



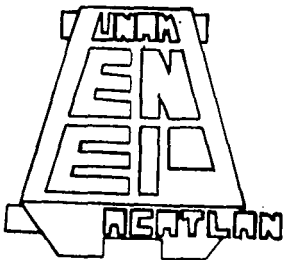
UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS
PROFESIONALES ACATLAN

UN SISTEMA PARA EL ANALISIS
DE MEDIOS CONTINUOS EN DOS
DIMENSIONES POR EL METODO
DEL ELEMENTO FINITO.

T E S I S
Que para obtener el título de:
LICENCIADO EN MATEMATICAS
APLICADAS Y COMPUTACION
P R E S E N T A :
Sergio González Castañeda

Asesor: Ma. de la Gracia Barquero Díaz
Barriga.



MEXICO, D.F.
1995

FALLA DE ORIGEN



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 está 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.

DEDICATORIA

**Dedico esta tesis como una muestra
de agradecimiento hacia las
siguientes personas:**

A Dios

Por estar siempre conmigo y por
ser todo lo grande que es en mi
vida.

**A mi padre
Rosendo González Salinas**

Quien siempre ha demostrado tener
interés por todo lo que hago, y
que como padre y amigo siempre
será un ejemplo para mí.

**A mi madre
Guadalupe Castañeda Valenzuela**

Quien es para mí la mejor de las madres
ya que ha dedicado parte de su vida
dándome cariño, confianza y un
incondicional respaldo para realizar
todas mis metas.

**A mi tía
Ma. de la Luz Castañeda Reyes**

De quien siempre obtuve apoyo en todo
momento y a la que quiero demasiado.

A mis hermanas
Adriana González Castañeda
Rosario González Castañeda
Beatriz González Castañeda

Porque me apoyaron y animaron en la realización de este trabajo y con quiénes espero poder estar en todo momento para poder ayudarlas en todo lo que pueda... por siempre.

A mi pequeño sobrino
Raul Leonel González Jaramillo

Quien siempre podrá contar conmigo como un amigo al tener que tomar decisiones en su vida y en todo lo que desee.

A mi cuñado
Baldomero Victoria Honroy

De quien he aprendido mucho y al que con el ejemplo me ha enseñado a lograr lo que deseo sin importar las dificultades a vencer.

A mis tío y primas

Quiénes en todo momento me han dado mucho cariño y a los que veo como una segunda familia... gracias.

Al M. ING. Fernando Vera Badillo

Por haber sugerido este trabajo siendo él quien tuvo que soportar las primeras equivocaciones en la realización del mismo y con el que quedo profundamente agradecido, puesto a su disposición en todo momento.

Al ING. Ma. de la Gracia Barquero Diaz Barriga

Por haber revisado este trabajo y por ser una persona muy agradable para conmigo... gracias.

A Ivonne Rodríguez Perez

Por haber confiado en mi en todo momento y por demostrarme ser una persona muy especial conmigo.

A ellos les dedico:

"La amistad es como una gota de mercurio; hay que mantener la mano abierta para conservarla; si la cerramos, se nos escapa".

-Luis Fernando F.

A mis amigos
Jesús Ruben Arzola Zúñiga
Ma. de Lourdes Covarrubias Zamorano

A los que nunca voy a olvidar ya que siempre podrán contar conmigo en todo momento puesto que hay muchas cosas común entre nosotros.

Sergio Ramírez Hernández
Uriel Norberto Castillo Lázaro

Por que la amistad que me brindaron fue muy especial y aunque cada quien a elegido un camino diferente en su vida nunca me olvidare de ellos.

A todos mis amigos les dedico:

"Algunos de los momentos más satisfactorios y bellos de una amistad ocurre en los impensados espacios abiertos que median entre una y otra actividad común proyectada. Importa mucho que los amigos permitan la existencia de estos espacios".

-Cristine Leefeld.

I N D I C E

	Pág.
INTRODUCCION.....	1
1. FORMULACION INTEGRAL PARA LA SOLUCION DE ECUACIONES LINEALES.....	3
1.1 Método de diferencias finitas.....	3
1.2 Métodos variacionales.....	4
1.3 Método de los residuos pesados.....	5
1.3.1 Método de colocación.....	6
1.3.2 Método de subdominios.....	7
1.3.3 Método de Galerkin.....	7
1.3.4 Método de los mínimos cuadrados.....	7
1.4 Ejemplo.....	8
2. ECUACIONES DE EQUILIBRIO PARA EL ESTADO PLANO DE ESFUERZOS Y SU SOLUCION POR EL METODO DEL ELEMENTO FINITO.....	18
2.1 Ecuación de equilibrio del medio continuo.....	19
2.2 Ecuaciones de forma.....	29
2.3 Secuencia del método del elemento finito.....	34
3. SOLUCION DE ECUACIONES LINEALES.....	47
3.1 Método de Gauss-Crout.....	47
3.2 Arreglos para almacenamiento en memoria.....	52
3.3 Solución de ecuaciones lineales empleando arreglos unidimensionales.....	55
4. CONSIDERACIONES PREVIAS EN CUANTO AL ENTORNO DE LA PROGRAMACION DEL SISTEMA.....	67
4.1 Equipo de cómputo a utilizar.....	68
4.2 Análisis de la versatilidad en los lenguajes propuestos para la programación del sistema.....	72
4.2.1 Ventajas y desventajas en FORTRAN.....	73
4.2.2 Ventajas y desventajas en el lenguaje C.....	75
4.2.3 Manipulación de archivos.....	78
4.3 Conceptos generales sobre representaciones en pantalla del lenguaje C.....	79
4.3.1 Representación en modo texto.....	80
4.3.2 Representación en modo gráfico.....	84
5. PROGRAMACION DEL SISTEMA POR COMPUTADORA.....	89
5.1 Análisis global del sistema.....	90
5.1.1 Partes que componen al sistema.....	93
5.1.2 Organización del sistema.....	95
5.1.3 Secuencia del proceso de operación del sistema.....	96
5.2 Preprocesador.....	99
5.2.1 Generalidades.....	99
5.2.2 Organización.....	101
5.2.3 Manual del usuario.....	107
5.3 Procesador.....	132
5.3.1 Generalidades.....	132
5.3.2 Organización.....	133

5.3.3	Manual del usuario.....	136
6.	EJEMPLOS DE APLICACION.....	142
6.1	Problema "A".....	142
6.1.1	Datos para el preprocesador.....	143
6.1.2	Datos para el procesador.....	149
6.1.3	Análisis de los resultados obtenidos del procesador.....	150
6.2	Problema "B".....	154
6.2.1	Datos para el preprocesador.....	155
6.2.2	Datos para el procesador.....	160
6.2.3	Análisis de los resultados obtenidos del procesador.....	161
	CONCLUSIONES.....	165
	BIBLIOGRAFIA.....	168
	APENDICE A. LISTADO DE PROGRAMAS DE COMPUTADORA -PREPROCESADOR -PROCESADOR.....	169
	APENDICE B. LISTADO DE ARCHIVOS GENERADOS POR EL -PREPROCESADOR -PROCESADOR.....	273

INTRODUCCION

Cuando se aborda el tema de diseño de estructuras se han de considerar una diversidad de aspectos para llevar a cabo dichos diseños, el diseño ha de ser idealizado cuidadosamente antes de ser llevado a la práctica ya que un mal modelo podría producir resultados erróneos.

Un aspecto de gran relevancia ha sido las condiciones de carga a que son sometidas las estructuras durante su largo período de utilización ya que son las que determinan bajo ciertas circunstancias el comportamiento de la estructura.

Esta labor de investigación ha motivado a la elaboración de un programa de computadora, para contribuir con la solución de estados planos de esfuerzos en forma eficiente, conjuntamente se ha pensado en la posibilidad de contar con apoyo de los métodos numéricos para lograr formar las bases de la teoría de la programación cuyas aportaciones resulten ser los cimientos bajo los cuales se puedan apoyar las ideas de planeación, diseño y programación del sistema a desarrollar.

Dada, la aportación de gente con conocimientos de ingeniería y de los lineamientos científicos con que cuenta dicha área, se ha logrado realizar un intercambio de ideas con el fin de elaborar un programa dentro del área computacional cuyo objetivo logre facilitar el análisis de estructuras mediante el método del elemento finito, por considerarse uno de los métodos numéricos más aproximados

para resolver ecuaciones de equilibrio esfuerzo-deformación para el análisis de esfuerzos de medios continuos.

Daremos paso al desarrollo del presente trabajo de investigación, esperando finalmente el cumplimiento de los objetivos para los que fue creado, siendo así que en un momento determinado se establezca la iniciativa para la realización de diferentes investigaciones sobre éste u otros temas de interés particular.

1.- FORMULACION INTEGRAL PARA LA SOLUCION DE ECUACIONES DIFERENCIALES.

Un camino para la solución de problemas físicos gobernados por ecuaciones diferenciales es obtener la solución analítica, pero en muchas situaciones ésta es difícil de obtener, ya que la región que se considera es irregular y esto significa que matemáticamente es imposible describir su frontera.

Los métodos numéricos pueden ser usados para obtener una solución aproximada cuando la solución analítica no puede ser desarrollada. Todas las soluciones numéricas dan valores en puntos discretos de un conjunto de parámetros independientes.

El procedimiento de solución completa es repetir el cálculo con el cambio en cada paso de esos parámetros.

Existen varios procedimientos para obtener la solución numérica de una ecuación diferencial.

Estos métodos se pueden separar en tres grupos básicos:

- 1.- Método de diferencias finitas.
- 2.- Métodos variacionales.
- 3.- Métodos de los residuos pesados.

1.1 Método de diferencias finitas.

El método de diferencias finitas aproxima las derivadas de la ecuación diferencial usando ecuaciones en diferencias. Este método es usado generalmente cuando la frontera de la región a estudiar es paralela a los ejes coordenados. Estos

métodos no son recomendables cuando la frontera es curva o irregular.

1.2 Métodos variacionales.

La aproximación variacional involucra la integral de una función que produce un nuevo número. La función que produce un número muy bajo tiene la propiedad adicional de satisfacer una ecuación diferencial específica. Para clarificar este concepto sea la integral:

$$\pi = \int_0^H \left[\frac{D}{2} \left[\frac{dy}{dx} \right]^2 - Qy \right] dx$$

(1.1)

El valor numérico de π puede ser calculado dando una ecuación específica:

$$y=f(x)$$

El cálculo variacional demuestra que una ecuación particular:

$$y=g(x)$$

El cual tiene en su límite un valor numérico bajo de π , es la solución de la ecuación diferencial:

$$D \frac{d^2y}{dx^2} + Q = 0$$

(1.2)

Con las condiciones frontera:

$$\begin{aligned} y(0) &= y \\ y(H) &= y \end{aligned}$$

El proceso puede ser reversible. Dando una ecuación diferencial se puede obtener una solución aproximada al sustituir diferentes funciones prueba dentro de la funcional, si éstas dan un valor mínimo de π .

Los métodos variacionales son la base para la formulación del método del elemento finito, pero su desventaja es que no es aplicable a cualquier ecuación diferencial que contenga términos en su primera derivada.

1.3 Métodos de los residuos pesados.

Los métodos de los residuos pesados involucran una integral.

En estos métodos una solución aproximada se sustituye en la ecuación diferencial. Dado que esta solución aproximada no satisface la ecuación, éste resulta ser un término residual. Supóngase que:

$$y = h(x)$$

Es una solución aproximada de la ecuación diferencial al sustituir se tiene:

$$D \frac{d^2 h(x)}{dx^2} + Q = R(x) \neq 0$$

(1.3)

Donde $y=h(x)$ no satisface la ecuación. El método de los residuos pesados requiere:

$$\int_0^H W_i(x) R(x) dx = 0$$

(1.4)

Se requiere que el residual $R(x)$ sea multiplicado por una función de peso $W_i(x)$ igual al número de incógnitas en la solución aproximada.

Dependiendo del tipo de función de peso $W_i(x)$ a escoger existen varios métodos, los más conocidos son:

1.3.1 Método de colocación.

La función de peso es $W_i(x) = \delta(x - X_{-i})$. El escoger esta función es lo equivalente a considerar que el residual vale cero en puntos específicos. El número de puntos a escoger es igual al número de incógnitas en la solución aproximada.

1.3.2 Método de subdominios.

La función de peso es unitaria $W_i(x)=1$ sobre una determinada región, esto equivale a hacer que, en la integral el residual sea para un intervalo de la región. El número de intervalos será igual al número de incógnitas en la solución aproximada.

1.3.3 Método de Galerkin.

Utiliza las mismas funciones que la ecuación aproximada como función de peso $W_i(x)$.

1.3.4 Método de los mínimos cuadrados.

Este método utiliza el residual como una función de peso y obtiene un nuevo error definido como:

$$E_r = \int_0^H [R(x)]^2 dx$$

(1.5)

Este error debe ser mínimo con respecto a las incógnitas en la solución aproximada.

El método de los mínimos cuadrados se puede utilizar para la formulación de la solución por el método del elemento finito.

El método variacional y el de los residuos pesados consideran una integral, por lo que se conoce como formulaciones integrales. El tipo de solución numérica basada en formulaciones integrales se utiliza de apoyo para formular la solución por el método de Galerkin.

1.4 Ejemplo.

Para ilustrar el uso de los métodos integrales para la obtención de una solución aproximada de una ecuación diferencial que modela un problema físico, se considera el siguiente ejemplo.

Sea una viga simplemente apoyada sujeta a momentos concentrados en cada extremo como se muestra en la fig. 1.1.

La ecuación diferencial es:

$$EI \frac{d^2y}{dx^2} - M(x) = 0$$

(1.6)

Sus condiciones frontera:

$$y(0) = 0 \quad ; \quad y(H) = 0$$

(1.7)

en donde:

E = módulo de elasticidad.
 I = momento de inercia.
 M(x) = es el momento flexionante.

La solución exacta de la ecuación diferencial es:

$$y(x) = \frac{M_0 x}{2EI} (x - H)$$

(1.8)

Método variacional.

La formulación integral de la ecuación diferencial es:

$$\Pi = \int_0^H \left[\frac{EI}{2} \left[\frac{dy}{dx} \right]^2 + M_0 y \right] dx$$

(1.9)

Se propone una solución aproximada de la deflexión la cual es:

$$y(x) = A \operatorname{sen} \frac{\pi x}{H}$$

(1.10)

donde A es un coeficiente indeterminado, esta solución puede ser aceptable porque satisface las condiciones frontera:

$$y(0) = 0 \quad \text{y} \quad y(H) = 0.$$

El valor de A debe ser una aproximación a la deflexión de la curva cuando π es mínimo, al evaluar A debe ser escrito en función de A y minimizar con respecto a A.

Al derivar la ecuación 1.10 con respecto a x se tiene:

$$\frac{dy}{dx} = \frac{A\pi}{H} \cos \frac{\pi x}{H}$$

(1.11)

Al sustituir la ecuación 1.11 en 1.9:

$$\pi = \int_0^H \left[\frac{EI}{2} \left[\frac{A\pi}{H} \cos \frac{\pi x}{H} \right]^2 + M_0 A \sin \frac{\pi x}{H} \right] dx$$

(1.12)

Al resolver la integral:

$$\pi = \left[\frac{EI\pi^2}{4H} \right] A^2 + \left[\frac{2M_0H}{\pi} \right] A$$

(1.13)

Al minimizar la función (ecuación 1.13) con respecto a

A:

$$\frac{\partial \pi}{\partial A} = 2 \left[\frac{EI\pi^2}{4H} \right] A + \frac{2M_0H}{\pi} = 0$$

(1.14)

Al despejar:

$$A = - \frac{4M_0H^2}{\pi^3EI}$$

(1.15)

La solución aproximada es:

$$y(x) = - \frac{4M_0H^2}{\pi^3EI} \operatorname{sen} \frac{\pi x}{H}$$

(1.16)

Método de colocación.

Se requiere que la ecuación residual para obtener la solución aproximada sea cero en algunos puntos para obtener el valor de A.

Al sustituir una solución aproximada (ecuación 1.10) en la ecuación diferencial (ecuación 1.6) se define el residuo:

$$R(x) = -EI \frac{A\pi^2}{H^2} \operatorname{sen} \frac{\pi x}{H} - M_0$$

(1.17)

donde:

$$\frac{d^2y}{dx^2} = -A \frac{\pi^2}{H^2} \operatorname{sen} \frac{\pi x}{H}$$

(1.18)

existe un coeficiente indeterminado A, por lo tanto debe de haber un punto donde el residual R(x) vale cero, al seleccionar $x=H/2$ se tiene:

$$R \left[\frac{H}{2} \right] = -EI \frac{A\pi^2}{H^2} \operatorname{sen} \frac{\pi}{2} - M_0 = 0$$

al despejar:

$$A = - \frac{M_0 H^2}{EI \pi^2}$$

Se puede escoger otro punto diferente y obtener otra solución aproximada.

Método de los subdominios.

El método requiere que la siguiente integral se cumpla en tantos intervalos como los coeficientes indeterminados existan:

$$\int R(x) dx = 0$$

Se debe escoger el tamaño del intervalo que se va a considerar, en nuestro ejemplo hay únicamente un coeficiente desconocido por lo tanto el intervalo puede ser de $[0, H]$ se tiene:

$$\int_0^H R(x) dx = \int_0^H \left[-EI \frac{A\pi^2}{H^2} \sin \frac{\pi x}{H} - M_0 \right] dx = 0$$

al efectuar la integración:

$$- \left[\frac{2EI\pi}{H} \right] A - M_0 H = 0$$

$$A = - \frac{M_0 H^2}{2\pi EI}$$

la solución aproximada es:

$$y(x) = - \frac{M_0 H^2}{2\pi EI} \operatorname{sen} \frac{\pi x}{H}$$

Método de Galerkin.

Cuando se usa el método de Galerkin, la integral $W_i(x)R(x)dx$ los valores de $W_i(x)$ son las mismas funciones que las que utilizan en la solución aproximada, en este ejemplo hay únicamente una función de peso $W_i(x) = \operatorname{sen} \frac{\pi x}{H}$ por lo que la ecuación residual en forma integral queda:

$$\int_0^H \operatorname{sen} \frac{\pi x}{H} \left[-EI \frac{A\pi^2}{H^2} \operatorname{sen} \frac{\pi x}{H} - M_0 \right] dx = 0$$

al integrar:

$$- \frac{EI\pi^2 A}{2H} - \frac{2M_0 H}{\pi} = 0$$

al despejar A de la ecuación anterior:

$$A = - \frac{4M_0 H^2}{\pi^3 EI}$$

y la solución aproximada es:

$$y(x) = -\frac{4M_0H^2}{\pi^3EI} \operatorname{sen} \frac{\pi x}{H}$$

Esta solución es igual a la obtenida usando un método variacional.

Método de los mínimos cuadrados.

Un nuevo error:

$$E_r = \int [R(x)]^2 dx$$

es formado cuando el método de los mínimos cuadrados al sustituir la ecuación residual:

$$E_r = \int_0^H \left[-\frac{EI\pi^2}{H^2} A \operatorname{sen} \frac{\pi x}{H} - M_0 \right]^2 dx$$

al integrar:

$$E_r = \frac{A^2 H}{2} \left[\frac{EI \pi^2}{H^2} \right]^2 + \frac{4M_0 EI \pi}{H} A + M_0^2 H$$

el error es mínimo con respecto a A:

$$\frac{\partial E_r}{\partial A} = AH \left[\frac{EI \pi^2}{H^2} \right]^2 + \frac{4M_0 EI \pi}{H} = 0$$

al despejar A y sustituir en la solución aproximada:

$$y(x) = - \frac{4M_0 H^2}{\pi^3 EI} \operatorname{sen} \frac{\pi x}{H}$$

es la misma solución obtenida por los métodos variacionales y de Galerkin.

Con base en los métodos anteriores es posible hacer la formulación para la obtención de las ecuaciones de solución por el método del elemento finito. Por el MEF se puede plantear la solución para problemas de equilibrio, de eigenvalores y de propagación; al considerar el tipo de solución a partir de los métodos de los residuos pesados o de los métodos variacionales, en el capítulo siguiente se hace el planteamiento para ecuaciones de equilibrio de la elasticidad lineal para el análisis bidimensional de esfuerzos de un continuo.



Figura 1.1 Esquema de la viga mencionada en el ejemplo.

2.-ECUACIONES DE EQUILIBRIO PARA EL ESTADO PLANO DE ESFUERZOS Y SU SOLUCION POR EL METODO DEL ELEMENTO FINITO.

El método del elemento finito en términos generales es un método para resolver ecuaciones diferenciales de problemas en la frontera o de valores en la frontera e iniciales.

En la mecánica estructural, el método del elemento finito es una extensión de los métodos matriciales (estructuras esqueléticas) para analizar medios continuos, el cual se discretiza como una estructura formada por un número de elementos interconectados entre sí en los puntos nodales como se indica en la fig. 2.1, este método resulta muy útil cuando la geometría del continuo es bastante compleja.

Para plantear las ecuaciones de equilibrio de la teoría de la elasticidad (estado plano de esfuerzos) y proponer su solución por el método del elemento finito, se pueden seguir dos caminos, uno es mediante el método de los residuos pesados en donde se demuestra que el principio del trabajo virtual es una forma débil de las ecuaciones de equilibrio y mediante una formulación de Galerkin se obtienen las ecuaciones del método del elemento finito que se utilizan para la solución, otro criterio es por los métodos variacionales que a continuación se expone, en ambos caminos se llega a las mismas ecuaciones.

2.1 Ecuación de equilibrio del medio continuo.

En la teoría de la elasticidad lineal la ecuación variacional viene dada por el principio del trabajo virtual, el cual se obtiene de las ecuaciones de Cauchy del movimiento, en forma esquemática el enunciado del principio es:

$$\begin{array}{ccccccc} \text{FUERZAS} & & \text{FUERZAS DE} & & \text{FUERZAS DE} & & \text{FUERZAS DE} \\ \text{INTERNAS} & + & \text{INERCIA} & = & \text{SUPERFICIE} & + & \text{CUERPO} \end{array}$$

esto en notación matricial es:

$$\int_{\Omega} \delta \underline{\epsilon}^T \underline{\sigma} d\Omega + \int_{\Omega} \rho \delta \underline{u}^T \ddot{\underline{u}} d\Omega = \int_{\Gamma} \delta \underline{u}^T \underline{\sigma}_{(n)} d\Gamma + \int_{\Omega} \rho \delta \underline{u}^T \underline{f} d\Omega$$

(2.1)

en donde:

Ω = punto interior de la estructura.

Γ = condiciones de frontera.

δ = primera variación que opera sobre las cantidades que proceden.

$\underline{\sigma}_{(n)}$ = componentes de las cargas que por unidad de superficie actúan.

sobre la frontera del cuerpo:

\underline{f} = vector de cargas de cuerpo.

$\underline{\epsilon}$ = componentes del tensor de deformación.

$\underline{\sigma}$ = componentes del tensor de esfuerzos.

\underline{U} = vector de desplazamiento.

ρ = densidad de masa por unidad de volumen.

Al considerar una solución aproximada del campo de desplazamientos:

$$\underline{U} \approx \tilde{\underline{U}}$$

(2.2)

la solución aproximada se propone mediante funciones de forma y se tiene:

$$\underline{U} \approx \tilde{\underline{U}} = \underline{N} \tilde{\underline{U}}$$

(2.3)

en donde $\tilde{\underline{U}}$ resultan los desplazamientos que se van a obtener en la solución y \underline{N} son las funciones de forma. De acuerdo a la teoría de la elasticidad se pueden definir las siguientes ecuaciones:

$$\underline{\epsilon} = \underline{L} \underline{U}$$

(2.4)

$$\underline{\sigma} = \underline{D} \underline{\epsilon}$$

(2.5)

en donde $\underline{\epsilon}$ y $\underline{\sigma}$ están definidas en la ecuación 2.1, \underline{L} es una matriz de operadores (ecuación 2.6) y \underline{D} es la matriz de coeficientes elásticos para el caso de estado plano de esfuerzos (ecuación 2.7):

$$\underline{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}$$

(2.6)

$$\underline{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-\nu) \end{bmatrix}$$

(2.7)

al considerar la solución aproximada se sustituye la ecuación 2.3 en las ecuaciones 2.4 y 2.5 :

$$\underline{\epsilon} = \underline{L} \underline{N} \underline{U}$$

(2.8)

$$\underline{\sigma} = \underline{D} \underline{I} \underline{N} \underline{U}$$

(2.9)

se define:

$$\underline{B} = \underline{L} \underline{N}$$

(2.10)

al sustituir la ecuación 2.10 en las ecuaciones 2.8 y 2.9 :

$$\underline{\sigma} = \underline{D} \underline{E} \underline{U}$$

(2.11)

$$\underline{\epsilon} = \underline{B} \underline{U}$$

(2.12)

al considerar la primera variación con los desplazamientos aproximados se tiene:

$$\delta \underline{\tilde{U}} = \delta \underline{\tilde{U}} = \underline{N} \delta \underline{\tilde{U}}$$

(2.13)

$$\delta \underline{\tilde{U}}^T = d \underline{\tilde{U}}^T = \delta \underline{\tilde{U}} \underline{N}^T$$

(2.14)

al sustituir las ecuaciones 2.11 a 2.14 en 2.1 se tiene:

$$\int_{\Omega} \delta \underline{\tilde{U}}^T \underline{B} \underline{D} \underline{B} \underline{U} \, d\Omega + \int_{\Omega} \delta \underline{\tilde{U}}^T \underline{N}^T \underline{N} \underline{\tilde{U}} \, d\Omega =$$

$$\int_{\Gamma} \delta \underline{\tilde{U}}^T \underline{N}^T \underline{\sigma}_{,n} \, d\Gamma + \int_{\Omega} \delta \underline{\tilde{U}}^T \underline{N}^T \underline{f} \, d\Omega$$

(2.15)

al factorizar y simplificar:

$$\int_{\Omega} \underline{B}^T \underline{D} \underline{B} \tilde{\underline{U}} d\Omega + \int_{\Omega} \underline{e} \underline{N}^T \underline{N} \tilde{\underline{U}} d\Omega =$$

$$\int_{\Gamma} \underline{N}^T \underline{\sigma}_{(n)} d\Gamma + \int_{\Omega} \underline{N}^T \underline{f} d\Omega$$

(2.16)

la ecuación 2.16 es válida para toda la región del continuo ($\Omega + \Gamma$) pero al utilizar el método del elemento finito se tienen las siguientes hipótesis:

- a) El medio continuo se divide en un número finito de regiones y a cada una de estas regiones se le llama elemento finito.
- b) Los elementos finitos se suponen están intercomunicados en un número finito de puntos nodales situados en las fronteras de los mismos y los desplazamientos de los puntos nodales son las incógnitas básicas del problema.
- c) Se define en forma única el campo de desplazamientos en cualquier punto del elemento finito en función de los desplazamientos de los puntos nodales.
- d) Conocidos los desplazamientos se pueden definir en forma única, las deformaciones y los esfuerzos.

Según las hipótesis anteriores la región $\Omega + \Gamma$ se divide en elementos finitos, esto es:

$$\Omega = \sum_{i=1}^{nelem} \Omega_i^e \quad \Gamma = \sum_{i=1}^{nelem} \Gamma_i^e$$

nelem = N° de elementos.

(2.17)

y la ecuación 2.16 se aplica en forma aislada a todos los elementos finitos por lo que se tiene:

$$\int_{\Omega^e} \mathbf{B}^T \mathbf{D} \mathbf{B} \mathbf{u}^e d\Omega^e + \int_{\Omega^e} \rho \mathbf{N}^{eT} \mathbf{N}^e \mathbf{u}^e d\Omega^e =$$

$$\int_{\Gamma^e} \mathbf{N}^{eT} \underline{\sigma}_{(n)} d\Gamma^e + \int_{\Omega^e} \mathbf{N}^{eT} \underline{f} d\Omega^e$$

Ω^e = región de cada elemento.
 Γ^e = frontera de cada elemento.

