda de una reduccion proporcionada por el usuario, esto es, minterminos que se combinan para formar un cubo de orden superior, se registra la informacion adicional respecto al vector IVD que contiene la secuencia de minterminos que forman todos los posibles cubos. En el vector LIN se guarda la posicion en la que inicia cada orden de cubos y en el vector LIM, cuantos cubos pertenecen a ese orden.

EJEMPLO:

$$\text{LIN}(1) = 1 ; \text{LIM}(1) = 10$$

Las variables anteriores indican que los cubos-0 inician en la posicion 1 y son 10 los cubos de ese orden.

De la misma manera:

$$\text{LIN}(2) = 10 ; \text{LIM}(2) = 5$$

indican que los cubos-1 inician en la posicion 10 y son 5 los cubos-1.

El proceso de comparacion de cubos para formar cubos de orden superior se realiza de la siguiente forma: 1) Si son cubos-0 los que se comparan, se cotejan digito a digito dos cubos pertenecientes a dos grupos adyacentes, si de la comparacion se llega a que solo difieren en un digito, los cubos se combinan y forman un cubo de orden superior; este cubo se construye a partir de uno de los cubos que se combinaron, sustituyendo el digito diferente por X.

b) Si los cubos que se compararon son de orden superior a 0:

Se localizan cubos de dos grupos adyacentes que contengan el caracter "X" en la misma posicion y se comparan los digitos restantes, combinandose si solo difieren en un digito.

Al terminar el proceso de comparacion de cubos para formar nuevos cubos de orden superior, se procede a identificar los implicantes primos, haciendo un recorrido a traves de todos los cubos almacenados en MAT y tomando aquellos cubos que no contienen el caracter "c" en la columna 10.

El vector TAB contendra todos los implicantes primos.

Descripcion del vector TAB:

Los elementos del vector TAB, son de tipo caracter, con un tamaño de 14 caracteres para almacenamiento por elemento.

La posicion 1 de cada elemento de TAB se usa para indicar si un implicante primo es esencial, para tal efecto, almacena el caracter "\*".

La posicion 2 indica el orden a que pertenece el implicante primo.

EJEMPLO:

Si el implicante primo pertenece a los cubos-0, la posicion 2 contendra un 1. Si pertenece a los cubos-1, contendra un 2 y asi sucesivamente.

El numero almacenado en la posicion 2 indicara entonces el nivel de reduccion y es un indicador del costo de su implementacion como un elemento de un circuito que realiza la funcion.

En la posicion 11-13 se encuentra la direccion del vector IVD, en la que se localizan los minterminos que forman al implicante primo.

Los implicantes primos se identifican de la siguiente manera:

Cada mintermino de la funcion se busca en los implicantes primos, si solo lo contiene un implicante, este se marca como esencial, marcando la columna 1 de TAB en que se encuentra el implicante con "\*".

El siguiente paso es marcar los minterminos que cubren los implicantes esenciales, para lo cual se usa el vector que contiene todos los minterminos que forman a los implicantes primos esenciales revisandolo para saber si se cubren todos los minterminos con los implicantes primos esenciales.

Si los implicantes esenciales cubren todos los minterminos:

La solucion optima es solo una y la dan todos estos implicantes.

Si no cubren a todos los minterminos:

- Se crea un arreglo que contiene la tabla de la que se pueden obtener los implicantes primos secundarios por dominio.

- Se aplica el metodo de Petrick para obtenerlos como a continuacion se describe:

a) se identifica que implicantes primos cubren cada mintermino no cubierto por los esenciales, de esto resulta una suma logica.

b) Al hacer el proceso anterior para cada uno de los minterminos no cubiertos, se obtendra una suma logica por cada uno de ellos, obteniendo finalmente un producto de estas sumas.

c) Aplicando las leyes distributivas y eliminando por multiplicacion, se obtiene una suma de productos, de ellos se seleccionan aquellos que contengan el menor numero de implicantes primos y se calcula su costo, sumando el costo de cada implicante que forman este producto, se seleccionan aquellos que tengan el menor numero de implicantes y el menor costo.

Si existen varios productos que cumplan esta condicion, existira una solucion optima por cada uno de ellos.



## ANEXO C

Diccionario de parámetros usados por las subrutinas de los programas elaborados en FORTRAN 77

Los parámetros se describen de la siguiente manera:

NOMBRE DEL PARÁMETRO: CLASE, FORMATO, TIPO

B: Caracteres, A6, Escalar \*Contiene los caracteres A, B, C, D, E, F, usados para representar a las variables en el mapa de Karnaugh. El cambio en la nomenclatura de las variables modifica el contenido de esta variable\*

BUFF: Caracteres, A1, Vector(640) \*Usado para mostrar simultáneamente el mapa de Karnaugh y un mapa con todos los minterminos posibles para el número de variables dado (al mostrarse por terminal, produce ocho renglones de ochenta columnas)\*

DIR: Entero, I2, Vector(3) \*Dirección a los productos de implicantes obtenidos por el método de Petrick que satisfacen menor número de elementos, menor costo\*