(2.18)

en donde:

$$\underline{k}^e = \int_{\Omega^e} \underline{B}^T \underline{D} \underline{B} \, d\Omega^e$$

\underline{k}^e = matriz de rigidez del elemento.

(2.19)

$$\underline{M}^e = \int_{\Omega^e} \rho \underline{N}^e \underline{N}^{eT} \, d\Omega^e$$

\underline{M}^e = matriz de masas del elemento.

(2.20)

$$\underline{f}_s^e = \int_{\Gamma^e} \underline{N}^{eT} \underline{q}_{(n)} \, d\Gamma^e$$

\underline{f}_s^e = vector de fuerzas de superficie del elemento.

(2.21)

$$\underline{f}_c^e = \int_{\Omega^e} \underline{N}^{eT} \underline{f}_d \, d\Omega^e$$

\underline{f}_c^e = vector de fuerzas de cuerpo del elemento.

(2.22)

al sustituir las ecuaciones de la 2.19 a la 2.22 en 2.18 se obtiene el modelo discreto de cada elemento, esto es:

$$\underline{k}^{eU} \underline{U}^e + \underline{M}^{eU} \underline{\ddot{U}}^e = \underline{f}_s^e + \underline{f}_c^e$$

(2.23)

para el caso en que únicamente las fuerzas que actúan son independientes del tiempo:

$$\underline{\ddot{U}}^e = 0$$

(2.24)

y la ecuación 2.23 para equilibrio estático queda:

$$\underline{k}^{eU} \underline{U}^e = \underline{f}_s^e + \underline{f}_c^e$$

(2.25)

en lo sucesivo el desplazamiento \tilde{U}^e se escribe U^e para la solución, una vez definida la ecuación de equilibrio de cada elemento (ecuación 2.27), estos elementos aislados se ensamblan para obtener las ecuaciones de equilibrio del medio continuo global la cual es:

$$\underline{K} \underline{U} = \underline{P}$$

(2.26)

en donde:

$$\underline{K} = \sum_{i=1}^{nelem} \underline{K}_i^e = \text{matriz de rigideces del continuo.}$$

(2.27)

\underline{U} = desplazamiento de toda la estructura en los puntos nodales.

(2.28)

$$\underline{P} = \sum_{i=1}^{nelem} (\underline{f}_{s_i}^e + \underline{f}_{c_i}^e) = \text{vector de cargas de toda la estructura.}$$

(2.29)

La ecuación 2.26 representa un sistema de ecuaciones algebraicas lineales simétrico no homogéneo en donde los desplazamientos son las incógnitas y cuya solución se describe en el capítulo 3.

2.2 Ecuaciones de forma.

En la solución aproximada considerada en las ecuaciones de equilibrio (ecuaciones 2.19 a la 2.22) se deben definir las funciones de forma, las cuales dependen de los siguientes factores:

- a) Geometría del dominio global.- Esto se refiere al tipo de espacio donde se va a integrar las ecuaciones, puede ser unidimensional, bidimensional y tridimensional.
- b) El grado de aproximación deseado en la solución.- Generalmente las funciones de forma son de interpolación, las cuales son polinomios con funciones exponenciales o trigonométricas. Si son polinomios lineales, únicamente se requieren los puntos nodales en las esquinas del elemento, si se utilizan polinomios cuadráticos se deben adicionar puntos nodales en la frontera del elemento.
- c) Facilidad de integración sobre el dominio del elemento.- La función de forma debe permitir la representación de cualquier forma lineal, de manera que se satisfaga el criterio de deformación constante.
- d) Las incógnitas deben presentar continuidad entre elementos.

Existen varias familias de funciones de forma que cumplen con las características anteriores (polinomial, lagrange, etc.) pero al considerar un grado de aproximación aceptable, se utiliza el elemento rectangular cuadrático de 8 puntos de la familia "serendipity", en la fig. 2.2 se

presenta este elemento con sus funciones de forma, la obtención de estas funciones de forma fueron deducidas por mera observación de tal manera que al sustituir en la función asociada al nodo su correspondiente coordenada tenga valor 1 y si en esa misma función se sustituye la coordenada de los otros puntos del elemento, su valor es cero.

Una vez definidas las funciones de forma, existe la posibilidad de mapear en dos sistemas de referencia elementos distorsionados (ver fig. 2.3) y esto es posible cuando existe una relación biunívoca entre coordenadas cartesianas y curvilíneas, un criterio de hacer estas transformaciones es mediante las funciones de forma. En este caso existen dos grupos de funciones de forma, unas para definir la geometría y por hacer el mapeo de un sistema de referencia a otro y otras para hacer la interpolación del elemento. Esto en forma esquemática se puede escribir así:

Aproximación

$$\underline{U} = \underline{N} \underline{U}^e$$

para n puntos nodales

Mapeo

$$\underline{X} = \underline{N}' \underline{\xi}$$

$$\underline{Y} = \underline{N}'' \underline{\eta}$$

sistema de referencia x - y
sistema de referencia $\xi - \eta$
para m puntos nodales.

Cuando se hacen coincidir los puntos nodales y las funciones de forma para la aproximación y el mapeo, a estos elementos se les conoce con el nombre de elementos isoparamétricos.

Al utilizar elementos isoparamétricos cuadráticos de la familia "serendipity", se deben definir las ecuaciones que efectúan ese mapeo y su influencia en las ecuaciones de equilibrio.

Las ecuaciones 2.19 a la 2.22 están referenciadas con respecto a un sistema local ξ, η para efectuar la transformación de estas ecuaciones al sistema de referencia x-y se considera la ecuación 2.30 para el cambio de coordenadas.

$$\begin{aligned} x_q &= N_i x_i + N_j x_j + N_k x_k + N_l x_l + N_m x_m + N_n x_n + N_o x_o + N_p x_p \\ y_q &= N_i y_i + N_j y_j + N_k y_k + N_l y_l + N_m y_m + N_n y_n + N_o y_o + N_p y_p \end{aligned}$$

$$x_q = i, j, k, l, m, n, o, p$$

(2.30)

en donde las funciones de forma están referenciadas al sistema ξ, η la ecuación 2.30 nos permite obtener los puntos en la referencia x-y donde se aplican los esfuerzos al resolver las ecuaciones de equilibrio en el sistema de referencia ξ, η .

Existe otra transformación a considerar, las derivadas de las funciones de forma, las cuales se deben de expresar en función de coordenadas locales ξ, η y cambiar convenientemente los límites de la integración, esta relación es:

$$\begin{bmatrix} \frac{\partial N_{\mathbf{q}}}{\partial x} \\ \frac{\partial N_{\mathbf{q}}}{\partial y} \end{bmatrix} = \underline{J}^{-1} \begin{bmatrix} \frac{\partial N_{\mathbf{q}}}{\partial} \\ \frac{\partial N_{\mathbf{q}}}{\partial} \end{bmatrix} \quad \mathbf{q} = \mathbf{i, j, k, l, m, n, o, p}$$

(2.31)

en donde:

$$\underline{J}^{-1} = \frac{1}{J} \begin{bmatrix} \frac{\partial y}{\partial} & -\frac{\partial y}{\partial} \\ -\frac{\partial x}{\partial} & \frac{\partial x}{\partial} \end{bmatrix} \quad J = \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \eta}$$

(2.32)

definida la relación de transformación (ecuación 2.31 a 2.32) se pueden expresar las ecuaciones 2.19 y 2.22 en función de coordenadas locales y se tiene:

$$\underline{k}^e = \int_{-1}^1 \int_{-1}^1 \underline{J}(\xi, \eta) \underline{B}^T(\xi, \eta) \underline{D} \underline{B}(\xi, \eta) d\xi d\eta$$

(2.33)

$$\underline{f}_c^e = \underline{t} \int_{-1}^1 \int_{-1}^1 \rho \underline{J}(\xi, \eta) \underline{N}^T(\xi, \eta) \underline{F}(\xi, \eta) d\xi d\eta$$

(2.34)

$$\rho \underline{v} = \rho v \begin{bmatrix} \cos \alpha \\ \text{sen } \alpha \end{bmatrix}$$

ρv = peso volumétrico del material.
 α = dirección de la gravedad en el sistema de referencia global.

(2.35)

Para efectuar la integración de las ecuaciones 2.33 , 2.34 se utiliza la cuadratura gaussiana por lo que las ecuaciones 2.33 y 2.34 quedan:

$$\underline{k}^e = t \sum_{m=1}^2 \sum_{n=1}^2 H_m H_n J(\eta_m, \eta_n) \underline{B}^T(\eta_m, \eta_n) \underline{D} \underline{B}(\eta_m, \eta_n)$$

(2.36)

$$\underline{f}_C^e = t \sum_{m=1}^2 \sum_{n=1}^2 H_m H_n J(\eta_m, \eta_n) \underline{N}^T(\eta_m, \eta_n) \underline{\theta}_f$$

(2.37)

Una vez definidas las ecuaciones a utilizar se menciona la metodología a seguir para obtener la solución.

2.3 Secuencia del método del elemento finito.

Supóngase una estructura que se va a analizar por los estados planos de esfuerzo, por lo que se utiliza un dominio bidimensional con elemento isoparamétrico lineal y se llevan a cabo los siguientes pasos:

1.- Se divide el continuo en un número finito de regiones o elementos por lo que se procede a elaborar una malla de la estructura formada por elementos cuadráticos (de ocho puntos) y puntos nodales, cada punto nodal admite dos grados de libertad, un desplazamiento en la dirección "x" y un desplazamiento en la dirección "y" por lo que cada elemento

está compuesto de dieciseis desplazamientos según se indica en la fig. 2.4.

2.- Una vez definida la malla se enumeran los nodos y los elementos en un sistema de referencia cartesiano global, se obtienen las coordenadas de cada punto nodal con sus restricciones frontera, esto significa que el punto nodal puede tener desplazamiento libre o restringido en alguno de sus componentes.

3.- Se obtiene la matriz de rigideces de cada elemento con la ecuación 2.36 con los valores correspondientes según los datos de cada elemento como lo son el tipo de material y coordenadas.

4.- Se obtiene el vector de cargas de cada elemento, según la discretización únicamente se aceptan cargas en los nodos, y cada uno tendrá, dos componentes F_{x_i} y F_{y_i} donde i es el número de nodo, esto significa que cada elemento tendrá un vector de cargas de dieciseis valores como se indica en la fig. 2.5.

5.- En el caso de considerar las fuerzas de cuerpo, éstas se calculan con la ecuación 2.37.

6.- Se procede al ensamble de la matriz de rigideces de la estructura. Definida la matriz de rigideces de cada elemento (matriz de 8×8), sus componentes ocupan una localidad en la matriz de rigideces global según el desplazamiento al que está asociado, en nodos comunes a dos elementos se sumarán en la misma localidad de la matriz de rigideces de la estructura, la contribución de cada elemento asociado a ese

desplazamiento. En forma esquemática se presenta un ejemplo desde la figura 2.6.1 a la 2.6.4. La forma de ensamble mediante el uso del indicador de ecuación para el programa de computadora se analiza en el capítulo 3.

7.- El ensamble del vector de cargas de la estructura consta del ensamble de las fuerzas de cuerpo y de las fuerzas de superficie. Las primeras se ensamblan en forma análoga a la matriz de rigideces en forma esquemática se presentan en la fig. 2.7 y las fuerzas que se aplican en los nodos según el desplazamiento asociado ocupan una localidad en \underline{P} , la forma de adición de estos elementos a \underline{P} queda establecida por la suma de las fuerzas de cuerpo más las fuerzas de superficie.

8.- Con la matriz de rigideces de la estructura y el vector de cargas se plantea la ecuación:

$$\underline{K} \underline{U} = \underline{P}$$

(2.38)

en donde el vector \underline{U} son desplazamientos. Esta ecuación representa un sistema de ecuaciones lineales algebraicas no homogéneas simétricas positivas definidas y \underline{U} son las incógnitas, el método de solución es el de Gauss-Crout con arreglo en silueta. En el capítulo 3 se detalla la obtención de estos algoritmos.

9.- Conocidos los desplazamientos de la estructura se obtienen los correspondientes desplazamientos asociados a

cada elemento y mediante la ecuación 2.9 se calculan los esfuerzos correspondientes.

Por la metodología de solución y el número de ecuaciones que se pueden llegar a tener es necesario el uso de la computadora, logrando realizar el trabajo en una computadora personal con las restricciones que la misma imponga, utilizando los aspectos numéricos que desde el punto de vista de computación controla la eficiencia del programa así como la organización más adecuada para que resulte óptimo en su ejecución.

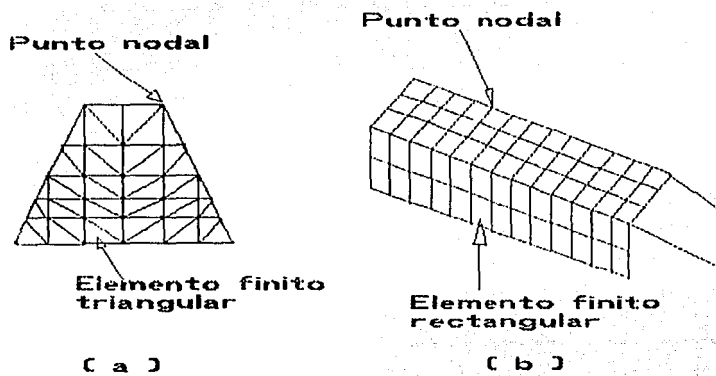


Figura 2.1 Idealización de un continuo por elemento finito:
a) Sección de la cortina de una presa.
b) Cubierta.

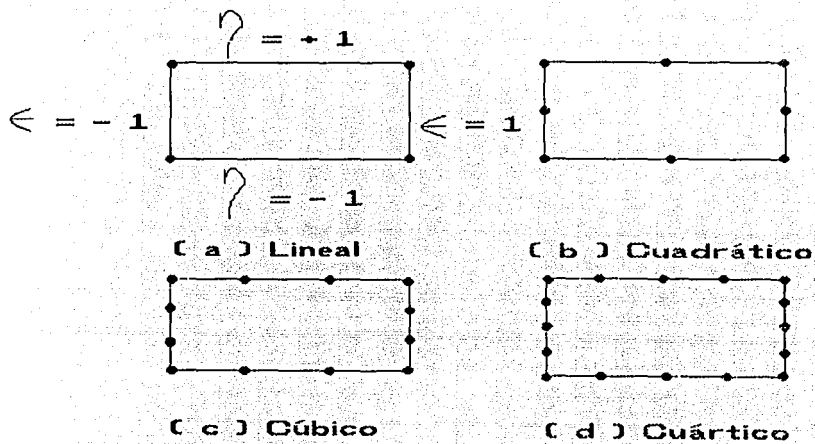
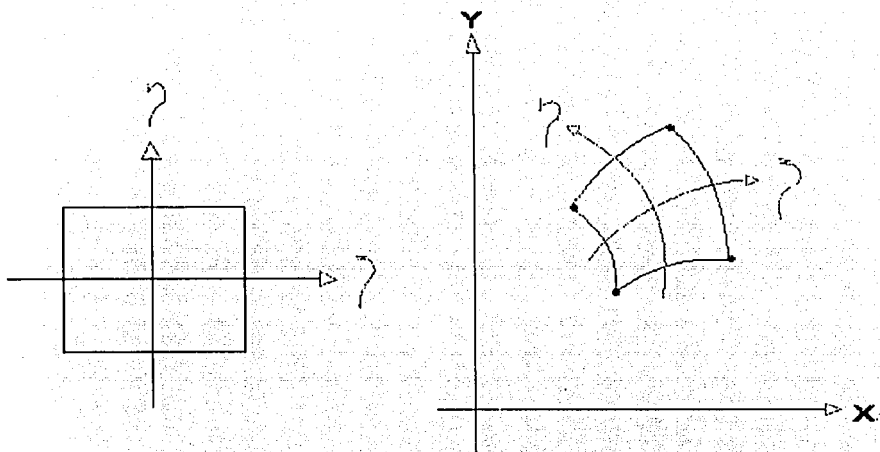
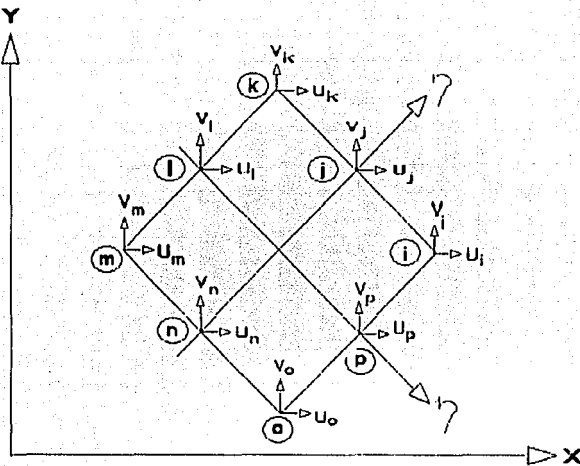


Figura 2.2 Funciones de forma para un elemento cuadrático de la familia "serendipity".



Representación Local Representación Cartesiana

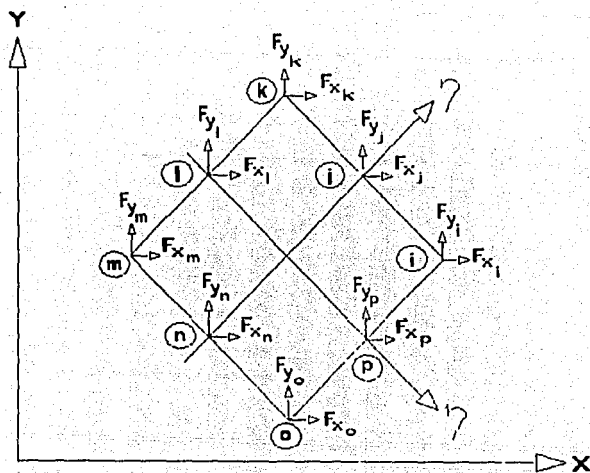
Figura 2.3 Transformación en dos dimensiones.



Vector de desplazamientos

$$\underline{U}^e = \begin{bmatrix} U_l \\ v_l \\ U_j \\ v_j \\ U_k \\ v_k \\ U_l \\ v_l \\ U_m \\ v_m \\ U_n \\ v_n \\ U_o \\ v_o \\ U_p \\ v_p \end{bmatrix} \quad (16 \times 1)$$

Figura 2.4 Representación del vector de desplazamientos para un elemento cuadrático.



$$\underline{f}_s^e = \begin{bmatrix} F_{x_l} \\ F_{y_l} \\ F_{x_j} \\ F_{y_j} \\ F_{x_k} \\ F_{y_k} \\ F_{x_l} \\ F_{y_l} \\ F_{x_m} \\ F_{y_m} \\ F_{x_n} \\ F_{y_n} \\ F_{x_o} \\ F_{y_o} \\ F_{x_p} \\ F_{y_p} \end{bmatrix}$$

Figura 2.5 Representación de las cargas de superficie para un elemento cuadrático.

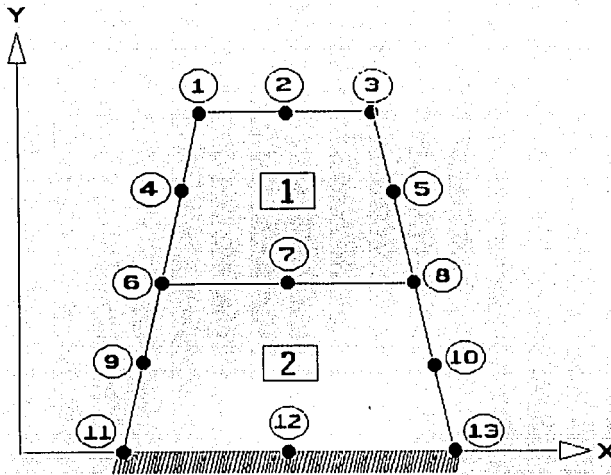


Figura 2.6.1 División en elementos cuadráticos del continuo formando una malla estructurada.

nodos	1	2	3	4	5	6	7	8	9	10	11	12	13
despl. X	1	3	5	7	9	11	13	15	17	19	0	0	0
despl. Y	2	4	6	8	10	12	14	16	18	20	0	0	0

Figura 2.6.2 Indicadores de ecuación de la estructura de la figura 2.6.1.

ELEMENTO 1

INDICADOR DE ECUACION

	5	6	3	4	1	2	7	8	11	12	13	14	15	16	9	10
5	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
6	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
11	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
12	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
13	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
14	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
15	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
16	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
9	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
10	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

INDICADOR DE ECUACION

ELEMENTO 2

INDICADOR DE ECUACION

	15	16	13	14	11	12	17	18	0	0	0	0	0	0	19	20
15	v	v	v	v	v	v	v	v							v	v
16	v	v	v	v	v	v	v	v							v	v
13	v	v	v	v	v	v	v	v							v	v
14	v	v	v	v	v	v	v	v							v	v
11	v	v	v	v	v	v	v	v							v	v
12	v	v	v	v	v	v	v	v							v	v
17	v	v	v	v	v	v	v	v							v	v
18	v	v	v	v	v	v	v	v							v	v
0																
0																
0																
0																
0																
0																
0																
19	v	v	v	v	v	v	v	v							v	v
20	v	v	v	v	v	v	v	v							v	v

INDICADOR DE ECUACION

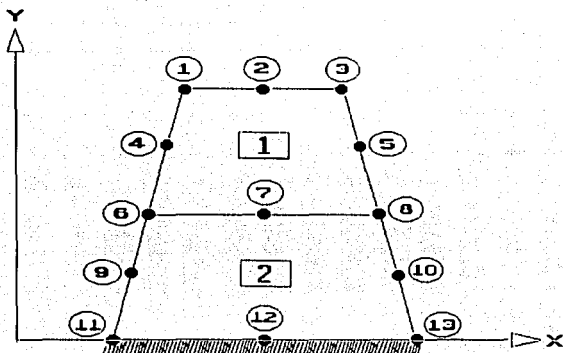
Figura 2.6.3 Matriz de rigideces del elemento 1 y 2.

INDICADOR DE ECUACION

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
2	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
3	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
4	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
5	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
6	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
9	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
10	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	
11	x	x	x	x	x	x	x	x	x	x	x+v	x+v	x+v	x+v	x+v	x+v	v	v	v	v
12	x	x	x	x	x	x	x	x	x	x	x+v	x+v	x+v	x+v	x+v	x+v	v	v	v	v
13	x	x	x	x	x	x	x	x	x	x	x+v	x+v	x+v	x+v	x+v	x+v	v	v	v	v
14	x	x	x	x	x	x	x	x	x	x	x+v	x+v	x+v	x+v	x+v	x+v	v	v	v	v
15	x	x	x	x	x	x	x	x	x	x	x+v	x+v	x+v	x+v	x+v	x+v	v	v	v	v
16	x	x	x	x	x	x	x	x	x	x	x+v	x+v	x+v	x+v	x+v	x+v	v	v	v	v
17	0	0	0	0	0	0	0	0	0	0	0	v	v	v	v	v	v	v	v	v
18	0	0	0	0	0	0	0	0	0	0	0	v	v	v	v	v	v	v	v	v
19	0	0	0	0	0	0	0	0	0	0	0	v	v	v	v	v	v	v	v	v
20	0	0	0	0	0	0	0	0	0	0	0	v	v	v	v	v	v	v	v	v

↑ INDICADOR DE ECUACION

Figura 2.6.4 Matriz de rigideces de la estructura.



$F_{x3} \leftarrow$ NODO

↑
DESPLAZAMIENTO
EN DIRECCION
X o Y.

VECTOR DE FUERZAS
DE CUERPO
DE LA ESTRUCTURA

ELEMENTO 1	ELEMENTO 2	VECTOR DE FUERZAS DE CUERPO DE LA ESTRUCTURA
5 F_{x3}	15 F_{x8}	1 F_{x1}
6 F_{y3}	16 F_{y8}	2 F_{y1}
3 F_{x2}	13 F_{x7}	3 F_{x2}
4 F_{y2}	14 F_{y7}	4 F_{y2}
1 F_{x1}	11 F_{x6}	5 F_{x3}
2 F_{y1}	12 F_{y6}	6 F_{y3}
7 F_{x4}	17 F_{x9}	7 F_{x4}
8 F_{y4}	18 F_{y9}	8 F_{y4}
11 F_{x6}	0 F_{x11}	9 F_{x5}
12 F_{y6}	0 F_{y11}	10 F_{y5}
13 F_{x7}	0 F_{x12}	11 $F_{x6} + F_{x6}$
14 F_{y7}	0 F_{y12}	12 $F_{y6} + F_{y6}$
15 F_{x8}	0 F_{x13}	13 $F_{x7} + F_{x7}$
16 F_{y8}	0 F_{y13}	14 $F_{y7} + F_{y7}$
9 F_{x5}	19 F_{x10}	15 $F_{x8} + F_{x8}$
10 F_{y5}	20 F_{y10}	16 $F_{y8} + F_{y8}$
		17 F_{x9}
		18 F_{y9}
		19 F_{x10}
		20 F_{y10}

↑ INDICADOR DE ECUACION ↑ INDICADOR DE ECUACION

Figura 2.7 Ensamble del vector de fuerzas de cuerpo de cada elemento.

3. SOLUCION DE ECUACIONES LINEALES.

Se tiene un sistema de ecuaciones algebraicas lineales no homogéneas de la forma:

$\underline{A} \underline{x} = \underline{b}$
(3.1)

donde:

\underline{A} = matriz de coeficientes cuadrada de $n \times n$.

\underline{b} = vector de coeficientes independientes.

\underline{x} = vector de incógnitas del sistema.

estos sistemas en ocasiones son extremadamente grandes por lo que se requiere trabajarlos en una computadora y utilizar métodos de solución que resulten eficientes. Los métodos de solución más conocidos se agrupan en iterativos y directos, los métodos directos se basan en la eliminación gaussiana y los que se adaptan a la computadora se denominan compactos, existen varios métodos directos compactos según las características de la matriz \underline{A} de la ecuación 3.1 (general, positiva definida), para el presente trabajo se desarrolla el método de Gauss-Crout para matrices simétricas en versión eficiente.

3.1 Método de Gauss-Crout.

Los métodos directos compactos se basan en un teorema del álgebra lineal el cual menciona que cualquier

matriz \underline{A} no singular, se puede descomponer en dos matrices triangulares, una superior \underline{U} y una inferior \underline{L} con la condición que una de ellas esté normalizada, esto es que los elementos de la diagonal principal sean iguales a la unidad, al apoyarse en este teorema se puede escribir:

$$\underline{A} = \underline{L} \underline{U}$$

(3.2)

al sustituir la ecuación 3.2 en 3.1 se tiene:

$$\underline{L} \underline{U} \underline{x} = \underline{b}$$

(3.3)

si se define:

$$\underline{U} \underline{x} = \underline{y}$$

(3.4)

en donde "y" se le llama vector auxiliar, al sustituir la ecuación 3.4 en 3.3 se tiene:

$$\underline{L} \underline{y} = \underline{b}$$

(3.5)

con las ecuaciones 3.2 , 3.4 y 3.5 se presenta el esquema general de la solución de ecuaciones lineales por los métodos compactos el cual consta de tres pasos:

a) Obtención de las matrices triangulares \underline{L} y \underline{U} conocida \underline{A} (ecuación 3.2) a este proceso se le conoce con el nombre de triangulación.

b) Obtención del vector \underline{y} conocida \underline{L} , \underline{b} (ecuación 3.5) a este proceso se le conoce con el nombre de sustitución hacia adelante.

c) Obtención del vector incógnita \underline{x} conocida \underline{y} , \underline{U} (ecuación 3.4) a este proceso se le conoce con el nombre de sustitución hacia atrás.

Para el caso de matrices simétricas se tiene la condición de la ecuación 3.6:

$$\underline{A} = \underline{A}^T$$

(3.6)

al aplicar la ecuación 3.6 en 3.2 y mediante transformaciones se tienen las siguientes expresiones:

a) Triangulación:

$$\underline{A} = \underline{U}^T \underline{D} \underline{U}$$

\underline{U} = matriz triangular superior normalizada.

\underline{D} = matriz diagonal.

(3.7)

b) Sustitución hacia adelante.

$$\underline{U}^T \underline{y} = \underline{b}$$

(3.8)

c) Sustitución hacia atrás.

$$\underline{D} \underline{U} \underline{x} = \underline{y}$$

(3.9)

al desarrollar las ecuaciones 3.7 , 3.8 y 3.9 se obtienen los siguientes algoritmos para la solución del sistema.

a) Triangulación

$$A_{11} = D_{11}$$

(3.10)

Primer paso:

$$G_{1j} = A_{1j} \quad j = 2, \dots, N$$

(3.11)

$$G_{ij} = A_{ij} - \sum_{k=1}^{i-1} U_{ki} G_{kj} \quad \begin{array}{l} i = 2, \dots, j-1 \\ j = 2, \dots, N \end{array}$$

(3.12)

Segundo paso:

$$U_{ij} = G_{ij} / D_{ii} \quad \begin{array}{l} j = 2, \dots, N \\ i = 2, \dots, j-1 \end{array}$$

(3.13)

$$D_{jj} = A_{jj} - \sum_{k=1}^{j-1} U_{kj} G_{kj} \quad j = 2, \dots, N$$

(3.14)

b) Sustitución hacia adelante.

$$y_1 = -b_1$$

(3.15)

$$y_j = b_j - \sum_{k=1}^{j-1} U_{kj} y_k \quad j = 2, \dots, N$$

(3.16)

c) Sustitución hacia atrás.

$$x_n = y_n / D_{nn}$$

(3.17)

$$x_j = (y_j - \sum_{k=j+1}^n D_{jj} U_{jk} x_k) / D_{jj} \quad j = n-1, \dots, 1$$

(3.18)

3.2 Arreglos para almacenamiento en memoria.

La matriz A de la ecuación 3.7 en principio da lugar a arreglos cuadrados bidimensionales, pero existen otro tipo de arreglos que aprovechan la simetría y la disposición de los coeficientes de esta matriz para lograr un ahorro en el uso de la memoria central, así como en las operaciones asociadas a dichos algoritmos (ecuaciones 3.10 a 3.17) este tipo de arreglos son los rectangulares (en banda) y los unidimensionales.

En los arreglos rectangulares (en banda) de acuerdo a la fig. 3.1 los coeficientes no nulos de una matriz A se encuentran alojados a lo largo de una franja paralela a la diagonal principal limitada por el contorno de banda, y se puede almacenar en un arreglo rectangular con el número de columnas igual al ancho de banda según se muestra en la fig. 3.2 y al escribirla por renglones y columnas se obtiene un arreglo rectangular mostrado en la fig. 3.3.

Al observar la matriz A de la fig. 3.1 y al compararla con la fig. 3.3 se aprecia que los

algoritmos de solución (ecuaciones 3.10 a 3.17) se modifican con una nueva localidad y las operaciones se limitan únicamente hasta el ancho de banda, en este arreglo como se indica en la fig. 3.2 existen localidades en la banda que no forman parte de la matriz A y se toman en cuenta en el dimensionamiento del arreglo esto da lugar a un desperdicio de memoria al utilizar este tipo de arreglo.

El arreglo unidimensional almacena la matriz A en un vector como se observa en la fig. 3.4 y cuenta con las siguientes características:

a) Se define un contorno de silueta que se forma a partir del primer elemento diferente de cero de cada columna que considera los coeficientes que se van a guardar en un arreglo unidimensional indicado por NEA (número de elementos de A).

b) Para poder identificar los elementos del arreglo unidimensional con el arreglo cuadrado se necesitan los datos del vector MD formado con las localidades que ocupan los elementos de la diagonal principal. La equivalencia de un coeficiente en un arreglo cuadrado con un arreglo unidimensional es:

arreglo cuadrado $A_{ij} = A_m$ arreglo unidimensional.

$$\begin{aligned} i &= 1, \dots, N \\ j &= 1, \dots, N \\ m &= i + \underline{MD}(j) - j \end{aligned}$$

(3.19)

c) En este arreglo no hay localidades desperdiciadas a diferencia de otros arreglos (en banda, cuadrado).

Desde el punto de vista de programación al comparar el arreglo unidimensional con el arreglo cuadrado y el arreglo en banda se puede analizar los siguientes puntos:

1.- En lo referente a memoria central, al observar la matriz de la fig. 3.1 en donde el arreglo cuadrado es de 324 localidades, en el bandeado es de 180 y el unidimensional es de 117.

2.- Puesto que en el contorno de silueta se eliminan ceros que reflejan operaciones no realizadas, se puede afirmar que los arreglos unidimensionales conducen a un ahorro en el tiempo de procesamiento.

Esto da como conclusión que el arreglo unidimensional es el más eficiente de los arreglos vistos.

3.3 Solución de ecuaciones lineales empleando arreglos unidimensionales.

Para modificar los algoritmos para arreglos unidimensionales se va a utilizar el vector auxiliar MD, debido a que conocido este dato es posible localizar cualquier elemento de la matriz A en la localidad del arreglo vector y se pueden definir los límites para efectuar la triangulación y sustitución del método de Gauss-Crout versión eficiente.

Al modificar los algoritmos de solución para arreglo unidimensional se tiene:

a) Triangulación.

Con base en las ecuaciones 3.10 a 3.14 y conocido el vector auxiliar MD:

$$A_1 = D_1$$

(3.20)

$$U_2 = A_2 / D_1 \quad \forall MD(2) \neq 2$$

(3.21)

$$U_3 = A_3 - D_1 U_2^2 \quad \forall MD(2) \neq 2$$

(3.22)

$$U_f = A_f / D_p$$

(3.23)

$$j = 3, \dots, n$$

$$f = MD_{j-1} + 1$$

$$p = MD$$

$$\theta = j + MD_{j-1} - MD_j + 1$$

$$\text{X } MD_j - MD_{j-1} - 1 = 0 = U_f = A_f$$

$$U_u = (A_u - \sum_{k=1}^{i-1} U_v \cdot D_{MD_k} U_w) / D_{MD_i}$$

(3.24)

$$j = 3, \dots, N$$

$$i = q, \dots, j-1$$

$$q = j - MD_j + MD_{j-1} + 2$$

$$t = i - MD_j + (MD_{j+1} + 1)$$

$$\theta = j - MD_j + (MD_{j+1} + 1)$$

$$U = i - j + MD_j$$

$$V = k - t + MD_{i-1} + 1$$

$$W = k - \theta + MD_{j-1} + 1$$

$$\text{X } MD_j - MD_{j-1} - 1 = 0 \quad U_u = A_u$$

$$\text{X } j - 1 \quad q \quad U_u = A_u$$

$$\text{X } MD_i - MD_{i-1} - 1 = 0 \quad U_u = A_u$$

$$D_{MD_j} = A_{MD_j} \sum_{k=\theta}^{j-1} U_{MD_k} D_w$$

(3.25)

$$j = 3, \dots, n$$

$$\theta = j - MD_j + (MD_{j+1} + 1)$$

$$w = k - \theta + MD_{j-1} + 1$$

$$\forall MD_j - MD_{j-1} - 1 = 0 \quad D_{MD_j} = A_{MD_j}$$

b) Sustitución hacia adelante.

Al tener las ecuaciones 3.15 y 3.16 y el vector

MD:

$$y_1 = b_1$$

(3.26)

$$y_j = b_j - \sum_{k=a}^d U_k y_c$$

(3.27)

$$j = 2, \dots, n$$

$$a = MD_{j-1} + 1$$

$$d = MD_{j-1}$$

$$c = k + j - MD_j$$

$$f_j = d - a$$

$$\forall f_j < 0 \quad y_i = b_j$$

c) Sustitución hacia atrás.

$$X_n = Y_n / D_{nn}$$

$$X_j = (Y_j - \sum_{k=j+1}^n D_{jk} U_{jk} x_k) / D_{jj}$$

$$j = n-1, \dots, 1$$

(3.28)

La sustitución hacia atrás de la matriz A en arreglo unidimensional se presenta en forma matricial en la fig. 3.5.

Al efectuar las operaciones correspondientes y despejar los elementos del vector X (incógnitas) queda desarrollado como se presenta en la fig. 3.6.

Al analizar la figura para este tipo de arreglo, resulta más sencillo efectuar esta sustitución por columnas e ir sumando a cada elemento del vector X según sea el renglón correspondiente, por lo que se debe definir para cada columna los límites para efectuar las operaciones correspondientes. En la columna 5 de la fig. 3.6 no se efectúa operación alguna.

Al comparar la ecuación 3.1 con el modelo

matemático a resolver (ecuación 3.26) se observa la similitud que existe, en donde la matriz A representa la matriz de rigideces K , el vector X (incógnitas) representa el vector de desplazamientos U y el vector b es el vector de cargas P . Debido a la metodología de la solución enunciada en el capítulo anterior, existen otras ventajas al utilizar el método de Gauss-Crout.

En el proceso de triangulación únicamente se necesita la matriz de rigideces de la estructura, la cual es independiente de las cargas que se apliquen, esto significa que para cada estructura que se va a resolver, únicamente se debe de hacer una vez el proceso de triangulación y con ésta se pueden hacer tantas sustituciones (paso b y c del esquema de los métodos compactos) con vectores de carga P que se quiera resolver, ésta da un ahorro también en el tiempo de proceso en comparación con otros esquemas de solución, por otro lado la triangulación representa ochenta y ocho por ciento del tiempo de proceso total, en tanto que las sustituciones únicamente el doce por ciento.

Otra ventaja que se presenta es el dimensionamiento debido no sólo al tipo de arreglo unidimensional sino también a la disposición de las operaciones, las localidades que se utilizan para almacenar la matriz de rigideces K , también se utilizan para almacenar el vector de cargas P se utiliza para guardar el vector auxiliar y y posteriormente guarda el vector de

desplazamientos \underline{U} , es decir el vector solución, por lo que únicamente se deben dimensionar la matriz de rigideces \underline{K} , el vector de cargas \underline{P} y el vector \underline{MD} .

El programa de computadora en que se realiza el proceso de solución de ecuaciones, se divide en dos partes, una es la triangulación y la segunda la sustitución hacia adelante y hacia atrás.

FALLA DE ORIGEN

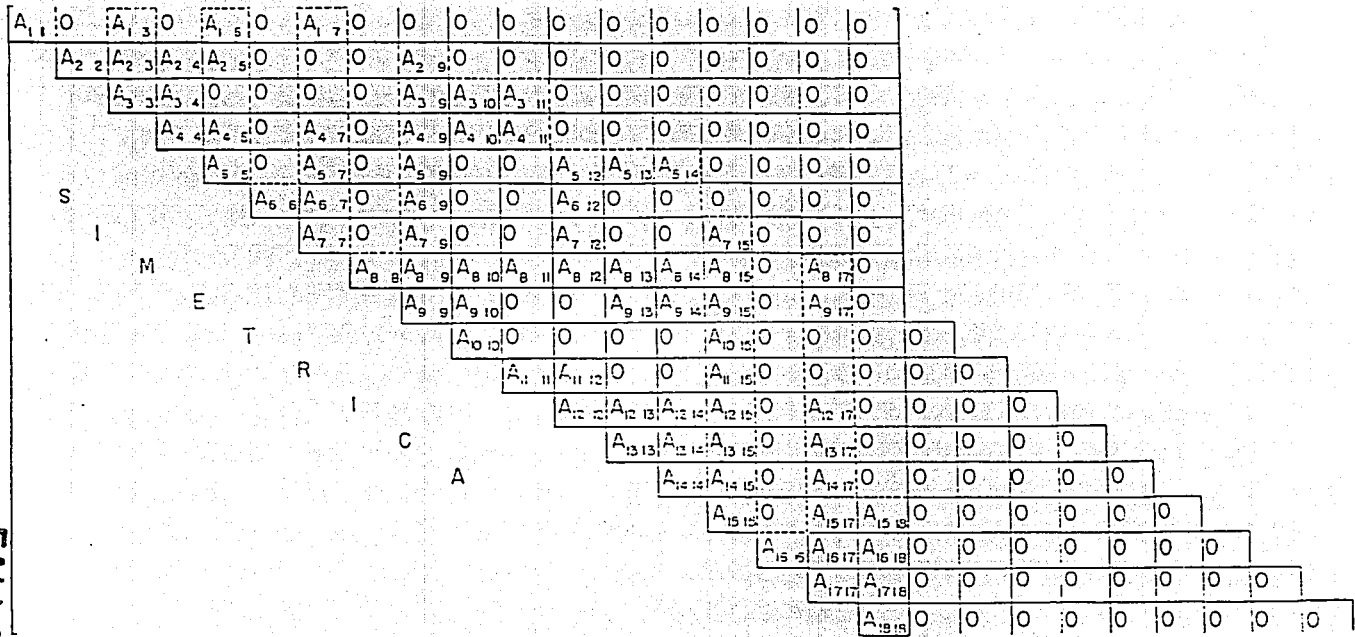


Figura 3.1 Representación esquemática de la matriz A , con diferentes tipos de almacenamientos

Ancho de banda = 13AN

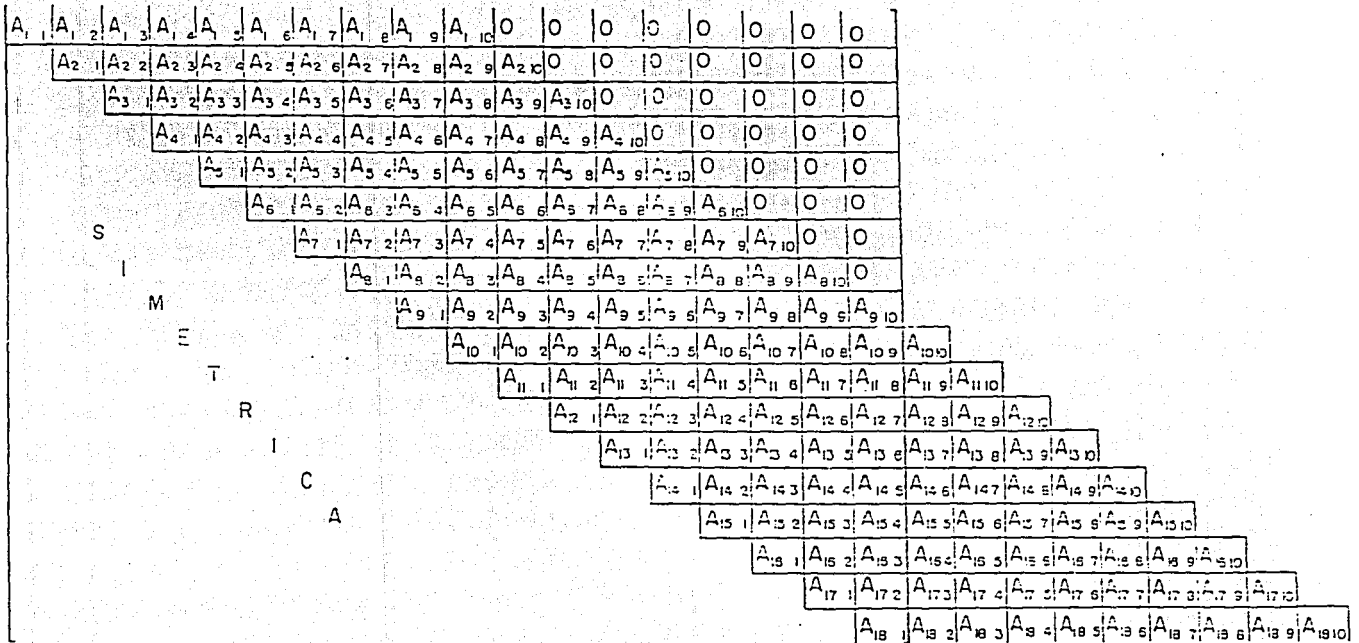


Figura 3.2 Organización de los coeficientes de la matriz A , en arreglos en banda (rectangular).

NBAN

$$\underline{A} \text{ (NBAN)} =$$

$A_{1,1}$	0	$A_{1,3}$	0	$A_{1,5}$	0	$A_{1,7}$	0	0	0
$A_{2,2}$	$A_{2,3}$	$A_{2,4}$	$A_{2,5}$	0	$A_{2,7}$	0	$A_{2,9}$	0	0
$A_{3,3}$	$A_{3,4}$	$A_{3,5}$	0	$A_{3,7}$	0	$A_{3,9}$	$A_{3,10}$	$A_{3,11}$	0
$A_{4,4}$	$A_{4,5}$	0	$A_{4,7}$	0	$A_{4,9}$	$A_{4,10}$	$A_{4,11}$	$A_{4,12}$	0
$A_{5,5}$	0	$A_{5,7}$	0	$A_{5,9}$	$A_{5,10}$	$A_{5,11}$	$A_{5,12}$	$A_{5,13}$	$A_{5,14}$
$A_{6,6}$	$A_{6,7}$	0	$A_{6,9}$	$A_{6,10}$	$A_{6,11}$	$A_{6,12}$	$A_{6,13}$	$A_{6,14}$	0
$A_{7,7}$	0	$A_{7,9}$	$A_{7,10}$	$A_{7,11}$	$A_{7,12}$	$A_{7,13}$	$A_{7,14}$	$A_{7,15}$	0
$A_{8,8}$	$A_{8,9}$	$A_{8,10}$	$A_{8,11}$	$A_{8,12}$	$A_{8,13}$	$A_{8,14}$	$A_{8,15}$	0	$A_{8,17}$
$A_{9,9}$	$A_{9,10}$	$A_{9,11}$	$A_{9,12}$	$A_{9,13}$	$A_{9,14}$	$A_{9,15}$	0	$A_{9,17}$	0
$A_{10,10}$	$A_{10,11}$	$A_{10,12}$	$A_{10,13}$	$A_{10,14}$	$A_{10,15}$	0	$A_{10,17}$	0	0
$A_{11,11}$	$A_{11,12}$	$A_{11,13}$	$A_{11,14}$	$A_{11,15}$	0	$A_{11,17}$	0	0	0
$A_{12,12}$	$A_{12,13}$	$A_{12,14}$	$A_{12,15}$	0	$A_{12,17}$	0	0	0	0
$A_{13,13}$	$A_{13,14}$	$A_{13,15}$	0	$A_{13,17}$	0	0	0	0	0
$A_{14,14}$	$A_{14,15}$	0	$A_{14,17}$	0	0	0	0	0	0
$A_{15,15}$	0	$A_{15,17}$	$A_{15,18}$	0	0	0	0	0	0
$A_{16,16}$	$A_{16,17}$	$A_{16,18}$	0	0	0	0	0	0	0
$A_{17,17}$	$A_{17,18}$	0	0	0	0	0	0	0	0
$A_{18,18}$	0	0	0	0	0	0	0	0	0

Figura 3.3 Matriz de rigideces en arreglo en banda

$$\begin{array}{cccccccc}
 \left[\begin{array}{cccccccc}
 D_1 & D_1 U_2 & 0 & D_1 U_6 & 0 & 0 & 0 & 0 \\
 & D_3 & D_3 U_4 & D_3 U_7 & 0 & 0 & 0 & 0 \\
 & & D_5 & D_5 U_8 & 0 & D_5 U_{11} & 0 & 0 \\
 & & & D_9 & 0 & D_9 U_{12} & 0 & 0 \\
 & & & & D_{10} & D_{10} U_{13} & 0 & D_{10} U_{17} \\
 & & & & & D_{14} & D_{14} U_{15} & D_{14} U_{18} \\
 & & & & & & D_{16} & D_{16} U_{19} \\
 & & & & & & & D_{20}
 \end{array} \right]
 \begin{array}{c}
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 x_5 \\
 x_6 \\
 x_7 \\
 x_8
 \end{array}
 =
 \begin{array}{c}
 y_1 \\
 y_2 \\
 y_3 \\
 y_4 \\
 y_5 \\
 y_6 \\
 y_7 \\
 y_8
 \end{array}
 \end{array}$$

Figura 3.5 Elementos para la sustitución hacia atrás en el método de Gauss-Crout y arreglos unidimensionales compactos.

$$\begin{array}{rcl}
 x_1 = y_1 / D_1 & -0 & -0 \quad -0 \quad -0 \quad -u_6 x_4 \quad -0 \quad -u_2 x_2 \\
 x_2 = y_2 / D_3 & -0 & -0 \quad -0 \quad -0 \quad -u_7 x_4 \quad -u_4 x_3 \\
 x_3 = y_3 / D_5 & -0 & -0 \quad -u_{11} x_6 \quad -0 \quad -u_8 x_4 \\
 x_4 = y_4 / D_9 & -0 & -0 \quad -u_{12} x_6 \quad -0 \\
 x_5 = y_5 / D_{10} & -0 & -0 \quad -u_{13} x_6 \\
 x_6 = y_6 / D_{14} & -u_{17} x_8 & -u_{15} x_7 \\
 x_7 = y_7 / D_{16} & -u_{18} x_8 \\
 x_8 = y_8 / D_{20} & &
 \end{array}$$

Figura 3.6 Desarrollo de la sustitución hacia atrás por el método de Gauss-Crout, en arreglo unidimensional compacto.

4.- CONSIDERACIONES PREVIAS EN CUANTO AL ENTORNO DE LA PROGRAMACION DEL SISTEMA.

En este capítulo se pretende dar un breve panorama acerca de los aspectos más sobresalientes que se han tomado en cuenta al realizar la programación de este sistema en particular.

Dado que el sistema está enfocado al área de ingeniería civil; impone el requerimiento de la realización de un programa de computadora para la solución de estados planos de esfuerzos en forma eficiente para una computadora personal; ello involucra la manipulación de datos de una manera eficiente, lo cual nos lleva a poder producir el mismo fenómeno bajo diversas circunstancias de una manera fácil y práctica siendo lo más aproximado posible a la realidad.

Todo esto nos lleva a pensar en la utilización de un equipo de cómputo formado con todas las herramientas posibles para la realización de dicho sistema y bajo el cual funcionen en todos y cada uno de sus aplicaciones para las que se requiere su desarrollo. Aunado a esto se contempla un análisis previo a la utilización de un lenguaje de programación que por demás contribuya en gran medida al desarrollo del sistema.

A continuación describiremos tanto el equipo de cómputo que se ha de emplear para la realización y aplicación del sistema como los lenguajes propuestos bajo los cuales el sistema ha sido desarrollado.

4.1 Equipo de cómputo a utilizar.

El hablar de computadoras y poder identificar el tipo de herramienta en computación con que se cuenta tiene gran importancia; ya que depende en gran medida de los elementos que la componen y que hacen de ella un instrumento útil para la aplicación específica que se requiere.

Los sistemas de computadoras pueden ser clasificados en micro, mini, macro y supercomputadoras en razón de su tamaño, costo y desempeño.

MICROCOMPUTADORAS

Es el sistema más pequeño de propósito general que puede ejecutar instrucciones de un programa para llevar a cabo una amplia variedad de tareas. Un sistema de microcomputadora tiene todos los elementos funcionales que se encuentran en un sistema grande. Está organizado para llevar a cabo el almacenamiento, la lógica aritmética, el control y las funciones de salida. La mayoría de las microcomputadoras son unidades compactas y tan ligeras que pueden ser trasladadas con facilidad. Están diseñadas para ser utilizadas por una sola persona. Además de la CPU, la microcomputadora común tiene un tablero para que el operador introduzca la información. Grabadoras de cinta magnética y/o lectoras de disco flexible se utilizan para introducir datos, programas y para recibir la información procesada. Se utilizan pequeñas cintas magnéticas y discos flexibles para

el almacenamiento secundario fuera de línea. Una pantalla de despliegue visual y/o una impresora de caracteres se utilizan para preparar la salida en una forma legible para el humano.

En un sentido amplio, podemos llegar a incluir a las computadoras personales bajo el nombre de microcomputadoras ya que generalmente lo que caracteriza a una computadora personal es su bajo costo y portabilidad entre otras cosas siendo que todas ellas se encuentran conformadas básicamente por un monitor, una CPU (Control Processing Unit) y un teclado. Estas computadoras satisfacen en gran parte las necesidades cotidianas para las que fueron adquiridas, ello se ha reflejado en el aumento de su accesibilidad a la hora de la compra de estos equipos.

MINICOMPUTADORA

Es una máquina pequeña de propósito general. Puede variar en el tamaño desde un modelo instalable sobre el escritorio, hasta una unidad con más o menos el tamaño de un archivero de cuatro gavetas. Los sistemas minis típicos sobrepasan a una microcomputadora en capacidad de almacenamiento, velocidad de operaciones aritméticas y capacidad de soportar gran variedad de dispositivos periféricos de rápida operación. Los sistemas minis pueden ser diseñados para manejar en forma simultánea las necesidades de proceso de varios usuarios. Por lo tanto las minicomputadoras se emplean normalmente en las organizaciones.

MACROCOMPUTADORAS

Es un sistema grande donde puede ser usado como el computador anfitrión central en una red de procesamiento distribuido de datos. La macro se comunica con y ejerce control sobre procesadores satélites más pequeños. Sin embargo estos equipos continúan desempeñando un papel muy importante dentro de la ingeniería principalmente, ya que las máquinas grandes representan una elección óptima en la realización de grandes cálculos matemáticos y por ende en el procesamiento de grandes cantidades de datos, situaciones que suelen enfrentar en varias ocasiones los ingenieros. No obstante, actualmente, se puede llegar a realizar un trabajo preliminar de gran escala en una microcomputadora, para que siendo así se pueda hacer uso de una línea de telecomunicaciones y poder transferir el programa a una macro o supercomputadora e implantarlo fácilmente.

SUPERCOMPUTADORAS

Son las computadoras más grandes ó más rápidas y más caras que se han llegado a fabricar.

Retomando todo lo anterior, podemos inclinarnos por el uso de una microcomputadora para nuestra aplicación, considerando las ventajas que proporciona sobre las demás opciones para facilitarnos la tarea de realización de nuestro trabajo a desarrollar. La microcomputadora o computadora

personal que nos servirá de herramienta para la creación del software o programas de aplicación que componen al sistema a desarrollar será la siguiente: es una computadora personal XT de Printaform, con un procesador 386 a 33 Mhz con una capacidad en memoria RAM de 2 megas, suficiente para soportar a los compiladores de los lenguajes de programación a utilizar, además de contar con un disco duro de 80 megas facilitándonos la velocidad de acceso a disco y a su vez contando con el espacio suficiente en disco para el almacenamiento de los paquetes que nos servirán de herramienta para la programación y edición del trabajo de investigación a desarrollar. El equipo de cómputo con que contamos tiene dos unidades de disco flexibles, la primera de ellas es de capacidad de 360 Kb lo que representa el empleo de discos de 5 1/4 de baja densidad y la otra es de capacidad de 1.44 Mb manejando discos de 3 1/2 de alta densidad. Finalmente el equipo se compone de un monitor VGA a color, un teclado de 101 teclas, un ratón y una impresora Star Micronics NX1001.

Con el equipo provisto anteriormente contamos con lo indispensable para la programación del sistema, en cuanto a la implementación del sistema podrá ser ejecutado en una microcomputadora con las mismas características anteriores o con un equipo más sofisticado que el anterior.

4.2 Análisis de la versatilidad en los lenguajes propuestos para la programación del sistema.

Una vez formada la idea del equipo de computo necesario para operar junto con el sistema es necesario puntualizar en el aspecto del lenguaje de programación bajo el cual correrá a cargo el diseño del programa del sistema.