IB: Entero, I1, Escalar \*IB = 1 para respuesta afirmativa, IB = 0 para respuesta negativa\*

IBAN: Entero, I1, Escalar \*Parámetro usado para control en la presentación del mapa de Karnaugh\*

ID: Entero, I2, Vector(64) \*Representación decimal de los minterminos\*

IDD: Entero, I2, Vector(64) \*Representación decimal de los minterminos agrupados para producir una reducción\*

IDIR: Entero, I2, Vector(64) \*Contiene todos los posibles minterminos en su equivalente decimal para un número de variables de la función\*

IFI: Entero, I2, Escalar \*Número de minterminos proporcionados por el usuario\*

IFII: Entero, I2, Escalar \*Número de minterminos agrupados para producir una reducción\*

IFORM: Entero, I2, Escalar \*Indica la forma en la que se interpretarán los minterminos proporcionados por el usuario\*

III: Entero, I2, Escalar \*Numero de implicantes primos\*

IMAT: Entero, I2, Arreglo(8x8) \*Minterminos para numero de variables de la funcion, arreglados en forma matricial\*

INDOP: Entero, I1, Escalar \*INDOP = 0 inicializa el mapa ubicando el caracter '0' en las posiciones que pueden ocupar los minterminos. INDOP = 1 ubica el caracter '1' en la posicion correspondiente a los minterminos proporcionados por el usuario\*

IVD: Entero, I2, Vector(1000) \*Contiene los minterminos que forman cada uno de los cubos de todos los ordenes\*

IX: Caracteres, A1, Vector(960) \*Variable usada para la graficacion del mapa de Karnaugh por medio de la terminal\*

KBA: Entero, I1, Escalar \*Selecciona presentacion del mapa de Karnaugh: KBAN = 0, muestra dos mapas, uno para ubicar los minterminos del problema y otro que presenta los posibles minterminos para el numero de variables de la funcion. KBAN = 1, solo presenta el segundo mapa descrito\*

LIN: Entero, I3, Vector(10) \*Direcciona posiciones del vector IVD, indicando inicio de cubos de cada orden\*

LIM: Entero, I3, Vector(10) \*Numero de cubos de cada orden\*

M: Caracteres, A6, Vector(64) \*Representacion de los minterminos en forma binaria\*

MAT: Caracteres, A14, Arreglo(10x64) \* -Inicialmente se almacenan los cubos-0 en MAT. - Estructura de trabajo para obtener cubos de orden superior a cubos-0. - Cada elemento de MAT, ademas de contener un cubo, contiene informacion que direcciona al vector IVD, en donde se encuentran los minterminos que forman el cubo y un caracter indicador de 'cubo combinado'\*

MATE: Caracter, A30, Vector(70) \*Contiene suma de productos de implicantes primos obtenidos por el metodo de Petrick\*

MCOL: Entero, I2, Escalar \*Numero de columnas del mapa de Karnaugh\*

N: Entero, I2, Escalar \*Numero de variables de la funcion de conmutacion\*

NIP: Entero, I2, Escalar \*Numero de agrupaciones de minterminos hechas por el usuario\*

NIS: Entero, I2, Escalar \*Numero de productos de implicantes primos que se obtuvieron por el metodo de Petrick, que cumplen la misma condicion del menor numero de elementos y menor costo\*

NOP: Entero, I2, Escalar \*Numero de implicantes primos esenciales + numero de implicantes primos secundarios\*

NREN: Entero, I2, Escalar \*Numero de rensiones del mapa de Karnaush\*

TAB: Caracteres, A30, Vector(40) \*Contiene a los implicantes primos, orden del cubo, direccionador al vector IVD que referencia minterminos componentes\*

VD: Alfanumerica, A1, Vector(64) \*Contiene el mapa de Karnaush inicializado con el caracter '0' y posteriormente ubica el caracter '1' en las posiciones de los minterminos de la funcion\*

## BIBLIOGRAFIA

- 1) Teoria de Conmutacion y Diseo y Diseo Logico  
Frederick J. Hill, Gerald R. Peterson  
Limusa 1980
- 2) Logica Digital y Diseo de Computadores  
M. Morris Mano  
Prentice/Hall International 1982
- 3) Digital Integrated Electronics  
Herbert Taub / Donal Schilling  
Mc. Graw Hill Cosacusha 1977,
- 4) Logic Design of Digital Systems  
Donald L. Dietweyer  
Allyn and Bancou 1978
- 5) Diseo con Circuitos Integrados TTL  
Texas Instrument Sexs 1978
- 6) Programacion en Pascal  
Groseno  
Fondo Educativo Interamericano 1984
- 7) Manual Report  
Niklaus Wirth
- 8) Structured Pascal  
Paul Tremblay - Richard Bund.  
Mc. Graw Hill 1980
- 9) FORTRAN 77  
Donald M. Monro  
Arnold 1982
- 10) Programing Standard Pascal  
R. C. Holt, J. N. P. Humen  
Reston 1880
- 11) Pascal for Burroughs B6700 - B7700. Supplement to Pascal  
User Manual and Report  
A. H. J. Sale 1977
- 12) Systematic Programing  
Niklaus Wirth