Una previa investigación a esto; refleja la importancia del lenguaje de programación FORTRAN en el desarrollo de programas de ingeniería estructural, dado que la mayoría de los programas del elemento finito utilizan este lenguaje. Para el caso de la microcomputadora el compilador FORTRAN posee proposiciones que lo hacen más potente tal es el caso de su rapidez, alta precisión en los cálculos matemáticos, etc. pero claro, tiene algunas desventajas al trabajar en forma interactiva, esto es razonable ya que su configuración fue hecha para trabajar con lectura de tarjetas para la entrada de datos desde sus inicios. Para resolver este inconveniente de interactividad se ha escogido al lenguaje de programación C como la herramienta adecuada ante el manejo de archivos y su contemplación de interactividad en todo momento, por todo esto se busca complementar ambos lenguajes para poder aprovechar sus herramientas de una manera práctica facilitándonos la tarea de programación y cumpliendo a su vez con los objetivos que han sido planteados desde sus inicios de una manera plena.

A continuación se enumeran las ventajas y desventajas tanto de FORTRAN como de C, no sin antes recordar que las

limitaciones presentadas por alguno de estos dos lenguajes serán resueltas por las opciones brindadas por el otro.

4.2.1 Ventajas y desventajas en FORTRAN.

Debido a que los cálculos del método requieren de operaciones simples de entrada y salida de datos, manejo de pequeñas cantidades de los mismos (en algunos casos) y una cantidad intrincada de cálculos, se ha contemplado la posibilidad de utilizar un lenguaje que haya sido diseñado precisamente para efectuar este tipo de operaciones matemáticas.

Por esta razón se ha tomado en cuenta al lenguaje FORTRAN cuyo nombre se compone de las palabras traducción de fórmulas (FORmula TRANslation) ya que uno de sus rasgos distintivos es que usa una notación para facilitar la escritura de fórmulas matemáticas en su empleo de la programación. Gracias a esta y muchas otras características, FORTRAN es el lenguaje propio de la mayoría de los ingenieros y científicos que han adquirido experiencia en máquinas grandes. Esto ha representado a FORTRAN como un lenguaje fundamental asociado con la ingeniería en las instalaciones de supercomputadoras.

A partir de su utilidad en cálculos, la asequibilidad de FORTRAN también ha contribuido a su popularidad, prueba de ello se ha reflejado en la existencia de diversos compiladores de FORTRAN para diversos tipos y tamaños de computadoras, incluso las microcomputadoras que originalmente

eran demasiado limitadas para manejar un compilador completo de FORTRAN han evolucionado a un punto tal que permiten utilizarlo.

Es importante mencionar que el desenvolvimiento del lenguaje FORTRAN a lo largo del tiempo propició el surgimiento de diversas versiones, las cuales se adhieren a un conjunto común de proposiciones y de sintaxis todas dentro de un mismo contexto.

Hablando de FORTRAN en general resulta interesante analizar los siguientes aspectos técnicos:

- 1) Es preciso en los cálculos complicados.
- 2) Es rápido ya que como el compilador que es, crea un módulo o código objeto una vez que el programa fuente está exento de errores y por ello puede usarse el módulo objeto compilado una y otra vez sin tener que repetir el proceso de compilación, lo cual ahorra tiempo y enfatiza aún más la posibilidad de que un programa compilado corra mucho más rápidamente que los interpretados.
- 3) Se han establecido estándares mínimos y versiones extendidas por lo que existe una variación entre sus dialectos.
- 4) Es relativamente poderoso en lo que respecta a las operaciones de entrada/salida.

En contraparte hay que pensar en algunos inconvenientes que llega a presentar dicho lenguaje:

- 1) No es un buen lenguaje de aprendizaje como el BASIC.

2) Su orientación no es interactiva lo que lo hace relativamente complejo.

3) Al compilar un programa fuente libre de errores se crea un módulo objeto, el cual es almacenado en forma de archivo en el disco ocupando espacio libre para otras aplicaciones.

4) Los gráficos no forman parte integrada del lenguaje.

Una vez ya analizadas las ventajas y desventajas que muestra el empleo del lenguaje FORTRAN en la programación del sistema, podemos observar que la facilidad para establecer una interactividad entre el sistema y el usuario del mismo resulta ser mínima; por lo que a este inconveniente se le ha buscado una solución mediante la utilización del lenguaje C, buscando implementar el lenguaje C en apoyo a la captura de los datos que requiere el sistema para trabajar, logrando realizar el trabajo restante por FORTRAN, el cual será el encargado de llevar a cabo todos los cálculos matemáticos con la aproximación más rápida que nos ofrezca.

4.2.2 Ventajas y desventajas en el lenguaje C.

El lenguaje C es transferible y de aplicación general con características de expresiones concisas y un diseño que permite elaborar programas estructurados.

Originalmente fue desarrollado para escribir software de sistemas por ejemplo el sistema operativo UNIX se escribió en C, sin embargo tiene características de lenguaje de alto nivel y por tanto tiene una amplia gama de aplicaciones.

Para utilizar al lenguaje C en un equipo de cómputo se requiere de la siguiente configuración mínima:

- 1) Un ordenador IBM PC o compatible con un sistema operativo DOS versión 3.0 o superior ó un OS/2 versión 1.1 o superior.
- 2) Un procesador mínimo 8088 de 8 Mhz. Se recomienda un procesador 80286.
- 3) 512k de memoria en RAM disponible bajo DOS. Se recomienda 1Mb.
- 4) 3Mb de memoria en RAM disponible bajo OS/2. Se recomienda 4Mb.
- 5) 384k de memoria extendida, por si tenemos que depurar programas largos.
- 6) Una unidad de disco duro con un mínimo de 8Mb disponibles y una unidad de disco flexible.

A continuación presentamos una serie de puntos a considerar en el empleo del lenguaje C:

- 1) Es legible dada la programación estructurada que se puede emplear en este lenguaje.
- 2) Es compacto y flexible.
- 3) Es versátil dado que se considera un lenguaje de nivel medio.
- 4) Es veloz ya que los programas trabajan más rápidamente bajo el ambiente del compilador.
- 5) Resulta ser portable puesto que se puede llevar de un equipo a otro.

- 6) Nos permite aprovechar las características del hardware del equipo de cómputo sin tener necesariamente un conocimiento exacto del mismo.
- 7) Al trabajar con un programa fuente en módulos o proyectos logramos que el compilador emplee la memoria provista para la compilación del programa de una manera más eficiente.
- 8) Permite la conversión de los tipos de datos entero, real, carácter.
- 9) Facilita la manipulación directa de bits, bytes, palabras y punteros principalmente en el manejo de parámetros y manipulaciones de archivos.
- 10) Almacena en disco el archivo ejecutable del programa fuente.

Por otro lado podemos citar lo siguiente:

- 1) Presenta una falta de estándares y reglas del compilador ya que existen muchas versiones del lenguaje C con pequeñas diferencias entre si lo que es el ANSI común.
- 2) El compilador muestra ciertos errores que llegan a ser consecuencias de otros dentro del mismo programa fuente, lo que provoca en ocasiones confusiones en una primer instancia ya que al solucionar uno de los dos errores conseguimos solucionar el segundo automáticamente.
- 3) Elabora un archivo objeto en disco con la finalidad de llegar a construir el archivo ejecutable del programa fuente.

Finalmente se puede asegurar que el problema de interactividad tendrá una adecuada solución al emplear el lenguaje C, valiéndonos de las funciones propias de dicho

lenguaje, para que siendo así, se logre obtener un amplio control tanto en el hardware como en la manera de guiar al usuario en el transcurso de la ejecución del sistema siendo esto de la manera más práctica posible.

4.2.3 Manipulación de archivos.

La manipulación de archivos, para este caso es muy importante, ya que el usuario tendrá que introducir datos en la entrada del proceso del sistema y dado que no son pocos el volver a introducir los mismos datos en diferentes ocasiones resulta ser algo tedioso además de emplear demasiado tiempo en ello.

Para esto se busca tener la opción de almacenar dichos datos en un archivo en disco ya sea flexible ó duro, con la finalidad de poder llamar al archivo existente con esos datos sin la necesidad de volver a introducirlos uno por uno logrando así una mayor efectividad en el proceso de rapidez en el manejo del sistema.

Se utiliza el manejo de archivos con un acceso secuencial tratando de lograr obtener una mayor seguridad en la lectura/escritura de los datos y además permitiendo la posibilidad de cambiar de dispositivo de almacenamiento según sea el caso.

Se contempla la manipulación de archivos con miras de impresión para una posterior verificación de los datos

introducidos por el usuario; ya que esta actividad suele ser lo más común posible en el proceso de revisión de datos introducidos por pantalla.

Por último una vez que se cuenta con los datos de entrada; éstos serán pasados a la siguiente parte del sistema la cual se encargará de realizar todos los cálculos pertinentes y de obtener los resultados más precisos posibles.

4.3 Conceptos generales sobre representaciones en pantalla del lenguaje C.

En el mundo de la computación se han llegado a establecer dos clases básicas de modos para la presentación en pantalla de imágenes las cuales son las de texto y las de gráficos. Para poder trabajar bajo cualquier modo de los anteriores hay que tomar en consideración el tipo de adaptador de video o tarjeta gráfica disponible en la línea de su computadora personal. Las más normales son:

- La monocromática.
- La CGA (Color Graphics Adapter, adaptador de gráficos en color).
- La EGA (Enhanced Graphics Adapter, adaptador de gráficos mejorado).
- La VGA (Video Graphics Array, array de gráficos de video).

Desde luego que hay algunos modos de texto y otros de gráficos para cada uno de estos modos de video los cuales son

soportados según la tarjeta contenida en la computadora en cuestión.

Tanto en los modos de texto como en los de gráficos las posiciones individuales se diferencian mediante su fila y columna, sin dejar de ser propias para cada uno de los casos de los que se trate.

A continuación se describirán más detalladamente cada uno de los modos mencionados con anterioridad. Recordando al lector que el enfocarse en el lenguaje C es de interés principal para la creación del programa que capturará los datos de entrada a la computadora de una manera fácil, rápida, agradable e interactiva, siendo posible la realización de esto bajo el ambiente de Turbo C++ por las facilidades que proporciona este software de programación, pero sin dejar a un lado al lenguaje Fortran ya que encontraremos su aplicación en el manejo óptimo de los cálculos implícitos por el método de elemento finito así como el almacenamiento de los resultados en disco, para consulta del usuario del sistema, razón por la cual fundamenta su uso.

4.3.1 Representación en modo texto.

Un modo texto es muy restrictivo, como lo indica su nombre, está orientado a mostrar o desplegar texto. La mínima unidad en un modo texto es un carácter, el cual se crea sobre

una rejilla de pixeles. En el modo texto la esquina superior izquierda corresponde a la posición 1,1.

Las funciones de texto funcionan por medio de ventanas. La ventana por default es la pantalla completa; por lo que de ahí en adelante se pueden crear en lo sucesivo ventanas especiales para usar las rutinas de texto. Consideremos a una ventana como una puerta que usan los programadores para mandar mensajes al usuario. Puede ser tan grande como la pantalla u ocupar tan sólo varios caracteres. En esta ocasión tendremos la oportunidad de observar varias ventanas activas al mismo tiempo, donde cada una de ellas trabajará de manera independiente en su tarea dentro de la ejecución del programa. El lenguaje C nos permite definir la posición y dimensión de una ventana en particular, de ahí en adelante cualquier función que manipule el texto sólo afectarán a la ventana activa y no al resto de la pantalla.

Las funciones de pantalla en modo texto se pueden dividir en las siguientes categorías:

1.- Entrada y salida básica.

Aquí encontramos todas las funciones de entrada/salida de caracteres que reconocen a la ventana activa tal es el caso de:

Función	Propósito
<code>cprintf()</code>	escribe (salida) formateada en la ventana activa.
<code>cputs()</code>	escribe una cadena en la ventana activa.

putchar() saca un carácter por la ventana activa.
 getch() lee un carácter de la ventana activa.
 gets() lee una cadena de la ventana activa.

Estas funciones no son redireccionables; es decir, no permiten redireccionar la salida hacia o desde un archivo en disco u otro dispositivo auxiliar de almacenamiento, además al no caber un texto de salida en la misma línea de manera completa se pasa automáticamente a la siguiente línea.

2.- Manipulación de la pantalla.

En seguida se presentan las funciones más importantes dentro del contexto de manipulación de pantalla:

Función	Propósito
clrscr()	limpia la ventana activa.
clreol()	borra desde el cursor hasta el final de la línea actual.
delline()	borra la línea sobre la cual está el cursor.
gettext()	copia parte de la pantalla en un buffer de caracteres.
gotoxy()	coloca el cursor en la posición especificada.
insline()	inserta una línea vacía debajo de la posición actual del cursor.
movetext()	copia texto de una parte de la pantalla a otra.
puttext()	copia texto de un buffer a la pantalla activa.
textmode()	establece el modo de texto de la pantalla.
window()	define y activa una ventana.

Como una observación podemos hacer notar que cuando las coordenadas no son válidas para la función `window()`, ésta no hace nada ya que una vez llevada a cabo una llamada a `window()` todas las referencias a coordenadas de posición se hacen relativas a esa ventana y no a la pantalla en sí.

3.- Control de atributos.

Es posible cambiar los modos de video, así como controlar el color del texto, del fondo y poner la pantalla en alta o en baja densidad, mediante la llamada a las siguientes funciones:

Función	Propósito
<code>hihgvideo()</code>	muestra el texto en alta densidad.
<code>lowvideo()</code>	muestra el texto en baja densidad.
<code>normvideo()</code>	muestra el texto en la intensidad normal.
<code>textattr()</code>	establece al mismo tiempo el color del texto y el fondo del mismo.
<code>textbackground()</code>	establece el color de fondo.
<code>textcolor()</code>	establece el color del texto.
<code>textmode()</code>	establece el modo de video.

Es importante darse cuenta de que el cambio del color del texto afecta solamente a las escrituras subsiguientes, ya que no cambia el color del texto que ya estaba en la pantalla con anterioridad.

4.- Estado de la pantalla.

Nos será de gran utilidad conocer el estado de la pantalla para diversos casos.

Función	Propósito
gettextinfo()	devuelve información sobre la ventana de texto actual.
wherex()	devuelve la coordenada x del cursor.
wherey()	devuelve la coordenada y del cursor.

Recordando que las coordenadas son relativas a la ventana activa del texto actual.

4.3.2 Representación en modo gráfico.

En un modo gráfico se pueden especificar exactamente cuales pixeles se quieren encender (en la realidad el término pixel se refiere originalmente al menor elemento de fósforo en el monitor de video que podía excitar el haz de electrones, sin embargo el término se ha generalizado para referirse al menor punto direccionable sobre un monitor gráfico por lo que la unidad elemental en un dibujo no es una celda (posición que ocupa un carácter) sino un punto en la pantalla) logrando así crear gráficas o imágenes detalladas y precisas.

En el modo de gráficos la esquina superior izquierda corresponde a la posición 0,0. Las funciones de gráficos de C operan con ventanas gráficas. Una ventana gráfica es en

realidad una ventana especial que tiene esencialmente las mismas características que una ventana de texto. La ventana gráfica por default es la pantalla completa, sin embargo se pueden crear ventanas gráficas de otro tamaño. Recordando que toda la salida de gráficos se hace relativa a la ventana gráfica actual, la cual no es necesariamente la pantalla entera.

No hay que olvidar que para usar la rutina de gráficos, la computadora debe de estar provista de un adaptador de video gráfico.

Las funciones de gráficos en modo gráfico se pueden dividir en las siguientes categorías:

1.- Control de modo de la tarjeta de video.

Es necesario poner el adaptador de video en algunos de sus modos de video gráficos antes de poder usar cualquiera de las funciones de gráficos.

Función	Propósito
<code>initgraph()</code>	carga en memoria el controlador de gráficos correspondiente al número en específico.
<code>setpalette()</code>	cambia el color de la paleta.
<code>setbkcolor()</code>	establece solo el color del fondo especificado.
<code>closegraph()</code>	sirve para dejar de utilizar un modo gráfico de video y volver a un modo texto.

`restorecrtmode()` establece el adaptador de video al modo en el que estaba antes de la primera llamada a `initgraph()`.

2.- Gráficos básicos.

Las funciones fundamentales de gráficos son las que dibujan un punto, una línea y un círculo como lo son:

Función	Propósito
<code>putpixel()</code>	pone la posición determinada por "x" y "y" del color especificado.
<code>line()</code>	dibuja una línea desde las posiciones especificadas.
<code>circle()</code>	dibuja un círculo de radio a partir de la posición especificada.
<code>setcolor()</code>	establece el color actual del dibujo.
<code>floodfill()</code>	rellena cualquier figura cerrada.
<code>setfillpattern()</code>	define un patrón de relleno.
<code>box()</code>	dibuja un cuadro de un color determinado dadas las posiciones específicas.

3.- Salida del texto.

Para mostrar texto en modo gráfico se puede hacer uso de las siguientes funciones, no sin antes recordar que en los modos gráficos no existe un cursor visible, pero la posición actual se reconoce como si existiera uno visible.

Función	Propósito
<code>outtext()</code>	escribe la cadena en la posición actual.

`settextstyle()` cambia el estilo tamaño o dirección del texto.

`outtextxy()` escribe el texto en una posición determinada de la ventana gráfica activa.

4.- Estado de la pantalla.

Se pueden usar varias funciones para obtener información sobre la pantalla de gráficos.

Función	Propósito
<code>getviewsettings()</code>	carga en una estructura información diversa sobre la ventana gráfica actual.
<code>viewporttype()</code>	controla la salida que sobre pasa los límites de la venta gráfica actual.
<code>gettextsettings()</code>	obtiene información sobre el tipo actual de texto gráfico.
<code>getpixel()</code>	devuelve el color del punto (pixel) especificado.
<code>textsettingstype()</code>	informa sobre la fuente, dirección, tamaño de los caracteres y la justificación de un texto específico.

5.- Manipulación de la pantalla.

Tenemos las siguientes funciones:

Función	Propósito
<code>clearviewport()</code>	limpia la ventana gráfica activa.
<code>getimage()</code>	copia parte de la ventana gráfica en

un buffer.

`imagesize()` devuelve el número de bytes que se necesitan para salvar una imagen.

`puttimage()` copia el contenido de un buffer a la ventana gráfica actual.

`setactivepage()` determina que página se verá afectada por las rutinas de gráficos.

`setviewport()` crea una ventana gráfica.

`setvisualpage()` determina la página que se muestra.

5.- PROGRAMACION DEL SISTEMA POR COMPUTADORA.

En este capítulo se hará mención a todo lo concerniente a la programación del sistema en sí, el título de este capítulo "Programación del Sistema por Computadora" engloba tres aspectos fundamentales a señalar:

1.- Programación: involucra la acción de crear un programa de computadora para cumplir con un fin específico. Durante este proceso de creación del programa se pueden señalar dentro de un contexto no muy riguroso tres etapas a cubrir, las cuales son:

- Etapa 1.- Diseño de la lógica del programa.
- Etapa 2.- Escritura del programa en computadora.
- Etapa 3.- Prueba del programa en computadora.

Dependiendo del éxito obtenido en los resultados se puede volver a hacer uso de cada una de las etapas anteriores con la finalidad de obtener los resultados deseados.

2.- Computadora: representa un término técnico que se le ha dado a las máquinas u ordenadores de propósito general que procesan datos de acuerdo con el conjunto de instrucciones que están almacenados internamente ya sea de manera temporal o permanente. De manera concreta a la computadora y a todo el equipo conectado a ella se le ha denominado HARDWARE; mientras que a las instrucciones que le indican lo que tiene que hacer se le llama SOFTWARE. Indudablemente un conjunto de instrucciones que llevan a cabo una tarea específica se le denomina programa ó programa de software, toda acción de programar (en sus etapas 2 y 3 específicamente) hacen

hincapié en el uso de una computadora para llevar a cabo dichas tareas.

3.- Sistema: es un término genérico que se utiliza para hacer referencia a un grupo de componentes relacionados entre si que interactúan para realizar una tarea en particular. A este respecto consideramos como un software de sistemas a la categoría de programas que se emplean para controlar a la computadora y que nos permiten ejecutar a otros programas de aplicación más específica aún. En nuestro caso específico el grupo de componentes del que se habló anteriormente estarán representados por las siguientes partes:

- A.- Problema a resolver logrando especificar la información que se nos proporciona para tal caso.
- B.- Programa de aplicación para resolver el problema anterior:
 - B.1.- Preprocesador.
 - B.2.- Procesador.

Los anteriores puntos conforman a un todo que es el sistema y bajo el cual cada una de estas partes están altamente relacionadas y que por demás son responsables del funcionamiento del mismo.

5.1 Análisis global de sistema.

Retomando todo lo anterior, se pretende desarrollar un sistema programado en computadora cuyo objetivo sea respaldar mediante recursos técnicos (como lo es la computadora) el análisis de medios continuos mediante la modelación de estados planos siguiendo el procedimiento del método del

elemento finito. El hablar del método del elemento finito implica considerar tanto la magnitud de los cálculos que le son propios, así como el manejo de cantidades de datos que en algunos casos llegan a ser extensos y que por ende resultan ser muy difíciles de manipular manualmente. Esto explica el motivo de la elaboración de un sistema que computacionalmente se encuentre capacitado tanto para el manejo de grandes cantidades de datos como para la manipulación (cálculos implícitos por el MELEFIN "Método del Elemento Finito") propia de los mismos; ambas partes llevadas a cabo de una manera eficiente por parte del sistema. Este sistema estará apoyado por el software de sistemas (DOS, Turbo C++, Fortran) propio de la computadora a utilizar y por las capacidades con que cuenta la misma para llevar a cabo el desarrollo de nuestro sistema en particular.

Sin embargo al desarrollar los programas de aplicación es necesario tomar a consideración la inexistencia de un criterio de programación el cual resulte totalmente óptimo para llevar a cabo la programación de los mismos, por ello se ha tenido la necesidad de recurrir a las herramientas básicas con que cuenta la programación, para llevar a cabo de la manera más fácil y práctica la elaboración de nuestros programas. Tal es el caso del empleo de la programación estructurada ya que mediante el empleo de ésta técnica ha sido posible escribir programas moderadamente complejos de una forma bastante sencilla, aunque al elaborar programas con gran tamaño su complicidad aumenta para el programador, este

se ve en la necesidad de recurrir al manejo de proyectos, o sea, en el entorno integrado de Turbo C++ a los programas de archivos múltiples se les llaman proyectos, ya que sólo hay que indicarle al compilador de Turbo C++ que el programa está compuesto por varios archivos para que la compilación automática de este código y de los demás archivos que componen a este programa puedan generar el archivo ejecutable de todo el programa. La importancia de utilizar esta técnica de programación es de no agotar los 384k de memoria RAM de que dispone el compilador al compilar un solo archivo (demasiado extenso) y tener que ajustarnos a esta regla sin poder compilar más código; ya que el compilador asigna esos 384k de memoria para cada uno de los archivos subiendo cada uno de éstos a la memoria principal para su compilación pertinente.

Desde un punto de vista específico para llevar a cabo la técnica de la programación modular (también considerada en el caso de nuestra programación) implica describir lo más detalladamente el problema o el objetivo del programa a desarrollar, dicha descripción será descompuesta en varios subproblemas más o menos independientes entre sí de tal forma que cada detalle pueda ser llevado a cabo a la programación de una manera única ó trivial haciendo referencia a su sola descripción. Lo que respecta a la contribución de esta técnica en el ámbito de la programación es muy importante y bastante extensa ya que nos permite dividir el problema en varios subproblemas y a su vez éstos programarlos en varios

módulos con la finalidad de que en primera, cada uno de ellos resuelvan el subproblema para el que fue creado y en segunda, cada parte pueda ser probada separadamente para una más fácil depuración de errores en el momento en que se lleguen a necesitar y por último podemos hacer uso del manejo de variables locales utilizadas para cada módulo como almacenamiento de información temporal así como de variables globales para facilitar su empleo a lo largo de la programación en cuestión.

Una vez conocidas las herramientas básicas de la programación de que se hará uso para el desarrollo de los programas de aplicación que conforman al sistema a desarrollar, nos permiten abordar el siguiente punto.

5.1.1 Partes que componen al sistema.

El sistema básicamente está conformado por tres partes, dos de ellas son programas de aplicación que como anteriormente se habló fueron desarrolladas en la computadora. Describiendo cada una de estas partes encontramos lo siguiente:

PROGRAMAS DE APLICACION

Es preciso mencionar que todo programa de aplicación requiere de entradas y salidas, las entradas regularmente son datos que al ser manipulados por estos programas producen salidas información que al usuario del programa le ha de

resultar de gran relevancia. Uno de estos programas está caracterizado por asimilar a un preprocesador (llamado PREPRO), término usualmente empleado para describir al editor de texto tal y como el que tiene Turbo C++. Nuestro preprocesador a diferencia de los demás sólo se encargará de almacenar y disponer en disco los datos con los cuales trabajará el procesador, estos datos no sufrirán ningún cambio desde su entrada hasta su salida del preprocesador. El segundo programa de aplicación lo es uno llamado procesador (PROCESA) dicho programa se encargará de recibir los datos introducidos por el usuario en su previo uso del preprocesador, para que mediante estos datos lleve a cabo todos los cálculos matemáticos pertinentes para los que fue creado y que por su medio pueda mostrar los resultados a los que llegó.

PROBLEMA Y SU ANALISIS

Otra de las partes que conforman al sistema lo es el problema a resolver y su análisis, esta parte es la que proporciona los datos que el usuario tendrá que dar al preprocesador como sus entradas, es una parte muy importante ya que el empleo del sistema mediante el uso de los programas de aplicación ha de tener una razón de ser, que por demás fundamentada, ha de responder a la necesidad de resolver un problema en cuestión y al que el sistema pueda responder plenamente mediante su uso.

Pues bien a groso modo, las partes que componen al sistema se han dejado ver y son ellas mismas las que nos conducen a la explicación del siguiente punto.

5.1.2 Organización del sistema.

Para explicar la organización del sistema recurriremos a la descripción del sistema en base al objetivo de cada una de las partes que lo componen.

En lo que respecta a los datos iniciales encontramos que corren a cargo del problema a resolver ya que es él quien proporcionará los datos reales para que los resultados puedan tener un buen fundamento; el preprocesador también se encuentra ubicado junto con los datos iniciales ya que él es quien indica cómo, en donde y cuales son los datos que deben de introducirse para su empleo posterior. Todos y cada uno de los datos pedidos por el preprocesador son indispensables ya que el hacer caso omiso de alguno de ellos podrían conducirnos a la obtención de resultados inesperados.

El objetivo del preprocesador es pedir los datos indispensables para el funcionamiento correcto del procesador, además de pedirlos, los almacena en un orden específico para que el procesador pueda hacer uso de ellos adecuadamente. El preprocesador cumple con una finalidad aún más, ya que facilita la introducción de los datos, así como su modificación o corrección de los que en su momento se

considere oportuno, estableciendo una interactividad muy práctica entre el usuario y el manejo de este programa de aplicación.

Por último la parte de los cálculos corren a cargo del procesador ya que es él quien es el responsable de manipular los datos iniciales para poder aplicar el método del elemento finito y producir los resultados que al usuario han de interesar.

La organización del sistema remarca la importancia de cada una de sus partes las cuales lo hacen aún más relevante ya que omitiendo algunas de ellas podría no funcionar adecuadamente.

5.1.3 Secuencia del proceso de operación del sistema.

La secuencia de operación del sistema es por demás sencilla ya que las partes que lo componen hacen referencia a su uso por si solas, pero para facilitar aún más el entendimiento de esto presentamos la fig. 5.1.

La fase I está representada por el problema a resolver, suponemos que el uso del sistema responde a la necesidad de resolver un problema concerniente al análisis estático por el método del elemento finito y al que el sistema pueda contribuir grandemente en cuanto a su resolución.

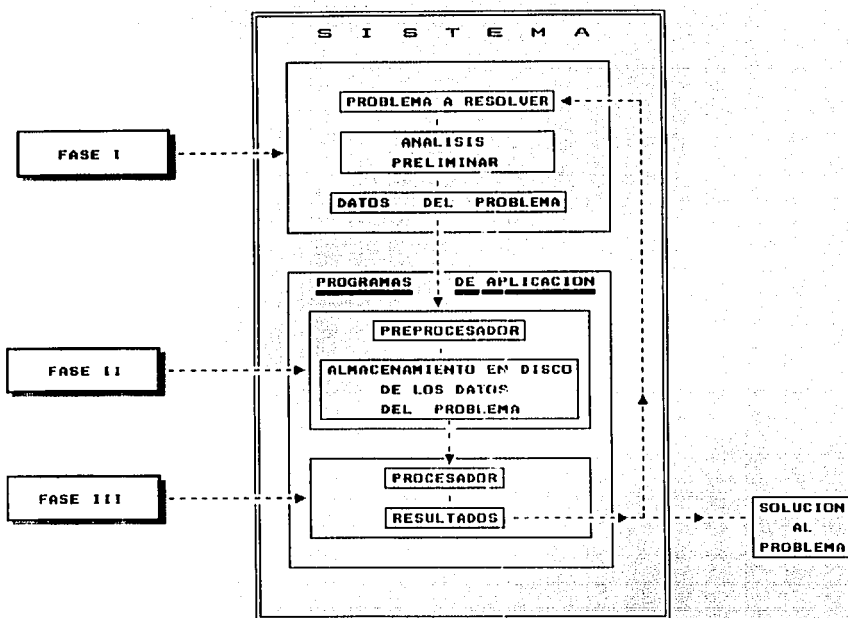


Figura 5.1 Esquema que muestra las partes y la secuencia de operación del sistema.

Una vez determinado el problema en cuestión es trabajo de primer instancia delimitar los datos con que cuenta el problema a resolver además de estar seguros que con la aplicación del sistema es como se puede llegar a la solución del problema en cuestión. La parte del análisis, implícita en la fase I, es importante ya que es mediante éste análisis la forma de determinar los datos de problema que le hacen referencia. A lo que se refiere esto es a lo siguiente; partiendo del problema a resolver, dado que hay que aplicar el método del elemento finito para lograr un resultado satisfactorio, hay que hacer un análisis del problema de tal

forma que la estructura a analizar se llegue a discretizar en una serie de elementos cuadráticos (8 puntos) con la finalidad de obtener los datos referentes a: -puntos nodales, -elementos, -fronteras, -materiales, -nodos cargados, -cargas y demás datos que requieren cada uno de los puntos anteriores para que finalmente contemos con los datos necesario para aplicar el MELEFIN a nuestro problema a resolver. Con todo lo anterior logramos pasar a la fase II , en la fase II como todo programa de aplicación el preprocesador tendrá la necesidad de recibir los datos del problema; dichos datos serán pedidos e introducidos conforme se haga uso del mismo hasta completar con todos los datos que se requieren para que la fase III pueda operar adecuadamente. En la medida que el preprocesador pida los datos, el usuario tendrá que introducir todos y cada uno de ellos ya que la omisión de algunos de ellos podrían producir resultados inesperados. El preprocesador terminará su uso en el momento en que genere con los datos correctamente introducidos un archivo en disco, el cual pueda ser usado por el procesador en la fase III. Por lo que respecta a la fase III los datos almacenados en disco serán utilizados por el procesador como su entrada para poder llevar a cabo la manipulación de los mismos y obtener como resultado de la ejecución del programa un archivo con los resultados de los cálculos realizados. Un último análisis por parte del usuario a estos resultados lo conducirán a tomar la decisión pertinente en cuanto a una respuesta satisfactoria

como a un nuevo intento en el empleo del sistema en una ocasión posterior.

5.2 Preprocesador.

Desde un punto de vista técnico, el término "preprocesador" representa el nombre asignado a cierto software el cual efectúa algún procesamiento preliminar de las entradas de datos, antes de que éstas sean manipuladas por algún otro programa principal; esto aplicado a nuestro caso simboliza a una de las partes del sistema mediante la cual podemos alimentar a la computadora con los datos referentes de la estructura a analizar.

5.2.1 Generalidades.

El preprocesador (llamado en esta ocasión PREPRO) fue programado en el lenguaje C, facilitando la interactividad usuario-computadora a lo largo de toda la ejecución del programa, con la finalidad de hacer lo más práctico posible las siguientes funciones: almacenamiento de datos de la estructura en la memoria de la computadora y después a disco o viceversa y disponer de los mismos datos de tal forma que

puedan ser consultados (mediante el acceso de la computadora) por el programa subsecuente.

También se ha considerado que el programa se encuentre capacitado para realizar todas las actividades que el manejo de datos ha de requerir, logrando realizarlas un gran número de veces en problemas subsecuentes. Además de contar con la posibilidades de poder manipular los datos ya almacenados sin tener que introducirlos por segunda vez.

Básicamente el preprocesador fue creado bajo la idea de una plena interactividad usuari-computadora, dado que su diseño ha seguido las normas de sólo almacenamiento de datos había que facilitar cualquier tarea que de ellos se hiciese necesario, tal fuera el caso en el que para una situación especial el usuario tuviese que molestarse en introducir sus datos del problema una sola vez y que por ende, a partir de ahí lograra tenerlos almacenados en la computadora para cualquier otro momento; en este aspecto se consideró la posibilidad del almacenamiento de los datos de la memoria de la computadora a un archivo específico en disco con la finalidad de revertir el proceso en el momento que se considerara preciso, con la plena seguridad de contar con todos los datos como si se hubiesen introducido por primera vez. La posibilidad de una posterior verificación de los datos introducidos también ha sido considerada ya que se cuenta con una opción de impresión de los datos en hojas o una consulta del archivo generado en disco, ambas opciones

presentan los datos tal y como fueron introducidos por el usuario.

5.2.2 Organización.

Como se mencionó anteriormente el programa fue realizado en el lenguaje de programación C, por tal razón fue fácil el manejo de los proyectos (programa formado por archivos múltiples) y el manejo de funciones tanto predefinidas por C como las definidas por el usuario ya que cuenta con un compilador propio el cual se encarga de la creación del archivo ejecutable de nuestro programa a compilar de una manera más o menos rápida. El preprocesador cuenta con 6 archivos uno de ellos el principal y de ahí son llamados los demás, además de tener aproximadamente 64 funciones definidas por el propio programador.

El proyecto creado se llama PREPRO.PRJ mientras que la lista de archivos que contiene este proyecto son los siguientes:

- ANEELEFIN.C
- MOUSE.C
- PRESENTA.C
- ARCHIVOS.C
- DATOS.C
- IMPRIMIR.C

El archivo ANEELEFIN hace referencia al programa principal en él encontramos funciones para controlar los menús de opciones tanto horizontales -MENUHORI- como verticales -MENUVERT- esto trae consigo la necesidad de

verificar la opción elegida por el usuario con las funciones -FUN_VALID- para determinar si es una tecla válida ya sea un "enter", alguna de las flechas direccionales o alguna otra tecla del teclado, así mismo -CHE_CAR- y -CHE_CAR2- ambas funciones son para el menú horizontal como para el vertical respectivamente ya que verifican si la tecla presionada hace alusión a una de las primeras letras de las opciones a elegir.

Por otro lado el archivo MOUSE engloba funciones del tipo que esperan la presión de una tecla cualquiera tal es el caso que para el primer menú se utilicen las siguientes funciones: -PRES_1_TECLA- la cual acompañada de -MOUSE1- y de -VALID_1_CLICK-, son las que realizan el trabajo de la espera de la presión de una tecla o del click del ratón, a su vez la función -VALID_1_CLICK- es la responsable de validar si la posición en que se efectuó el click es válida para alguna de las opciones a seleccionar o no. Estas funciones van acompañadas por los diferentes menús que presenta el programa a lo largo de su ejecución por lo que las funciones anteriores son referentes al nivel 1 y para el nivel 2 encontraremos otras tantas similares como lo son: -PRES_2_TECLA- , -MOUSE2- , -VALID_2_CLICK- las funciones de este nivel son para las alternativas previas a la introducción de los datos, por otro lado las encargadas de recibir cada dato perteneciente a una opción seleccionada son: -PRES_3_TECLA- , -MOUSE3- y -VALID_3_CLICK-.

En el archivo PRESENTA se encuentra la función -CURSOR- la cual detecta la presencia del cursor para nuestro programa, la función -FONDO- dibuja en la pantalla un marco el cual servirá de fondo y estará presente a lo largo de toda la ejecución del programa, -PRESENTA- muestra las pantallas referentes a los datos del programador, asesor, escuela de procedencia así como el nombre del programa, -MARCO- dibuja en cada una de las ventanas que se abren un marco para delimitar el espacio de escritura en ellas, -SOMBRA- y -SOMBRA_2- pretenden dibujar las sombras de las ventanas abiertas, y por último -ABRE_VENTANA- es la que abre y activa cada ventana en el lugar y del tamaño que se necesita para dar un imagen al usuario del uso de ventanas colgantes o multiventanas. Ahora bien en el archivo con nombre ARCHIVO están incluidas todas la funciones que hacen referencia a las opciones que contempla la opción "archivo" del programa, tal es el caso de la función -NUEVO-, su objetivo es borrar de la memoria de la computadora todos los datos cargados hasta ese momento y dejarla lista como si se entrara al programa nuevamente, -ABRIR_ARCHIVO- se refiere a la tarea de abrir y cargar en memoria un archivo con los datos guardados en el disco de trabajo, para realizar esto utilizamos la función -CURRENT_DIRECTORI- la cual proporciona al usuario una opción de ruta para buscar los archivos a abrir, -VENTANA_ARCHIVOS- abre una ventana y muestra al usuario los archivos con extensión .dat para que él pueda elegir con libertad el que desee; dado que la cantidad de

archivos a mostrar puede variar se contempló la necesidad de manejar dos funciones más -WRITE_ARCHIVOS- la cual escribe en pantalla los archivos almacenados en el disco con extensión .dat y -COLUMNA_ARCHIVOS- la cual controla la cantidad de los archivos mostrándolos en 2 columnas de 10 en 10. Por último una vez determinado el archivo a abrir se procede a dar paso a la función -TRAER_DATOS_ARREGLO- la cual leerá la información contenida dentro del archivo en cuestión y la almacenará en memoria para que el usuario la pueda consultar. La función -GUARDAR- es similar a la de -ABRIR_ARCHIVO- sólo que el proceso es invertido, ya que con el uso de una función que muestra una opción de ruta para guardar los datos cargados en memoria -CURRENT_DIRECTORI- y con un nombre específico con extensión .dat se pueda aplicar la función que los grabará en el disco -SALVAR_DATOS_ARREGLO- bajo un nombre específico. Dado que también se crean dos archivos adicionales con la finalidad de que uno de ellos pueda servir de consulta de datos introducidos por el usuario (archivo con extensión .cns) y el otro tenga almacenados los datos de cierta forma que el procesador pueda leerlos (archivo con extensión .fmt) se necesitó de las siguientes funciones: para el primer caso, la función -ARCHI_EXTRA- englobó las funciones siguientes:

-ARRE_ARCHI_EXTRA- , -LIN_ARCHI_EXTRA- y -*FORMATO- cuyo objetivo es grabar los datos en orden descendente de tal forma que el procesador los pueda leer e interpretar correctamente. Para el segundo caso la función

-CONSULTA_ARCHI- permitió grabar los datos en un archivo en disco tal y como fueron introducidos en la captura inicial, tal medida corrió a cargo de las funciones -ARCHIVO_ESCRITURA-, -LINEA_ESCRITURA- y -*FORMATO- para permitir que el usuario pudiera consultar los datos sin la necesidad de tener una impresora para imprimirlos.

La función -DIR- realiza un proceso similar a la función -ABRIR_ARCHIVO- ya que nos muestra mediante las funciones -CURRENT_DIRECTORI-, -VENTANA_ARCHIVOS-, -WRITE_ARCHIVOS-, -COLUMNNA_ARCHIVOS-, los archivos con cualquier nombre y extensión los cuales se encuentran en la ruta especificada, con la finalidad de conocer los archivos que se encuentran en una ruta sin la necesidad de salir del programa en ese preciso momento.

Por otro lado la función -BORRAR- mediante una ruta específica -CURRENT_DIRECTORI- borra del disco el archivo que se especificó anteriormente.

En lo que respecta al archivo DATOS encontramos varias funciones indispensables para el manejo de la entrada de los datos del problema ya que aunque la ventana que se abre para escribir en ella los datos es uniforme los datos que son escritos en ella no lo son, ya que su tamaño y cantidad pueden variar, para ello utilizamos la función -DETERMINA_TAMAÑO_SIZE- que junto con la función -TAMAÑO_VENTANA- determinan la cantidad de los datos a escribir, así se escriben los formatos de cada uno de los datos con ayuda de las funciones siguientes:

-ESCRIBE_LINEA_CEROS-, -*FORMATO-, -POSICION-, -RIGHT-,
-TAMANO-; sin lugar a dudas también hay que considerar que
con el movimiento de donde nos sitúa el programa por default
hasta la posición que deseamos llegar, para una consulta o
para una modificación de algún dato almacenado en ella se
tiene que hacer uso de las flechas direccionales o del ratón,
por ende debemos de considerar otras funciones que permitan
dar la impresión al usuario de los movimientos que se han
hecho, todo esto se lleva a cargo mediante las funciones:
-ESCRIBE_LINEA_CON_SOMBRA-, -*FORMATO-, -POSICION-, -RIGHT-,
-TAMANO-.

Por último mencionaremos el archivo IMPRIMIR, el cual
como su nombre lo indica se encuentran las funciones
referentes a la impresión de los datos cargados en memoria
hacia la impresora tales funciones son: -IMPRIMIR-,
-ARREGLO_IMPRESO-, -LINEA_IMPRESA- y -*FORMATO-.

Es importante mencionar que los datos que se imprimen
son escritos tal y como fueron capturados por el usuario, por
tal motivo puede ser un buen medio para la verificación de
errores, para guardarlos en hojas impresas para una consulta
individual o simplemente representara un buen respaldo de los
mismos.

NOTA: En el archivo libreria.h se encuentran
almacenadas todas las variables globales para que puedan ser
empleadas a lo largo de toda la ejecución del programa sin
tener que definir las en cada archivo que conforman al
proyecto.

5.2.3 Manual de usuario.

El programa de aplicación preprocesador por haber sido elaborado en Turbo C++ y compilado por él mismo, ha generado un archivo PREPRO.EXE cuyo programa puede ser ejecutado desde el sistema operativo DOS. Sólo es necesario escribir A:>PREPRO, como consecuencia a esto aparecerá en la pantalla la fig. 5.2, haciendo alusión a los datos del programa que se está ejecutando en ese momento, posteriormente aparecerá la fig. 5.3, la cual especifica el nombre del programa y lo que representa el hacer uso de él.

Dado que se pide al usuario que espere un momento a que termine la presentación del programa, no es necesario presionar ninguna tecla, posteriormente el programa crea el adorno que le servirá de fondo y en la parte de arriba dibuja las opciones del primer nivel fig. 5.4.

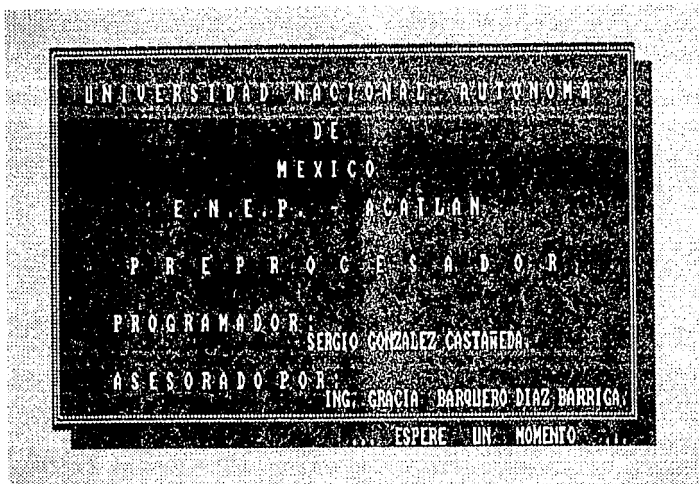


Figura 5.2 Pantalla de presentación del preprocesador.



Figura 5.3 Pantalla que muestra el significado a las siglas que conforman al nombre del preprocesador.

FALLA DE ORIGEN

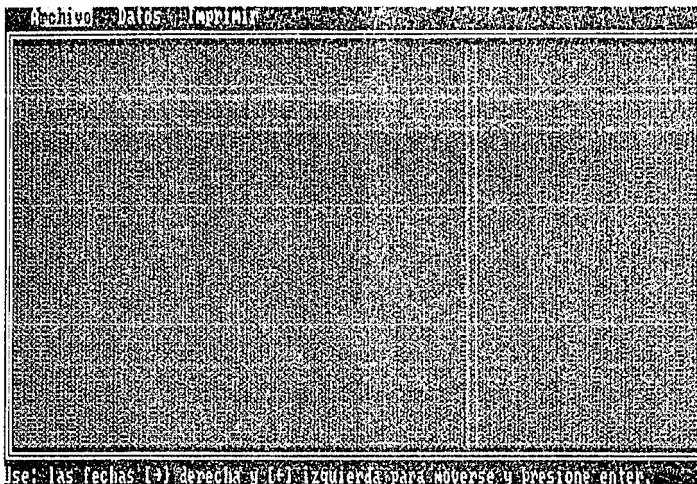


Figura 5.4 Menú que muestra las opciones principales.

Este menú se encontrará siempre presente en la pantalla, ya que es una manera de mostrar al usuario la parte del programa que se está ejecutando. La manera de manejar al programa es por medio de opciones ejecutadas con la ayuda de menús desplegable con estructura arborescente ya que un menú arborescente posee varios niveles de los cuales cada uno constituye una subdivisión del nivel superior. El nivel principal o superior consta de 3 opciones las cuales son: Archivo, Datos, Imprimir como se ha visto en la figura anterior. Dichas opciones pueden ser ejecutadas con el manejo del cursor a la opción deseada y pulsando "enter" ó mediante la colocación del cursor del ratón sobre la opción deseada y con la sola presión del botón izquierdo del ratón. Una última opción es mediante el uso de la presión de la letra resaltada de la opción deseada ya sea minúscula o mayúscula ya que

FALLA DE ORIGEN

oprimiéndola se ejecuta automáticamente la opción esto asemeja la funcionalidad del ratón.

Comenzando por la opción ARCHIVO encontramos que cuenta con un submenú, como lo muestra la fig. 5.5, el hablar de archivos implica referirse específicamente al manejo de archivos; es decir, al lugar donde van a ser almacenados los datos, de que disco van a ser leídos, guardados, borrar datos tanto en memoria como el contenido de archivos en disco, etc. hasta encontrar la salida del sistema. Como podemos observar las opciones son las siguientes ver fig. 5.5.

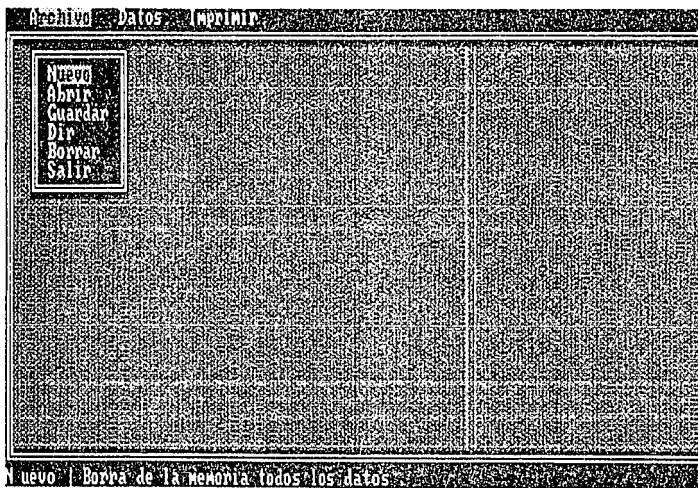


Figura 5.5 Menú de opciones referentes a la opción ARCHIVO.

NUEVO:

Esta opción borra los datos que se encuentren cargados en la memoria de la computadora hasta ese momento.

FALLA DE ORIGEN

Nota: cualquier dato que no haya sido previamente almacenado en disco será removido de la memoria principal corriendo el riesgo de perderse fig. 5.5.

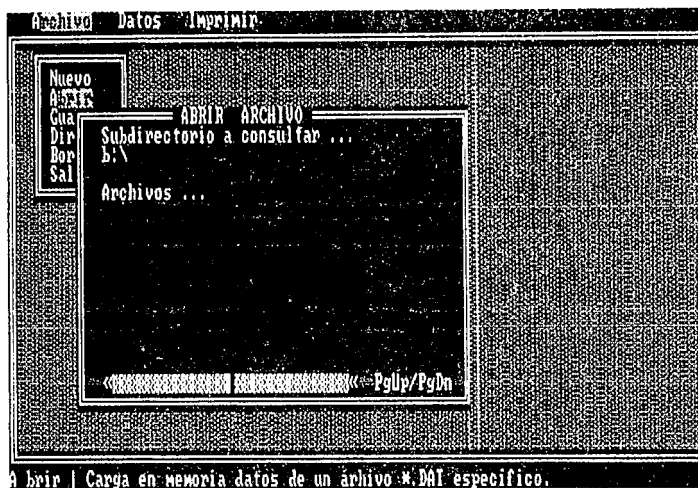


Figura 5.6 Captura de la unidad y directorio de trabajo.

ABRIR:

Se refiere a cargar en memoria datos referentes a algún archivo almacenado previamente en disco, esta opción se divide en dos partes; la primera de ellas se refiere a que el usuario introduzca la ruta o la dirección donde se encuentra el archivo a abrir fig. 5.6, el programa siempre por default determina la ruta en base donde se ha cargado el programa, si el usuario no considera ser la correcta entonces podrá modificarla y presionar "enter". A continuación el programa muestra al usuario los archivos con extensión .DAT que se encuentran almacenados bajo la ruta especificada en el disco especificado fig. 5.7.

FALLA DE ORIGEN

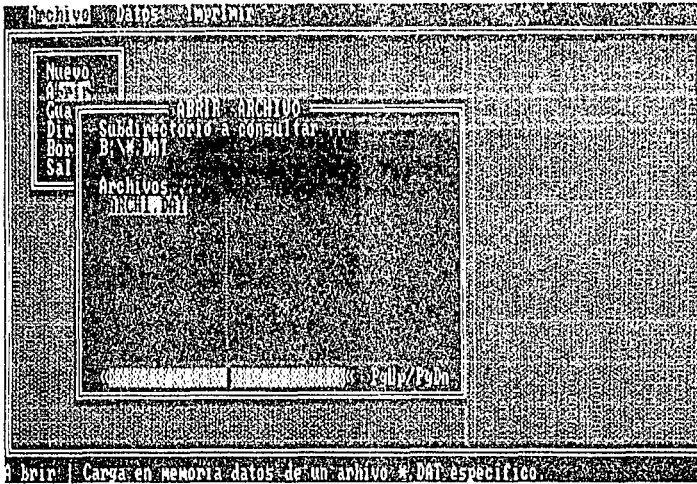


Figura 5.7 Presentación de los archivos contenidos dentro del directorio a consultar.

Se pide que seleccione cualquiera de los archivos mostrados con la finalidad de cargar los datos almacenados en ellos en la memoria, la selección puede correr a cargo del ratón colocando el cursor del ratón sobre el nombre del archivo y pulsando el botón izquierdo o moviéndose a lo largo de la ventana hasta llegar al archivo deseado y presionando "enter", los archivos pueden no caber en toda la ventana por lo que también pueden hacer uso de la teclas Pgdn/Pgup o de manera similar dar un clic a lo largo de la barra de Pgdn/Pgup, uso exclusivo del ratón, para mostrar el resto de los archivos que no caben y poder aplicar el criterio de selección. Al salirnos de esta opción el usuario podrá consultar la opción de datos para verificar si se han cargado los datos del archivo especificado del disco a la memoria.

FALLA DE ORIGEN

GUARDAR:

Aquí se pide la ruta del archivo que se generará para almacenar los datos cargados en memoria fig. 5.8, después de introducir la ruta o dirección donde será almacenado el archivo se pedirá el nombre *.DAT con que serán guardados los datos fig. 5.9.

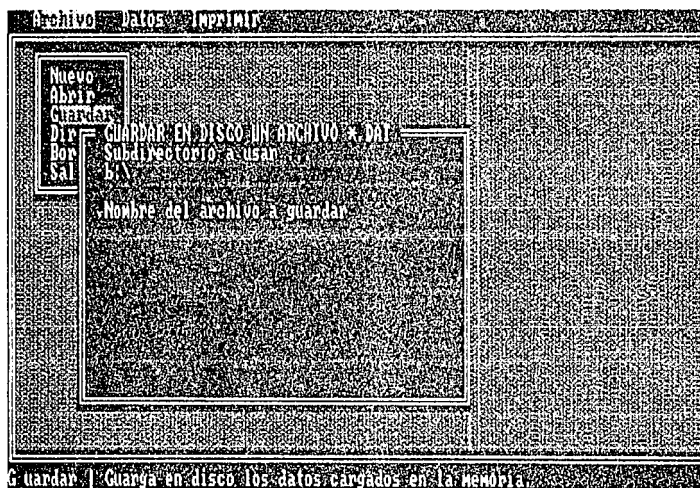


Figura 5.8 Modificación de la unidad y directorio de trabajo.

Es importante mencionar que cuando el nombre del archivo *.DAT introducido exista, el programa preguntará si desea reescribir la información almacenada en ese archivo ó no, en un caso afirmativo el programa borrará la información del archivo anteriormente encontrado en el disco y guardará en él la información proveniente de la memoria realizando un reemplazo de la información del disco por la de la memoria de la computadora.

FALLA DE ORIGEN

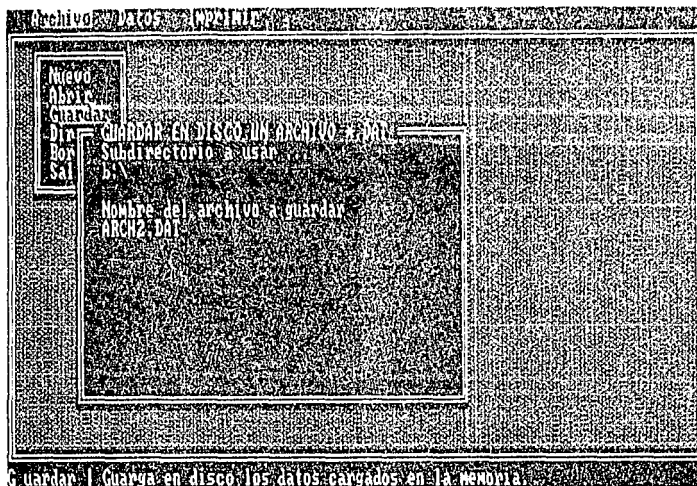


Figura 5.9 Especificación del nombre del archivo para guardar la información.

Finalmente la opción guardar finaliza creando dos archivos adicionales: el primero de ellos es guardado bajo el mismo nombre de archivo que el usuario escribió pero con extensión .FMT, su utilidad se basa en el empleo de este archivo para el almacenamiento de los datos de la memoria pero bajo ciertas reglas todo de acuerdo al funcionamiento del procesador para el que fue creado, y el segundo de ellos también conserva el mismo nombre que los anteriores archivos sólo que con extensión .CNS, la finalidad de este archivo es poder almacenar los datos tal y como fueron introducidos por el usuario para poder consultarlos en pantalla en el momento que se necesite fig. 5.10.

FALLA DE ORIGEN

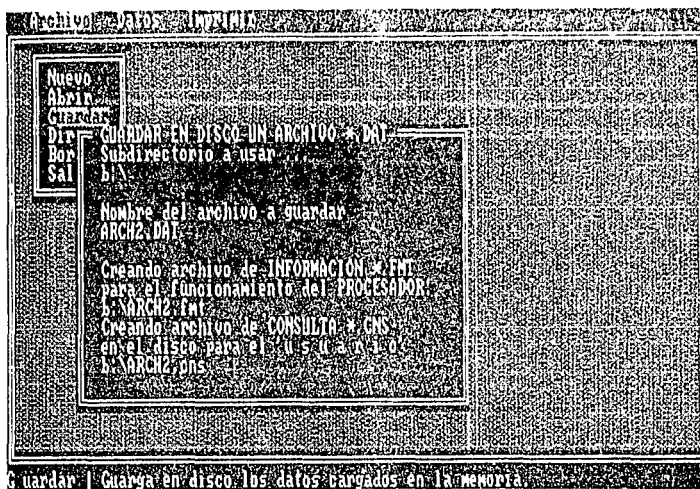


Figura 5.10 Presentación de la variedad de archivos creados por el preprocesador.

DIR:

Aquí se muestra mediante una ruta específica el contenido de archivos *.* almacenados en disco fig. 5.11.

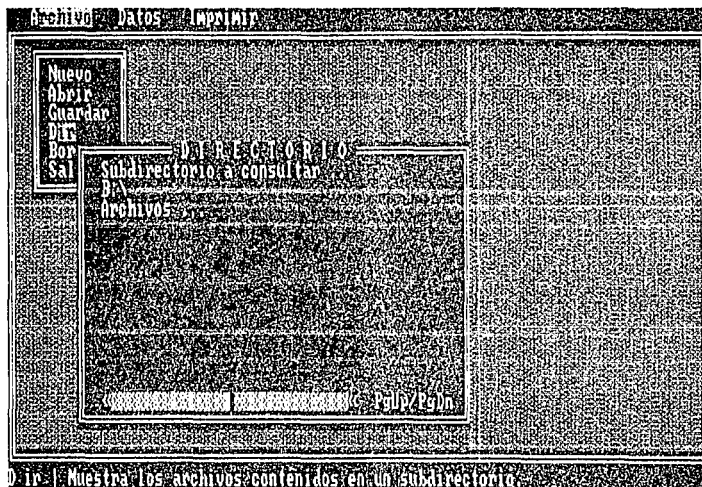


Figura 5.11 Captura de la unidad y directorio de trabajo.

FALLA DE ORIGEN

De manera similar a la opción abrir, se mostrarán los archivos contenidos en la ruta deseada, para mostrar más archivos que no aparezcan en la ventana podemos hacer uso de las teclas Pgdn/Pgup o de la barra dibujada casi al final de la ventana para el uso del ratón.

Para salir de esta opción debemos de presionar "enter" en cualquier nombre de archivo o mover el cursor del ratón hacia cualquier nombre de archivo y dar un click con el botón izquierdo o simplemente cancelando esta operación solamente presionando la tecla "escape" fig. 5.12.

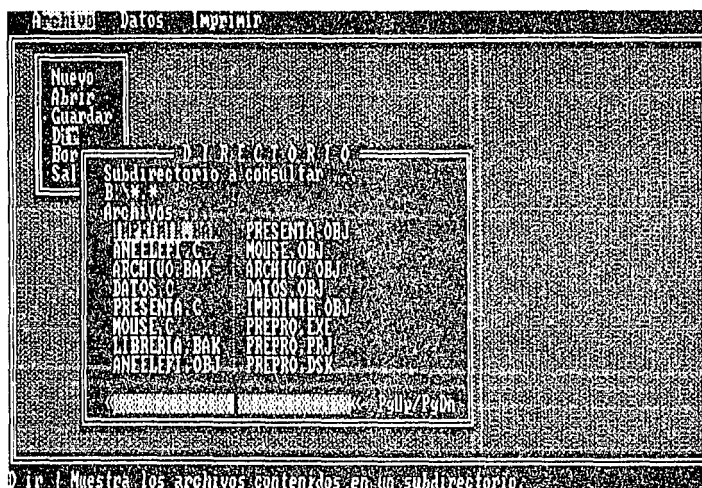


Figura 5.12 Presentación de archivos del directorio actual.

BORRAR:

En el caso en que deseemos borrar un archivo del disco ya sea porque se encuentra lleno o porque en definitiva ya no nos sirva entonces procedemos a hacer uso de esta opción,

primeramente nos preguntara la ruta del archivo a borrar
fig. 5.13.

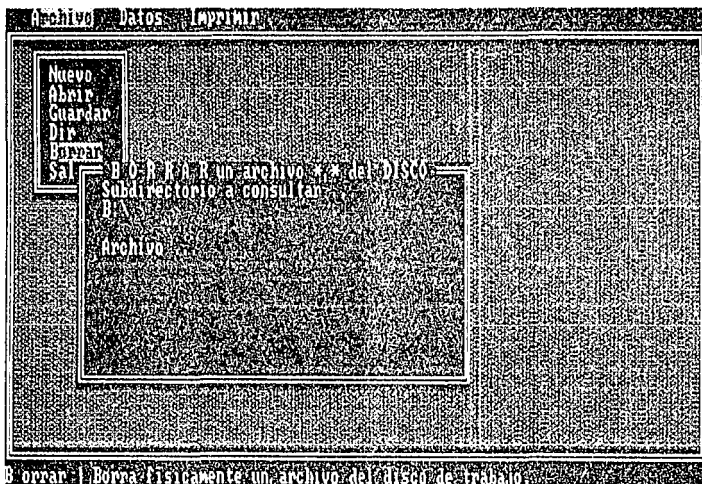


Figura 5.13 Captura de la unidad y directorio de trabajo.

En segunda nos pedirá el nombre del archivo a borrar
*. * basado en el direccionamiento especificado fig. 5.14.

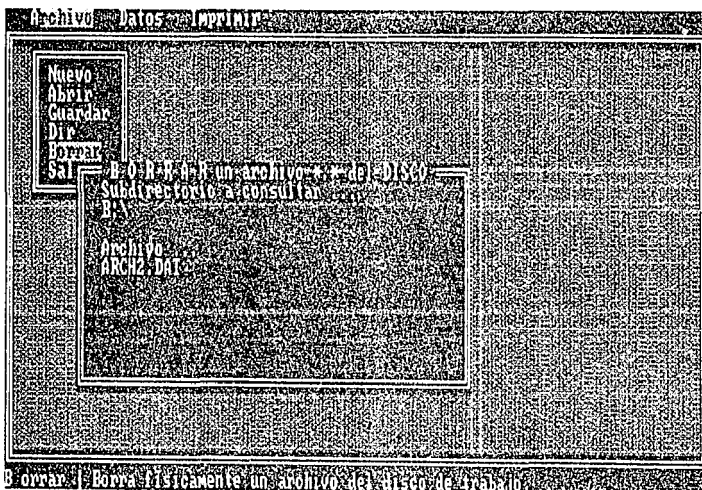


Figura 5.14 Especificación del nombre del archivo a borrar.

FALLA DE ORIGEN

En caso de que el archivo no existiera el programa informará de esto inmediatamente al usuario, en caso contrario volverá a preguntar si se desea borrar para que el usuario confirme la decisión fig. 5.15.

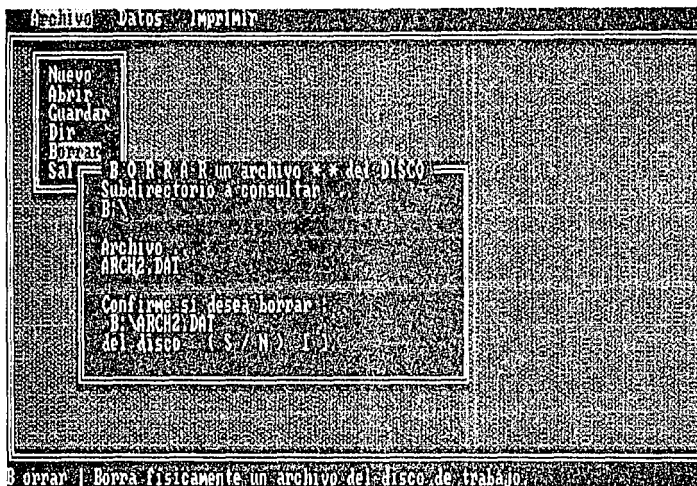


Figura 5.15 Confirmación del archivo a borrar.

SALIR:

Esta opción termina el proceso de ejecución del PREPRO.EXE y nos conduce al sistema operativo DOS fig. 5.16.

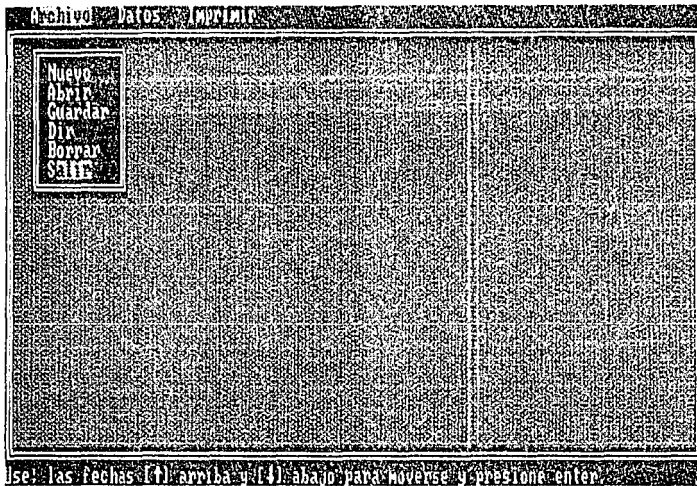


Figura 5.16 Opción para salir de la ejecución del preprocesador.

Continuando con la opción del menú principal encontramos a DATOS fig. 5.17, ubicamos todo lo referente a los datos que nos proporciona la estructura a analizar, recordamos al usuario que al considerar la solución de estados planos de esfuerzos por medio del elemento finito es indispensable discretizar la estructura mediante una serie de elementos cuadráticos (8 puntos) de la familia serendipity y en base a ellos obtener los siguientes datos.

FALLA DE ORIGEN

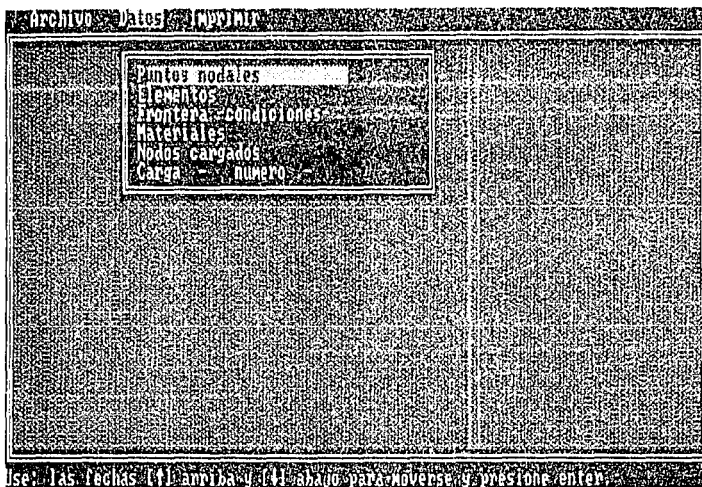


Figura 5.17 Menú de opciones de la opción DATOS.

PUNTOS NODALES:

Primeramente hay que posicionarse en la opción Puntos Nodales ya sea por medio del ratón, de las flechas direccionales y pulsando "enter" o por la primera letra resaltada de la opción en cuestión fig. 5.18.

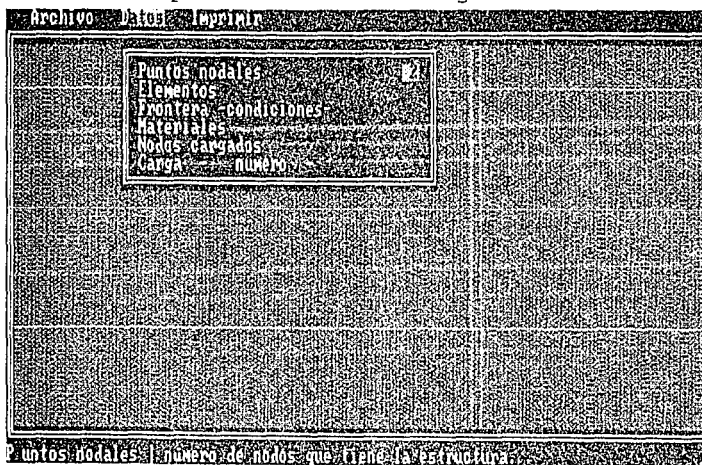


Figura 5.18 Definición del número de puntos nodales.

Una vez ubicados a este respecto, el programa esperará que indiquemos el número de nodos que tiene la estructura, nótese que este valor no puede ser nulo, y pulsemos "enter".

Finalmente el programa esperará todos y cada uno de los datos a introducir referentes a los puntos nodales de la estructura en cuanto a las coordenadas de cada punto nodal, es decir; la abscisa del punto nodal "X" y ordenada del punto nodal "Y" así sucesivamente para todos los puntos nodales fig. 5.19.

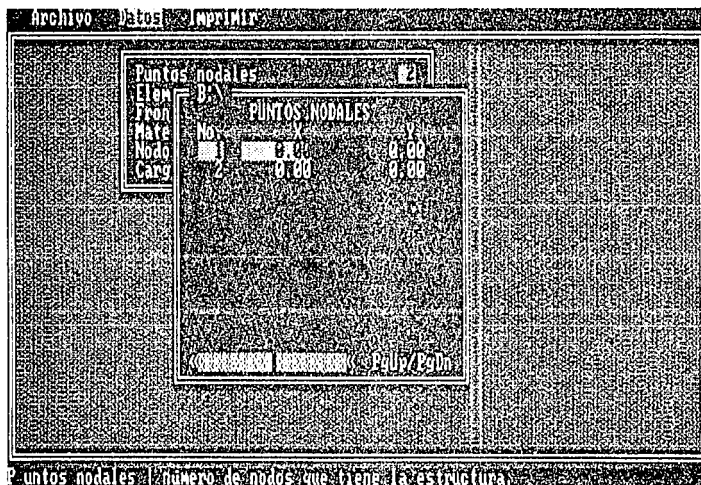


Figura 5.19 Pantalla de interacción para introducir las coordenadas de cada punto nodal.

La forma de introducir los datos será moviéndonos con la flechas direccionales a lo largo de la pantalla activa y pulsando "enter" en la posición que nos ha de interesar, o bien, haciendo uso del cursor del ratón moviéndolo a la posición deseada y presionando el botón izquierdo. Cuando hayamos introducido el valor correspondiente solamente

bastará con presionar la tecla "enter" para que el programa lo almacene en la memoria y quede completamente capturado. Para salir de esta opción y regresar al menú anterior sólo hay que oprimir la tecla "escape".

ELEMENTOS:

Mediante los medios que ya conocemos activamos esta opción logrando que el programa nos pregunte el número de elementos con que cuenta nuestra estructura fig. 5.20.

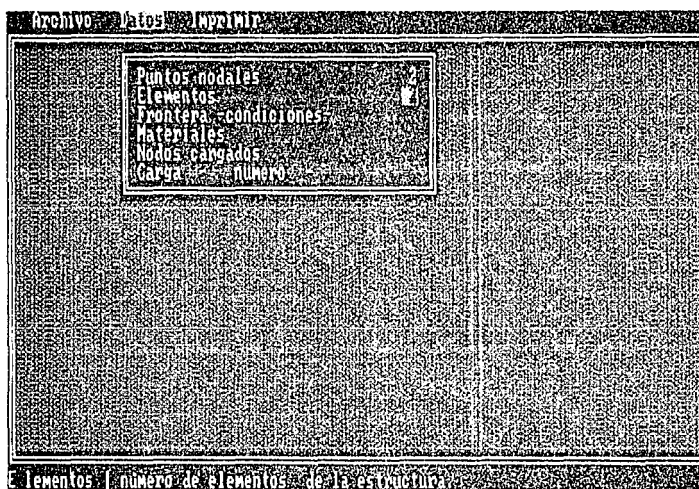


Figura 5.20 Definición del número de elementos que conforman a la estructura.

A continuación presionamos "enter", nos conducirá a la ventana donde introduciremos los datos, los elementos deben de estar en forma secuencial, la manera de acceder a cada una de las posiciones de los datos puede ser mediante las flechas o mediante el uso del ratón para que se presione "enter" al principio y al final de la introducción de los datos quedando finalmente capturado fig. 5.21. En particular esta opción

necesita que se le indique el orden de los nodos que conforman a cada elemento y el número de material que le corresponde a cada elemento.

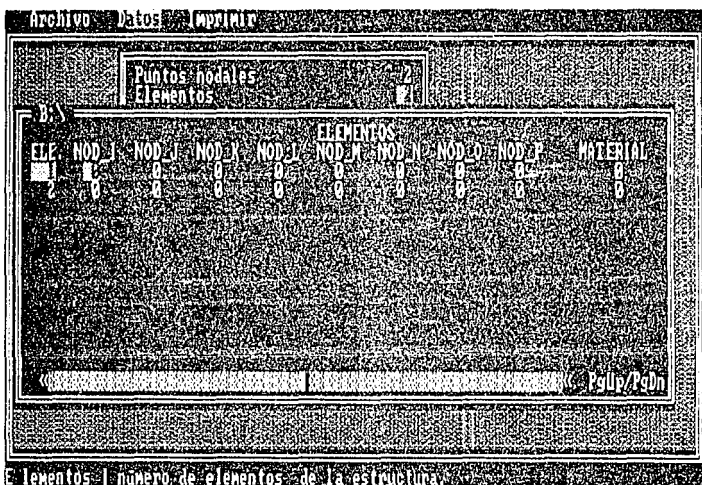


Figura 5.21 Pantalla de interacción para la captura de los puntos nodales de cada elemento.

Si deseamos continuar con la ejecución del programa debemos presionar la tecla "escape" para poder regresar al menú anterior.

FRONTERA:

Moviéndonos a la opción de frontera y presionando "enter" nos pedirá el número de puntos que son restringidos de desplazamiento dentro de nuestra estructura fig. 5.22.

Nota: este número de datos depende del total de puntos nodales.

FALLA DE ORIGEN

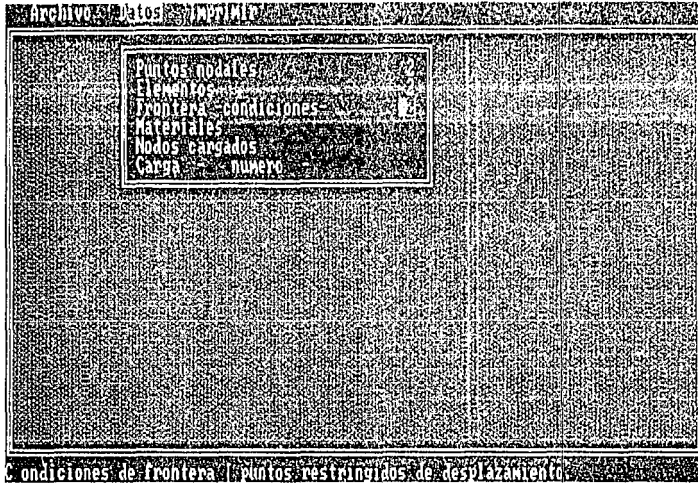


Figura 5.22 Definición de los puntos nodales.

Al terminar de introducir esta cantidad bastará con que presionemos "enter" para que el programa nos conduzca a la parte donde podemos introducir los datos pertenecientes a este respecto fig. 5.23.

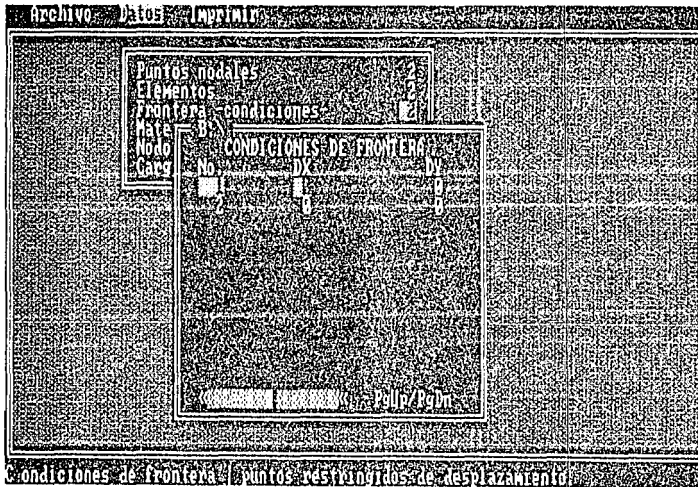


Figura 5.23 Ubicación de los puntos nodales restringidos de desplazamiento.

Las flechas direccionales nos conducirán a un pleno movimiento a lo largo de la ventana activa de datos y el ratón nos producirá el mismo resultado sólo que lo hará instantáneamente siendo un poco más rápido y práctico.

Los datos aquí pedidos nos indican, como sabemos, que cada punto nodal cuenta con dos grados de libertad, un desplazamiento en la dirección x y un desplazamiento en la dirección y , por lo que cada punto debe de tener una condición de desplazamiento, puede ser desplazamiento libre 0 o desplazamiento restringido 1.

Por ello aquí definimos los puntos nodales que tienen desplazamiento 0 ó 1.

Presionamos "enter" en cada una de las posiciones que deseamos modificar y posteriormente volvemos a presionar "enter" para finalizar la modificación del dato en la posición que nos encontramos y al finalizar todas las modificaciones presionamos "escape" para regresar al menú anterior.

MATERIALES:

Al hacer uso de esta opción tendremos que introducir el número de materiales que tiene la estructura fig. 5.24.

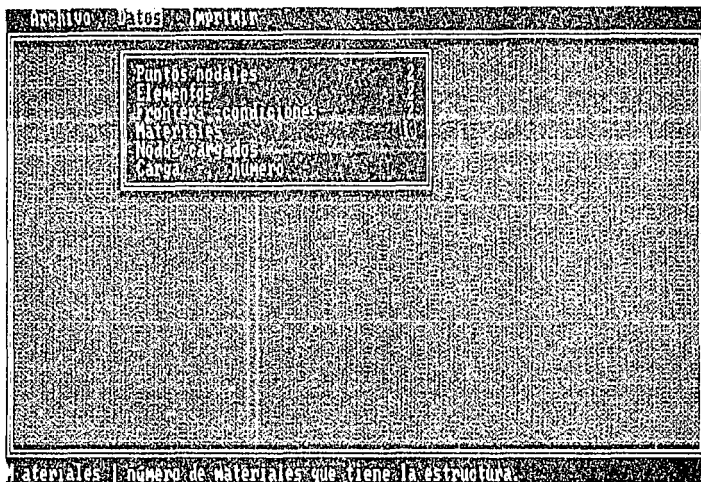


Figura 5.24 Estableciendo el número de materiales.

Una vez introducido este número presionamos la tecla "enter" para poder observar la ventana que hace alusión a estos datos en ella vamos a encontrar la parte donde se pregunta el módulo de elasticidad, la relación de poisson de cada material y su espesor fig. 5.25.

A cada material se le asigna un número en forma secuencial al que servirá de referencia para identificar al tipo de material de cada elemento como fue introducido en la opción de elementos.

Los movimientos permitidos por el programa son en esencia los mismos que los anteriores: hacer uso de las flechas direccionales para movernos de una posición a otra y pulsar "enter" cuando llegemos a ella o bien, por medio del cursor del ratón situarnos en la posición deseada y presionar el botón izquierdo para poder introducir nuestro dato.

FALLA DE ORIGEN

No.	E	ESPESOR
1	0.00	0.00
2	0.00	0.00
3	0.00	0.00
4	0.00	0.00
5	0.00	0.00
6	0.00	0.00
7	0.00	0.00
8	0.00	0.00
9	0.00	0.00
10	0.00	0.00

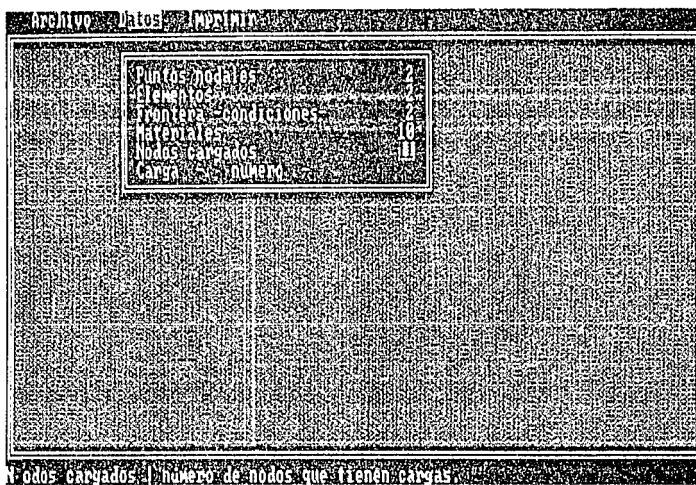
Figura 5.25 Pantalla de interacción para la captura de los datos de cada material.

Al terminar la captura de ese dato es preciso presionar una vez más "enter" para que el programa lo cargue en memoria, y podamos modificar algún otro dato. Si se desea salir hay que presionar la tecla "escape" para regresar al menú anterior.

NODOS CARGADOS:

El activar dicha opción produce la espera de la computadora por el número de nodos que tienen cargas, al terminar de teclear este número hay que presionar "enter" para que el programa abra la ventana de datos de la que estamos haciendo uso fig. 5.26.

FALLA DE ORIGEN



Nodo capturado: el número de nodos que tienen cargas.

Figura 5.26 Definición del número de nodos al que pertenecen las cargas.

Una vez abierta esta ventana podemos observar que se nos presenta en forma vertical el número de cargas y el nodo al que le corresponde dicha carga, el cual podemos modificar mediante nuestros desplazamientos por medio de las flechas direccionales o mediante el recurso del ratón fig. 5.27.

Quando encontremos un nodo a modificar debemos presionar "enter" para poder cambiarlo y al terminar debemos de volver a presionar la tecla "enter" para que quede capturado. Si deseamos salir de esta opción hay que presionar "escape" y estaremos de vuelta en el menú de datos.

FALLA DE ORIGEN

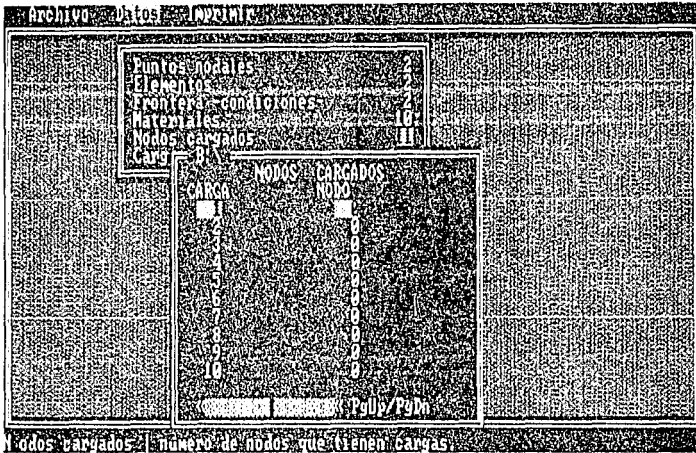


Figura 5.27 Asignación de las cargas a los nodos.

CARGAS -NUMERO- :

Finalmente esta opción opera igual que las demás ya que nos pedirá el número de cargas con que cuenta nuestra estructura, hay que introducirlo finalizando con la presión de la tecla "enter" fig. 5.28.

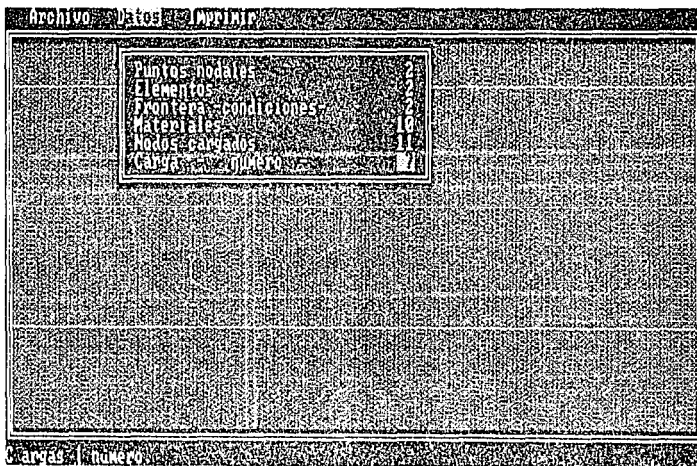


Figura 5.28 Definición del número de cargas.

COPIA DE ORIGEN

El uso de las flechas direccionales o del ratón nos permitirán modificar cualquier dato que en la ventana activa se encuentre presente, recordando que para que un dato quede capturado en memoria es necesario primero colocarse en la posición a donde se encuentra el dato a corregir, segundo presionar "enter" si el movimiento fue con las flechas en caso contrario estará listo para la captura, tercero introducir el dato correcto y al terminar volver a presionar la tecla "enter", bajo estos pasos el programa grabará en memoria el dato que hayamos modificado fig. 5.29. Si deseamos salir de esta opción y probar cualquier otra debemos de presionar la tecla "escape".

Cuando deseemos salir del menú de datos debemos de presionar "escape" o dar un click con el botón izquierdo del ratón pero apuntando afuera del menú de datos.

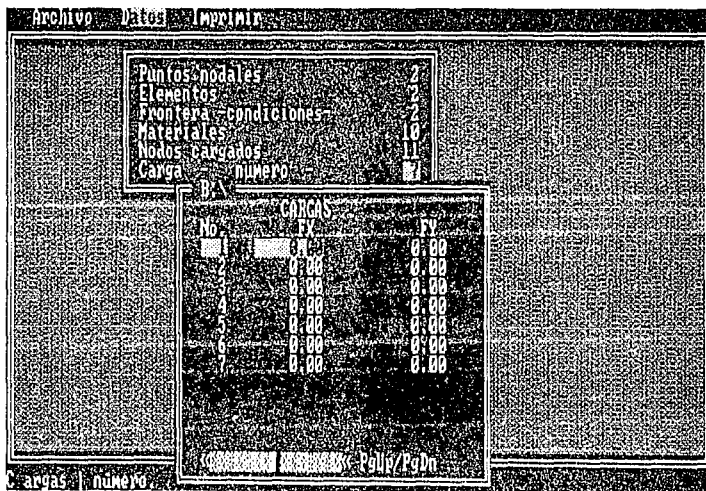


Figura 5.29 Pantalla para la captura de la dirección de las cargas.

FALLA DE ORIGEN

La última opción del menú principal que falta por describir es la de IMPRIMIR fig. 5.30, aquí el programa mandará los datos cargados en la memoria -provenientes ya sea desde un archivo en disco previamente almacenado y que fue abierto o provenientes de la memoria directamente- a la impresora, donde aparecerán tal y como fueron introducidos en su momento fig. 5.31.

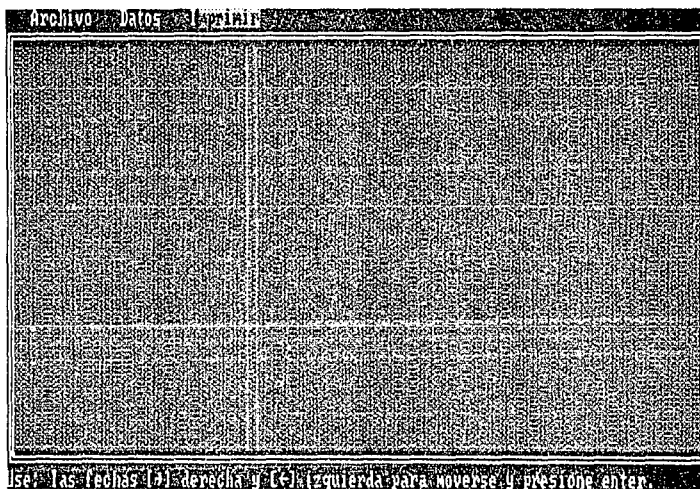


Figura 5.30 Pantalla que muestra el acceso a la opción principal IMPRIMIR.

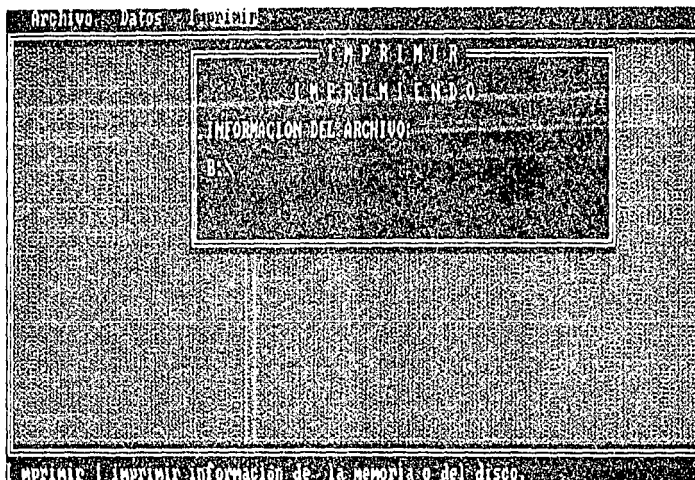


Figura 5.31 Redireccionamiento a la impresora de los datos cargados en memoria.

5.3 Procesador.

El procesador es el segundo programa del sistema, el término procesador de manera general lleva implícita la acción de procesar o sea de manipular datos en la computadora, dicha manipulación engloba una acción de utilizarlos para llevar a cabo uno que otro cálculo matemático.

5.3.1 Generalidades.

El procesador que en esta ocasión lo llamamos PROCESA (elaborado en el lenguaje FORTRAN) hace referencia a los cálculos matemáticos necesarios que requieren ser llevados a cabo por el MELEFIN. En base a los datos almacenados y

FALLA DE ORIGEN

puestos a disposición (en disco) por el preprocesador, el procesador efectuará mediante ésta información los cálculos y el almacenamiento de los resultados de las matrices de rigideces tanto de cada elemento como de la estructura; también almacenará las matrices de deformaciones de cada elemento, la matriz de ensamble de la estructura, el vector de cargas, el de desplazamientos y los esfuerzos que eran necesarios llegar a calcular. Todo lo anterior se encuentra almacenado en un archivo de resultados para llegar a ser consultados en cualquier momento por parte del usuario; además de hacer uso de un archivo extra donde se guardarán los resultados de las operaciones de multiplicación de matrices para evitar la pérdida de tiempo en su cálculo al emplearlas en diferentes ocasiones y que por su almacenamiento sea más fácil y rápida su consulta ya que el MELEFIN basa sus procedimientos básicamente en la multiplicación de matrices. Por todo lo anterior el procesador cumple su objetivo cuando calcula los esfuerzos y desplazamientos de la estructura que se analiza por lo que es trabajo del usuario analizar estos resultados y elaborar sus propias conclusiones.

5.3.2 Organización.

El procesador fue realizado en el lenguaje Fortran, por su rápida compilación, porque genera archivos ejecutables para poder correrlos desde un sistema operativo, por su gran

capacidad y aproximación en el manejo de grandes cantidades de datos y además por el manejo de subrutinas para hacer más fácil la programación. El programa en general consta de 20 subrutinas propias definidas por el programador, las cuales se describen a continuación:

PROCESA.FOR

PRESENTA
ABRIRA
CD
CROUT
DATINIC
ENSAM
ESFUER8
FINAR
FUERZA
IMATR1
IMAR2
IND
LECAR
LECTMAT
MATB8
MATK8
MATK88
MATRAN
MULMAT
REVISA

La subrutina -PRESENTA- muestra los datos referentes al nombre del programador, nombre del asesor del programa, datos referentes a la escuela de procedencia así como el nombre del programa y su significado.

La subrutina -ABRIRA- pide el nombre de un archivo, para poder abrirlo, leyendo el drive en que se encontrará o se guardará según sea el caso.

La subrutina -CD- se encarga de calcular la matriz D la cual hace referencia a la matriz de estados planos de esfuerzos.

La subrutina -CROUT- resuelve el sistema de ecuaciones de orden $N \times N$ por el método de Gauss-Crout.

La subrutina -DATINIC- lee del archivo proporcionado por el preprocesador *.fmt el número de puntos nodales, número de elementos, número de materiales, número de nodos cargados y número de las cargas.

La subrutina -ENSAM- ensambla la matriz de rigideces de la estructura, mediante las matrices de rigideces de cada elemento.

La subrutina -ESFUER8- calcula los esfuerzos de cada elemento y de cada combinación de $c_{1,2}$ y de $n_{1,2}$.

La subrutina -FINAR- cierra el archivo cuya unidad le es especificada.

La subrutina -FUERZA- genera el vector de cargas de la estructura.

La subrutina -IMATR1- imprime los datos de la matriz A (que es la que se le asigna como parámetro) en la pantalla.

La subrutina -IMATR2- al igual que la anterior graba los datos de la matriz A en un archivo cuya unidad es pasada como parámetro.

La subrutina -IND- calcula los indicadores de ecuación del sistema, guardándolos en una matriz resultante.

La subrutina -LECAR- lee el valor del vector de datos del archivo especificado desde NI hasta NF-1.

La subrutina -LECMAT- lee una matriz A con la dimensión NR número de renglones por NC número de columnas, pidiendo cada uno de los datos que la componen.

La subrutina -MATB8- contiene las fórmulas de las funciones de forma, las fórmulas de los puntos gaussianos y demás para calcular la matriz B.

La subrutina -MATK8- calcula la matriz de rigideces de cada elemento.

La subrutina -MATK8- en base a las subrutinas -ENSAM- y MATK8- calcula y ensambla las matrices de rigideces de cada elemento para obtener la matriz de rigideces de la estructura.

La subrutina -MATRAN- calcula la transpuesta de la matriz que se le pasa como argumento.

La subrutina -MULMAT- realiza la multiplicación de las matrices que le son pasadas como argumentos, guardando el resultado en una tercera matriz.

La subrutina -REVISA- verifica que el dimensionamiento de A, de la memoria dinámica dado como constante, sea mayor que el dimensionamiento que utiliza el programa, el cual depende de las dimensiones de la estructura.

5.3.3 Manual del usuario.

El manejo del procesador (llamado PROCESA) es sencillo por los pocos datos que usa, ya que el usuario solamente debe saber como se llama su programa *.fmt el cual contiene los

datos almacenados por el preprocesador para que este programa solamente los tome y lleve a cabo sus cálculos.

El usuario debe teclear A:>PROCESA y presionar "enter" para que en la primera pantalla aparezca el nombre de la escuela, el nombre del programa, los nombres tanto del programador como del asesor fig. 5.32, etc.

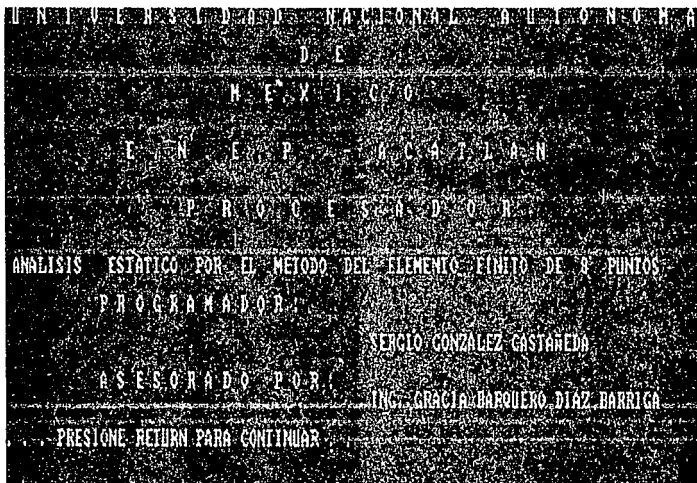


Figura 5.32 Pantalla de presentación del procesador.

La segunda pantalla de presentación corre a cargo del nombre del programa con sus respectivas siglas para distinguir el objetivo del programa fig. 5.33 que se esta ejecutando.

FALLA DE ORIGEN



Figura 5.33 Pantalla que muestra el significado a las siglas que conforman al nombre del procesador.

Finalmente pedirá en secuencia el drive donde se encuentra el archivo de datos *.fmt para su utilización, para poder proceder a abrirlo (para leer su contenido), en caso de algún error el programa lo marcará y detendrá su ejecución regresándonos al sistema operativo fig. 5.34.

FALLA DE ORIGEN

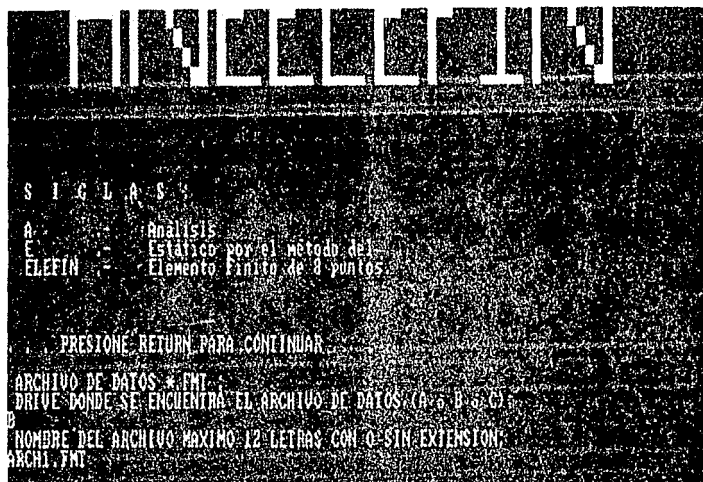


Figura 5.34 Pantalla de interacción para la modificación de la unidad y nombre del archivo a consultar.

Posteriormente el procesador indicará al usuario que introduzca el drive donde será almacenado el archivo de resultados así como el nombre que a éste se le dará, después de introducir tanto la letra del drive como el nombre del archivo (hasta 8 letras sin extensión ó 12 con extensión) hay que teclear "return" para indicarle al programa que lo procese. De igual manera necesitará que se indique el drive y el nombre del archivo auxiliar donde guardará los resultados de las multiplicaciones de matrices para su empleo posterior fig. 5.35.

FALLA DE ORIGEN

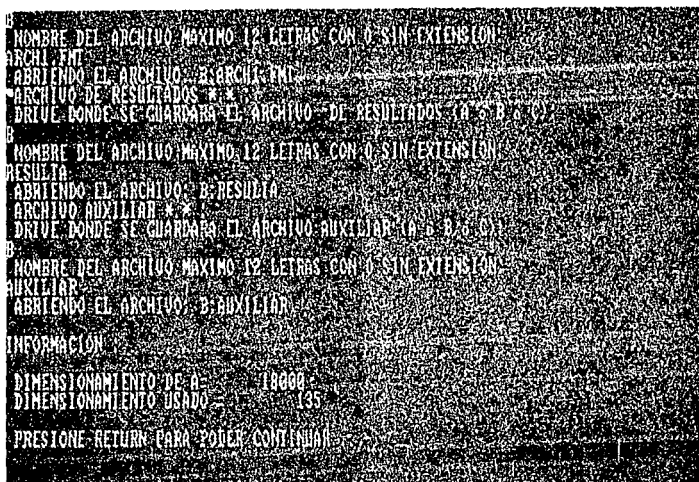


Figura 5.35 Pantalla que describe la unidad de trabajo y nombre de los archivos a generar así como el dimensionamiento usado por el procesador.

En la fig. 5.35, muestra el procesador el dimensionamiento usado dado que estamos empleando un dimensionamiento dinámico es preciso mencionar al usuario la cantidad empleada en los cálculos pidiéndonos que presionemos "enter" para poder continuar.

Finalmente el procesador realiza todos los cálculos necesarios para elaborar el archivo de resultados que el usuario le especifico que generará fig. 5.36. Es importante mencionar que el tiempo de ejecución del programa depende de la magnitud del problema.

FALLA DE ORIGEN

```
NOMBRE DEL ARCHIVO MAXIMO DE GERACION DE RESULTADOS EN EL ARCHIVO  
AUXILIAR  
ABRIENDO EL ARCHIVO DE AUXILIAR  
INFORMACION  
DIMENSIONAMIENTO DE A= 18000  
DIMENSIONAMIENTO USADO = 135  
PRESTONE RETURN PARA PODER CONTINUAR  
NC = 40  
INFORMACION  
DIMENSIONAMIENTO DE A= 18000  
DIMENSIONAMIENTO USADO = 1775  
PRESTONE RETURN PARA PODER CONTINUAR  
FIN DEL PROCESO  
B:\
```

Figura 5.36 Pantalla que muestra el direccionamiento completo empleado al resolver el problema.

Por último el procesador terminara su ejecución regresándonos al sistema operativo para que si es deseo del usuario pueda consultar el archivo de resultados y el auxiliar.

FALLA DE ORIGEN

6.- EJEMPLOS DE APLICACION.

Con la finalidad de mostrar el uso del sistema, presentar resultados y resolver un problema específico dentro del mismo contexto se ha dejado esta tarea al capítulo de ejemplos de aplicación. Es importante mencionar que en base a los capítulo I, II y III se cuenta con las herramientas para llevar a cabo la resolución del problema de manera manual, para esta ocasión solamente se incluyen las entradas/salidas de los programas de aplicación, los resultados a los que llega el sistema han sido verificados al llevar a cabo todo el procedimiento de solución de manera manual, con la finalidad de poder contar con resultados verídicos y confiables.

6.1 Problema "A".

Para el problema "A" nos referiremos a una viga empotrada con una carga puntual con las características geométricas y tipo de material mostrado en la fig. 6.1.

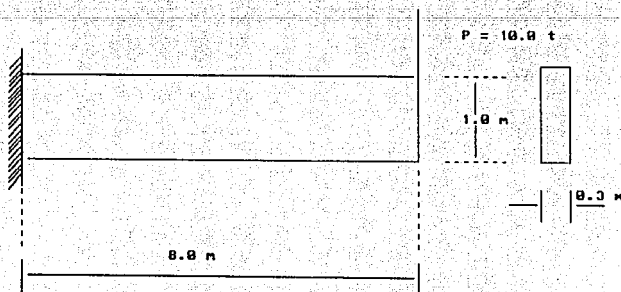


Figura 6.1 Viga empotrada en la pared.

Los datos referentes a esta estructura son los siguientes:

$$E = 2.0 \times 10^6 \text{ t/m}^2 \quad \nu = 0.2$$

$$\text{Espesor} = 0.3 \text{ m}$$

En base a los datos de la estructura anterior podemos observar que ante las condiciones de cargas a las que es sometida requiere hacerse un análisis de medios continuos para resolver las incógnitas de equilibrio entre esfuerzo-deformación, por tal motivo recurriremos al uso del método del elemento finito. El siguiente paso a seguir es analizar la estructura por medio de una discretización de la misma en elementos cuadráticos (8 puntos nodales) para poder introducir los datos que el preprocesador ha de requerir.

6.1.1 Datos para el preprocesador.

Dado que el análisis de la estructura está ligada con los datos que ha de requerir el preprocesador, lo haremos en esta parte. Primeramente la viga se discretiza en 4 elementos como se muestra en la fig. 6.2, es importante mencionar que dado que cada elemento está conformado por 8 puntos nodales (cuadráticos) brinda una muy buena aproximación a la resolución del problema por lo que no hay necesidad de dividir la estructura en demasiados elementos. Ya que está división dependerá de la forma de la estructura.

A continuación presentamos la estructura ya discretizada en sus elementos respectivos fig. 6.2.

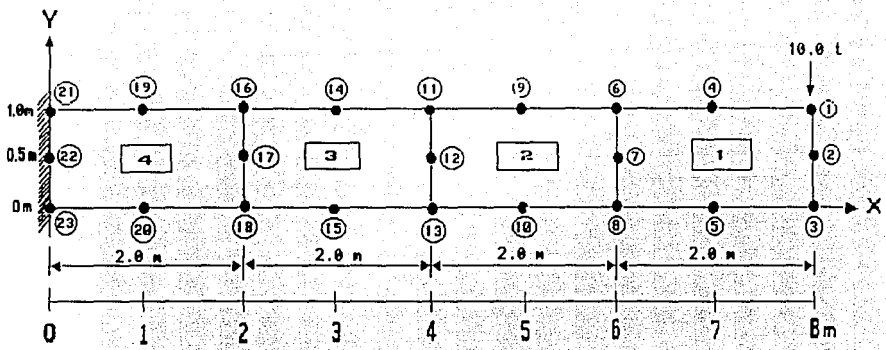


Figura 6.2 Estructuración en puntos nodales de la viga empotrada en la pared.

De la anterior figura podemos observar que los datos que nos pedirá el preprocesador serán los siguientes:

PUNTOS NODALES

Recordemos que hay que introducir las coordenadas referentes a cada punto nodal, es decir, para los 23 nodos hay que introducir la abscisa al punto nodal respectivo y la ordenada del mismo ya que estamos hablando de coordenadas en el plano X,Y.

Datos para el preprocesador:

NODO	X	Y
1	8.0	1.0
2	8.0	0.5
3	8.0	0.0
4	7.0	1.0
5	7.0	0.0
6	6.0	1.0
7	6.0	0.5
8	6.0	0.0
9	5.0	1.0
10	5.0	0.0
11	4.0	1.0
12	4.0	0.5
13	4.0	0.0

14	3.0	1.0
15	3.0	0.0
16	2.0	1.0
17	2.0	0.5
18	2.0	0.0
19	1.0	1.0
20	1.0	0.0
21	0.0	1.0
22	0.0	0.5
23	0.0	0.0

ELEMENTOS

Aquí tendremos que enfocarnos en los 4 elementos que dividen a nuestra estructura, debemos distinguir cada uno de los puntos nodales que conforman a cada elemento, para el preprocesador la secuencia de estos nodos en el elemento al que corresponden es muy importante ya que hay que introducirlos mediante la siguiente manera: para cada elemento el orden será de izquierda a derecha de abajo hacia arriba comenzando desde el nodo mayor que lo componen hasta completar el recorrido en el sentido contrario a las manecillas del reloj.

También debemos de especificar el material que componen a cada uno de los elementos, el material se refiere al módulo de elasticidad (representada por el símbolo E), el valor de NU del material (representado por ν) y el espesor del material.

Datos para el preprocesador:

ELE	NOD_I	NOD_J	NOD_K	NOD_L	NOD_M	NOD_N	NOD_O	NOD_P	MATERIAL
1	8	5	3	2	1	4	6	7	1
2	13	10	8	7	6	9	11	12	1
3	18	15	13	12	11	14	16	17	1
4	23	20	18	17	16	19	21	22	1

CONDICIONES DE FRONTERA

Es imprescindible tener en mente que cada punto nodal cuenta con dos grados de libertad, un desplazamiento en la dirección X y un desplazamiento en la dirección Y, por lo que debemos ser específicos en cuanto a sus desplazamientos, para poder distinguir los nodos libres de desplazamiento de los nodos restringidos les asignamos el siguiente código:

- 0 = desplazamiento libre.
- 1 = desplazamiento restringido.

Desde luego que debemos ser más específicos aún, ya que hay que indicar si el desplazamiento es en la dirección X ó Y. Con el código anterior definimos los puntos nodales restringidos que generalmente coinciden con ser puntos de frontera de la estructura, por último no hay que olvidar que se debe especificar este dato para cada uno de los puntos nodales que conforman nuestra estructura.

Datos para el preprocesador:

No.	X	Y
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0

20	0	0
21	1	1
22	1	1
23	1	1

Como podemos observar en la fig. 6.2 los únicos puntos restringidos de desplazamientos son los puntos nodales 21, 22, 23 ya que son los que por su posición en la estructura se encuentran empotrados en la pared, por lo que no cuentan con ningún desplazamiento en ninguna dirección.

MATERIALES

Dado que en ELEMENTOS hablamos del material que componen a cada elemento, debemos ser más concretos, especificando el módulo de elasticidad, el valor de poisson y el espesor de cada material que componen a nuestros elementos fig. 6.1. Por ejemplo en nuestro caso el material de que está compuesta nuestra estructura es uno solo.

Datos para el preprocesador:

No.	E	ν	Espesor
1	2000000.0	0.2	0.3

NODOS CARGADOS

Similar a la relación ELEMENTOS-MATERIALES encontramos otra más NODOS CARGADOS-CARGAS, en ésta parte debemos describir el número de cargas para especificar al nodo al que se relaciona directamente con la carga.

Datos para el preprocesador:

CARGA	NODO
1	1

La explicación a lo anterior es sencilla ya que según la fig. 6.2. observamos que la carga está hecha sobre el nodo 1 indicado por la flecha.

CARGAS -NUMERO-

Por último debemos de explicar la carga que en el anterior párrafo fue asignada al nodo "1". Un nodo admite una carga concentrada en la dirección X y otra en la dirección Y, dependiendo de la situación que se éste analizando. Al introducir la carga con su dirección correspondiente debemos de tomar en cuenta el sentido en el que se hace la carga.

Datos para el preprocesador:

No.	FX	FY
1	0.0	-10.0

La razón del -10.0 en la dirección Y de la carga "1" es sencilla ya que se tiene que el sentido en la dirección Y es negativo por que la carga se hace hacia abajo ver fig. 6.2.

Hasta aquí solamente se ha realizado un análisis preliminar de la estructura a analizar, los datos concretos a los que se ha formulado son las entradas para el preprocesador, por lo que solamente resta correr el programa del preprocesador (PREPRO) para poder almacenarlos en la computadora y continuar con el funcionamiento del sistema.

Continuamos con el preprocesador de la siguiente manera, encendemos la computadora e introducimos el diskette donde se encuentran almacenados nuestros programas de aplicación (PREPRO, PROCESA), en seguida desde el prompt de nuestro diskette tecleamos a:>PREPRO y damos "enter", el

programa comenzará a correr y cuando nos muestre el menú principal del programa donde se encuentran las opciones de ARCHIVO, DATOS E IMPRIMIR, comenzamos a capturar nuestros datos según lo requiera la opción de DATOS, después de introducir todos nuestros datos anteriores hacemos uso de la opción ARCHIVO para poder almacenarlos en el disco, dentro de la opción de ARCHIVO encontramos la opción GUARDAR, al entrar a ella daremos el nombre del archivo EJEMPLO1.DAT (puede ser cualquiera con extensión .dat) para que almacene los datos en forma legible para la computadora y además creará dos archivos más: uno lo llamará EJEMPLO1.FMT éste es el archivo con nuestros datos para el procesador y el otro será EJEMPLO1.CNS para que el usuario lo pueda consultar en pantalla y verificar sus datos introducidos al salir del preprocesador. Después de haber realizado todas las acciones anteriores, podemos salirnos del programa para correr el procesador o se puede continuar consultando las demás opciones que nos restan.

6.1.2 Datos para el procesador.

Los datos para el procesador se encuentran almacenados en un archivo con extensión .fmt (EJEMPLO1.FMT) ver apéndice B en LISTADO DE ARCHIVOS GENERADOS POR EL PREPROCESADOR Y PROCESADORES archivo B.2. El procesador deberá ser ejecutado desde el sistema operativo tecleando a:>PROCESA y al finalizar hay que presionar "enter", es importante estar

seguros que nuestro archivo de datos EJEMPLO1.FMT se encuentra en la raíz de nuestro disco de trabajo.

El procesador nos pedirá el drive donde se encuentra nuestro archivo de datos (A) y después nos preguntará por el nombre del archivo de datos (EJEMPLO1.FMT) para que posteriormente nos pida el drive (A) y el nombre del archivo (RESULTA1) donde queramos que direcciona nuestro archivo de resultados así como el drive (A) y el nombre del archivo auxiliar (AUXILIA1) para la realización de los cálculos del método del elemento finito.

Al terminar la ejecución del procesador el usuario podrá consultar los archivos de resultados y auxiliares (ver apéndice B archivo B.3 y archivo B.4 respectivamente) para poder observar y analizar los resultados a los que llegó en cuanto a los esfuerzos y desplazamientos.

6.1.3 Análisis de los resultados obtenidos por el procesador.

Para representar correctamente los resultados obtenidos por el procesador debemos de hacer uso de la graficación de nuestros resultados mediante el plano cartesiano X,Y.

Sin lugar a dudas, los datos para lograr este propósito son los desplazamientos y los esfuerzos. Por parte de los desplazamientos debemos de tomar en cuenta lo siguiente: cada nodo permite un desplazamiento en la dirección X y en la dirección Y desde su posición original ver fig. 6.2, como podemos observar en los resultados del archivo RESULTA1

(consultar el apéndice B archivo B.3.) contamos con 40 resultados en el vector de desplazamientos ya que los nodos libres de desplazamiento son del 1 al 20 y dado que el desplazamiento suele ser en la dirección X y Y para cada uno de los elementos entonces eso nos remarca la necesidad de tener 40 resultados dos de ellos para cada uno de los nodos de acuerdo a su numeración progresiva del 1 al 40. Basándonos en una graficación en torno a los resultados logramos obtener la siguiente figura 6.3. Donde podemos observar cuál es el nodo que presenta mayor desplazamiento y cuál es el de menor desplazamiento hasta llegar a distinguir los que siempre estuvieron restringidos de movimiento.

De una manera similar los esfuerzos también han de ser graficados a escala dependiendo del valor mayor y menor de los resultados obtenidos, aquí los puntos nodales estarán referenciados por los puntos gaussianos los cuales se sitúan cerca del punto nodal que coincide con las esquinas del elemento al que pertenecen, el valor asignado en cada una de estas esquinas está representado por el esfuerzo XX, como lo podemos ver en la fig. 6.4. logrando observar quién es el que cuenta con un mayor y un menor esfuerzo.

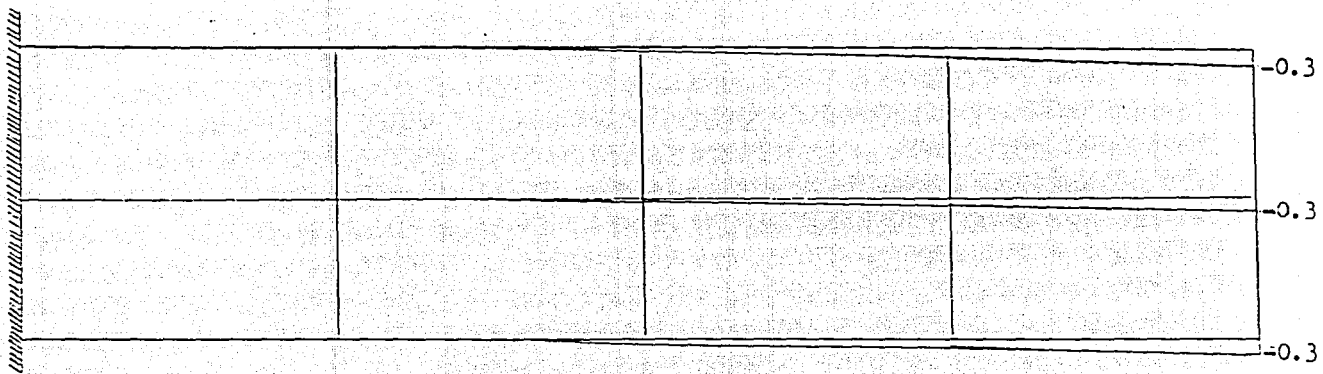


Figura 6.3 Representación de los desplazamientos calculados por el programa mediante el método del elemento finito.

876.26	742.63	644.48	511.12	413.4	279.96	182.25	48.82
4		3		2		1	
-876.25	-742.63	-644.48	-511.21	-413.4	-279.95	-182.24	-48.82

a) Calculado por el programa

876.59	742.93	645.10	511.46	413.65	280.05	182.30	48.84
4		3		2		1	
-876.56	-742.90	-645.07	-511.4	-413.63	-280.04	-182.29	-48.8

b) Calculado por formulas manualmente

Figura 6.4 -- Esquemas comparativos de los esfuerzos σ_x obtenidos para los puntos gaussianos discretizando la figura 6.1 en 4 elementos.

6.2 Problema "B".

En este ejemplo nos referiremos a una estructura empotrada en ambos lados con una carga puntual en el centro además de contar con las características geométricas y tipo de material mostrado en la fig. 6.5.

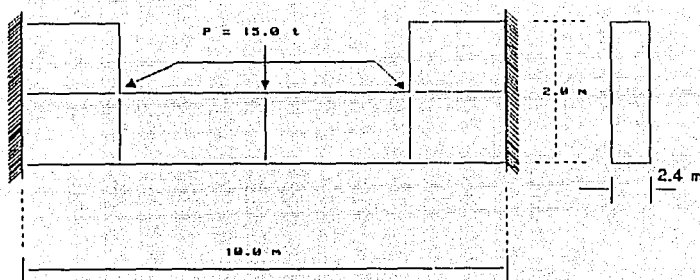


Figura 6.5 Viga empotrada de ambos lados en la pared.

Los datos referentes a esta estructura son los siguientes:

$$E = 2.0 \times 10^6 \text{ t/m}^2 \quad \nu = 0.2$$

$$\text{Espesor} = 2.4 \text{ m}$$

En base a los datos de la estructura anterior podemos observar que ante las condiciones de cargas a las que es sometida requiere al igual que el problema "A" realizar un análisis de medios continuos para resolver las incógnitas de equilibrio entre esfuerzo-deformación, por lo que utilizamos una vez más el método del elemento finito. El siguiente paso

a seguir es analizar la estructura por medio de una discretización de la misma en elementos cuadráticos (8 puntos nodales) para poder introducir los datos que el preprocesador ha de requerir.

6.2.1. Datos para el preprocesador.

Siguiendo los pasos del ejemplo anterior analizaremos la estructura. Primeramente la estructura se discretiza en 6 elementos como se muestra en la fig. 6.5.

A continuación presentamos la estructura ya discretizada en sus elementos respectivos formando la malla deseada fig. 6.6.

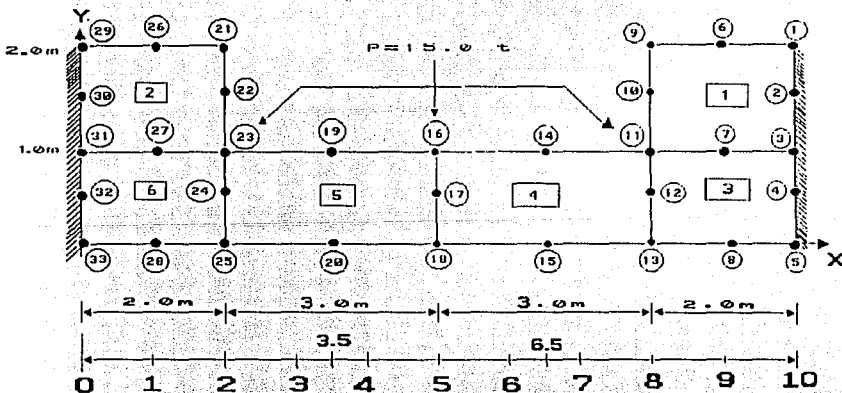


Figura 6.6 Estructuración en puntos nodales de la viga empotrada en la pared.

De la anterior figura podemos observar que los datos que nos pedirá el preprocesador serán los siguientes:

PUNTOS NODALES

Debemos introducir las coordenadas referentes a cada punto nodal, es decir, para los 33 nodos hay que introducir la abscisa al punto nodal respectivo y la ordenada del mismo ya que estamos hablando de coordenadas en el plano X,Y.

Datos para el preprocesador:

NODO	X	Y
1	10.0	2.0
2	10.0	1.5
3	10.0	1.0
4	10.0	0.5
5	10.0	0.0
6	9.0	2.0
7	9.0	1.0
8	9.0	0.0
9	8.0	2.0
10	8.0	1.5
11	8.0	1.0
12	8.0	0.5
13	8.0	0.0
14	6.5	1.0
15	6.5	0.0
16	5.0	1.0
17	5.0	0.5
18	5.0	0.0
19	3.5	1.0
20	3.5	0.0
21	2.0	2.0
22	2.0	1.5
23	2.0	1.0
24	2.0	0.5
25	2.0	0.0
26	1.0	2.0
27	1.0	1.0
28	1.0	0.0
29	0.0	2.0
30	0.0	1.5
31	0.0	1.0
32	0.0	0.5
33	0.0	0.0

ELEMENTOS

Enfocándonos en los 6 elementos que dividen a nuestra estructura, distinguiremos cada uno de los puntos nodales que

conforman a cada elemento, recordamos que para el preprocesador la secuencia de estos nodos en el elemento al que corresponden es muy importante ya que hay que introducirlos mediante la siguiente manera: para cada elemento el orden será de izquierda a derecha de abajo hacia arriba comenzando desde el nodo mayor que lo componen hasta completar el recorrido en el sentido contrario a las manecillas del reloj.

Indudablemente debemos de especificar el material que componen a cada uno de los elementos, el material se refiere al módulo de elasticidad (representada por el símbolo E), el valor de NU del material (representado por ν) y el espesor del material.

Datos para el preprocesador:

ELE	NOD_I	NOD_J	NOD_K	NOD_L	NOD_M	NOD_N	NOD_O	NOD_P	MATERIAL
1	11	7	3	2	1	6	9	10	1
2	31	27	23	22	21	26	29	30	1
3	13	8	5	4	3	7	11	12	1
4	18	15	13	12	11	14	16	17	1
5	25	20	18	17	16	19	23	24	1
6	33	28	25	24	23	27	31	32	1

CONDICIONES DE FRONTERA

Considerando el desplazamiento en la dirección X y el desplazamiento en la dirección Y para cada punto nodal, y distinguiendo los nodos restringidos de desplazamiento obtenemos lo siguiente:

Datos para el preprocesador:

No.	X	Y
1	1	1
2	1	1

3	1	1
4	1	1
5	1	1
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	1	1
30	1	1
31	1	1
32	1	1
33	1	1

Como podemos observar en la fig. 6.6 los puntos restringidos de desplazamientos son los puntos nodales 1, 2, 3, 4, 5, 29, 30, 31, 32, 33 ya que son los que por su posición en la estructura se encuentran empotrados en la pared, por lo que no cuentan con ningún desplazamiento en ninguna dirección.

MATERIALES

Siendo más específicos en cuanto al material que componen a cada elemento describimos los siguiente:

Datos para el preprocesador:

No.	E	v	Espesor
1	2000000.0	0.2	2.4

NODOS CARGADOS

Describiremos el número de cargas especificando al nodo al que hacen referencia.

Datos para el preprocesador:

CARGA	NODO
1	11
2	16
3	23

La explicación a lo anterior es sencilla ya que según la fig. 6.6. observamos que la carga está hecha sobre el nodo 11, 23, 16 indicado por las flechas.

CARGAS -NUMERO-

Por último debemos de remarcar que un nodo admite una carga concentrada en la dirección X y otra en la dirección Y, dependiendo de la situación que se éste analizando. Al introducir la carga con su dirección correspondiente debemos de tomar en cuenta el sentido en el que se hace la carga.

Datos para el preprocesador:

No.	FX	FY
1	0.0	-15.0
2	0.0	-15.0
3	0.0	-15.0

La razón del -15.0 en la dirección Y de la carga es sencilla ya que se tiene que el sentido en la dirección Y es negativo por que la carga se hace hacia abajo ver fig. 6.6.

Una vez introducidos estos datos en el preprocesador y guardados en un archivo en disco con el nombre EJEMPLO2.DAT podremos salirnos de él para poder ejecutar al procesador.

6.2.2 Datos para el procesador.

Los datos para el procesador se encuentran almacenados en un archivo con extensión .fmt (EJEMPLO2.FMT) ver apéndice B en LISTADO DE ARCHIVOS GENERADOS POR EL PREPROCESADOR Y PROCESADOR archivo B.6.

De la misma forma que en el ejemplo "A" el procesador nos pedirá el drive donde se encuentra nuestro archivo de datos (A) y después nos preguntará por el nombre del archivo de datos (EJEMPLO2.FMT) para que posteriormente nos pida el drive (A) y el nombre del archivo (RESULTA2) donde queramos que direccione nuestro archivo de resultados así como el drive (A) y el nombre del archivo auxiliar (AUXILIA2) para la realización de los cálculos del método del elemento finito.

Al terminar la ejecución del procesador el usuario podrá consultar los archivos de resultados y auxiliares (ver apéndice B archivo B.7 y archivo B.8 respectivamente) para poder observar y analizar los resultados a los que llegó en cuanto a los esfuerzos y desplazamientos.

6.2.3 Análisis de los resultados obtenidos por el procesador.

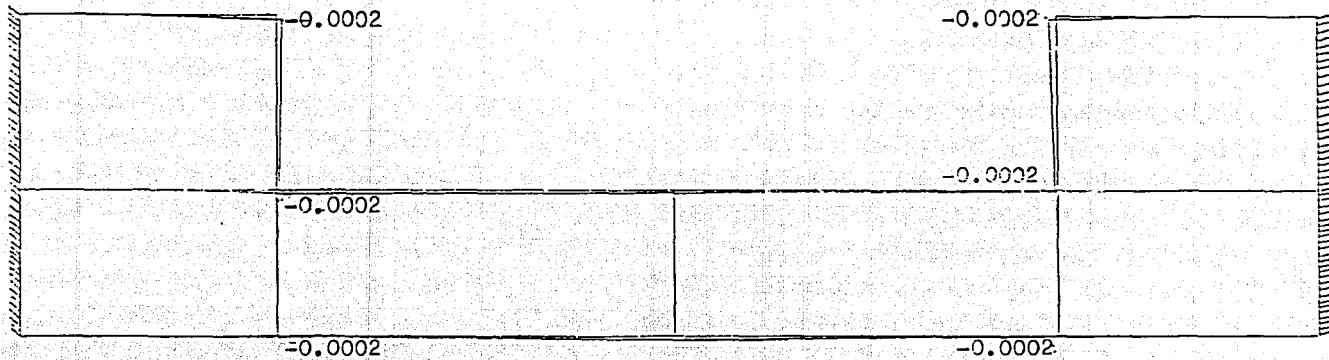
Graficando los resultados obtenidos por el procesador en dos partes: desplazamientos por una y esfuerzos por otra encontramos la explicación a las figuras 6.7 y 6.8.

Tomando en cuenta lo siguiente: cada nodo permite un desplazamiento en la dirección X y en la dirección Y por lo que como podemos observar en los resultados del archivo RESULTA2 (consultar el apéndice B archivo B.7) contamos con 46 resultados ya que los nodos libres de desplazamiento son del 6 al 28, dado que el desplazamiento suele ser en la dirección X y Y para cada uno de los puntos nodales, eso nos remarca la necesidad de tener 46 resultados dos de ellos para cada uno de los nodos de acuerdo a su numeración progresiva del 1 al 40 y comenzando la asignación desde el nodo 6. Basándonos en una graficación en base a los resultados logramos obtener la siguiente figura 6.7. Donde podemos observar cual es el nodo que presenta mayor desplazamiento y cual es el de menor desplazamiento hasta llegar a distinguir los que por su desplazamiento siempre estuvieron restringidos de movimiento.

De una manera similar los esfuerzos también han de ser graficados a escala dependiendo del valor mayor y menor de los resultados obtenidos, aquí los puntos nodales estarán referenciados por los puntos gaussianos los cuales se sitúan cerca del punto nodal que coincide con las esquinas del elemento al que pertenecen, el valor asignado en cada una de estas esquinas está representado por el esfuerzo XX, como lo

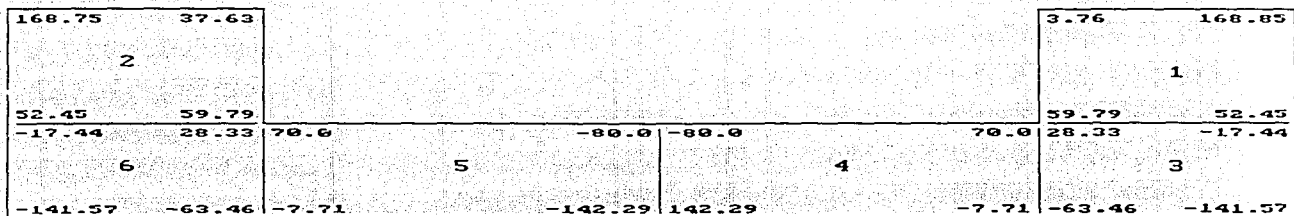
podemos ver en la fig. 6.8. logrando observar quién es el que cuenta con un mayor y un menor esfuerzo.

FALLA DE ORIGEN



— calculado por el programa

Figura 6.7 Representación de los desplazamientos calculados por el programa mediante el método del elemento finito.



calculado por el programa

Figura 6.8 Esquema representativo de los esfuerzos σ_x obtenidos para los puntos gaussianos discretizando la estructura de la figura 6.5 en 6 elementos.

CONCLUSIONES

El ámbito de desarrollo de la computación ha aumentado grandemente con el paso del tiempo, sus múltiples aplicaciones refuerzan las bases bajo las cuales han sido creadas, además de proporcionar el soporte técnico que permiten ampliar los conocimientos del hombre en cada uno de los campos bajo los cuales se desempeña.

Para tal caso, en un principio fue de primer instancia la preocupación de encontrar la manera de diseñar estructuras bajo diversas condiciones de carga ya que ello nos conduciría a lograr un equilibrio entre el esfuerzo y la deformación hasta lograr idear un diseño que nos lo permita.

Desde el punto de vista de la ingeniería se trató de idealizar este tipo de estructuras mediante dos maneras, la primera de ellas fue mediante un modelo estructural experimental lo cual resulta muy poco práctico, en segunda se consideró un modelo estructural numérico; después de encontrar diversos métodos cuyos resultados son bastante favorables, se han seguido utilizando dependiendo de las condiciones de la estructura a analizar, todos estos cálculos que involucran a cualquier método en específico han sido llevados a mano por lo que los errores de aproximación y el trabajo tedioso en el manejo de grandes cantidades de datos resultó estar presente en todo momento.

Por todo lo anterior y haciendo uso a la vez de varios conocimientos de ingeniería en el análisis de estructuras se

ha intentado automatizar mediante el uso de la computadora sólo un método numérico estructural, el cual cuenta con bastantes bases científicas que a su vez sirven de soporte a los resultados que de su aplicación se llegan a obtener. Por ello se eligió el método del elemento finito para resolver lo más aproximadamente posible las ecuaciones de equilibrio de esfuerzo y deformación para el análisis de los esfuerzos de medios continuos de una manera totalmente confiable.

El sistema que se presentó en este trabajo de investigación va enfocado principalmente a gente de ingeniería pero está explicado de tal manera que pueda ser comprendido por cualquier gente con conocimientos generales en computación para poder hacer un uso eficiente del sistema en cuestión.

El sistema justifica su uso por sí solo ya que la persona que tenga la necesidad de utilizarlo sólo tendrá que introducir a la computadora los datos de la estructura a analizar y completar la secuencia de operación del mismo, sin que el usuario del sistema tenga la necesidad de preocuparse en los cálculos y en el manejo de grandes cantidades de datos junto con sus aproximaciones respectivas, por lo que dejará de pensar en estos aspectos para enfocarse por completo en algo que requiera más de su atención como lo es el análisis de los resultados a los que llega el sistema.

El objetivo cubierto hasta el momento por parte del sistema ha sido resolver problemas de análisis estructurales de regular tamaño, con un grado de aproximación

bastante aceptable y llevando a cabo su tarea de la manera más rápida posible, el tiempo de solución del método del elemento finito dependerá del número de nodos con que cuenta la estructura, mientras que la malla que se diseña sobre la estructura a analizar dividiéndola en elementos no es necesario cerrarla demasiado, ya que dado que cada elemento cuenta con ocho puntos nodales esto nos ofrece un grado de aproximación bastante aceptable en nuestros resultados. Siendo así se logró cubrir un aspecto más allá de los propuestos, el cual se reflejó en la portabilidad del sistema ya que dado que su ejecución se puede hacer desde un sistema operativo DOS versión 3.3 en adelante, representa una gran ventaja para el usuario, ya que generalmente suele contar con una computadora personal con DOS versión 3.0 más o menos indistinta y 1 mega en RAM.

Estar conscientes todo el tiempo del uso y resultados que arroja el sistema ha de ser necesario para poder llegar a entender como funciona todo el tiempo, logrando establecer cómo, cuándo y dónde puede ser usado para poder interactuar con la computadora en sus actividades de trabajo cotidianas.

BIBLIOGRAFIA

- 1.-O. C. Zienkiewicz, THE FINITE ELEMENT METHOD, Third Edition, Mc Graw - Hill, 1977.
- 2.-K. J. Bathe y E. L. Wilson, NUMERICAL METHOD IN FINITE ELEMENT ANALYSIS, Prentice - Hall, 1976.
- 3.-L. J. Segerlind, APPLIED FINITE ELEMENT ANALYSIS, John Wiley, 1976.
- 4.-S. Timoshenko y J. N. Goodier, TEORIA DE LA ELASTICIDAD, Segunda edición, URMO, 1975.
- 5.-G. Salazar Polanco, ESTABILIDAD DE LAS CONSTRUCCIONES, Representaciones y servicios de Ingeniería, 1980.
- 6.-F. Vera Badillo, SOLUCION DE ECUACIONES ALGEBRAICAS EN EL ANALISIS ESTRUCTURAL, Tesis profesional, 1980.
- 7.-A. Ghali y A. Neville, ANALISIS ESTRUCTURAL, Primera Edición, Diana, 1980.
- 8.-Herbert Schildt, TURBO C/C++ MANUAL DE REFERENCIA, Mc Graw Hill, 1993.
- 9.-Francisco Javier Ceballos, ENCICLOPEDIA DEL LENGUAJE C, Addison - Wesley Iberoamericana, 1993.
- 10.-Roberto H. Hammond y William B. Rogers, INTRODUCCION AL FORTRAN 77 Y LA PC, Mc Graw - Hill, 1988.

A P E N D I C E A

**LISTADO DE PROGRAMAS
DE COMPUTADORA :**

**-PREPROCESADOR.
-PROCESADOR.**

PREPROCESADOR INCLUYE AL ARCHIVO: ANEELEFIN.C

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <stat.h>
#include <dir.h>
#include <errno.h>
```

```
#define atribut1 WHITE + (BLUE << 4)
#define atribut2 BLACK + (CYAN << 4)
#define atribut3 YELLOW + (BLUE << 4)
#define atribut4 BLACK + (LIGHTGRAY << 4)
#define atribut5 YELLOW + (LIGHTGRAY << 4)
#define atribut6 WHITE + (BLACK << 4)
#define atribut7 BLACK + (BLACK << 4)
#define atribut8 YELLOW + (GREEN << 4)
#define atribut9 WHITE + (BLUE << 4)
#define atribut10 RED + (LIGHTGRAY << 4)
#define atribut11 YELLOW + (BLACK << 4)
#define atribut12 YELLOW + (BLUE << 4)
#define atribut13 WHITE + (CYAN << 4)
#define atribut14 GREEN + (BLUE << 4)
#define atribut15 WHITE + (RED << 4)
#define atribut16 BLACK + (WHITE << 4)
#define atribut17 WHITE + (GREEN << 4)
#define atribut22 LIGHTBLUE + (BLUE << 4)
#define atribut23 LIGHTGREEN + (BLUE << 4)
#define ENTER 13
#define ESC 27
#define FLEADEL 205
#define FLEATRA 203
#define FLEARRI 200
#define FLEABAJ 208
#define ESQSUIZQ 201
#define ESQINIZQ 200
#define ESQSUDER 187
#define ESQINDER 188
#define LADOIZDE 186
#define LADOARAB 205
#define TABIQUE 177
#define HOME 199
#define PGUP 201
#define PGDN 209
#define END 207
#define INS 210
#define DEL 211
```

```
int menuhori(int cursor_x,int cursor_y,int auxbloc);
int menuvert(char *letre_ar[],char *carac2[],int valida2[],
int solo_ca2[],int auxblocl);
```

```

int fun_vali(int tecla,int posi_act[],int aux_men1_mouse[]);
int chec_car(int tecla,char *letra[],char *carac[]);
int chec_ca2(int tecla,char *letra[],char *carac[],int
posi_act[]);

int bloque;
int cursor_x,cursor_y;
char *men_prin[] = { "Archivo" , "Datos" , "Imprimir", NULL};
char *car_prin[] = { "A" , "D" , "I" , NULL };
char *men_arch[] = { "Nuevo" , "Abrir" , "Guardar" , "Dir" ,
"Borrar" , "Salir" , NULL };
char *car_arch[] = { "N" , "A" , "G" , "D" , "B" , "S" , NULL
} ;
char *men_dato[] = {
"Puntos nodales" , " ,
"Elementos" , " ,
"Frontera -condiciones-" , " ,
"Materiales" , " ,
"Nodos cargados" , " ,
"Carga - numero -" , " ,
NULL
} ;
char *car_dato[] = { "P" , "E" , "F" , "M" , "N" , "C" , NULL
} ;
char *car_carga[] = { "C" , "S" , NULL } ;
int teclprin[] = { ENTER , ESC , FLEADEL , FLEATRA , 65 , 68 ,
73 , NULL } ;
int letrprin[] = { 65 , 68 , 73 , NULL } ;
int teclarch[] = { ENTER , ESC , FLEARRI , FLEABAJ , 78 , 65
, 71 , 68 , 66 , 83 , NULL } ;
int letrarch[] = { 78 , 65 , 71 , 68 , 66 , 83 , NULL } ;
int tecldato[] = { ENTER , ESC , FLEARRI , FLEABAJ , 80 , 69
, 70 , 77 , 78 , 67 , NULL } ;
int letrdato[] = { 80 , 69 , 70 , 77 , 78 , 67 , NULL } ;
int teclcarga[] = { ENTER , ESC , FLEARRI , FLEABAJ , 67 , 83
, NULL } ;
int letrcarga[] = { 67 , 83 , NULL } ;
struct ventana {
int left;
int top;
int right;
int bottom;
char *titulo;
char *subtitulo;
int columnas;
char *formato;
char *posiciones;
};
struct ventana pun_nod = {
20,5,55,19,
" PUNTOS NODALES " ,
" No. X Y " ,
// 999 99999.99 99999.99
4,

```

```

"a%8.2fb%8.2f",
"n04x9y22"
};
struct ventana elementos = {
2,6,79,20,
"
ELEMENTOS
",
ELNOD_I NOD_J NOD_K NOD_L NOD_M NOD_N NOD_ONOD_P MATERIAL",
// 999 99 99 99 99 99 99 99 99
11,
"a%2.0fb%2.0fc%2.0fd%2.0fe%2.0fg%2.0fh%2.0fi%2.0fj%2.0f",
"n03x9y16z23a30b37c44d51e58f69"
};
struct ventana frontera = {
20,7,57,21,
"
CONDICIONES DE FRONTERA
",
" No. DX DY ",
// 999 99 99
4,
"a%2.0fb%2.0f",
"n04x15y30"
};
struct ventana materiales = {
20,8,69,22,
"
MATERIALES
",
" No. E v Espesor",
// 999 99999.99 99999.99 99999.99
5,
"a%8.2fb%8.2fc%8.2f",
"n04x10y24z38"
};
struct ventana nodos_cargados = {
20,9,57,23,
"
NODOS CARGADOS
",
" CARGA NODO ",
// 999 999
3,
"a%3.0f",
"n04x20"
};
struct ventana cargas = {
20,10,57,24,
"
CARGAS
",
" No. FX FY ",
// 999 99999.99 99999.99
4,
"a%8.2fb%8.2f",
"n04x10y24"
};
int dato_ini[] = { 0, 0, 0, 0, 0, 0, 0 };
double p_n_x_1[100],p_n_y_1[100];double p_n_x_2[100],
p_n_y_2[100],p_n_z_2[100],p_n_a_2[100],p_n_b_2[100],
p_n_c_2[100],p_n_d_2[100],p_n_e_2[100],p_n_f_2[100];
double p_n_x_3[100],p_n_y_3[100];
double p_n_x_4[100],p_n_y_4[100],p_n_z_4[100];

```

```

double p_n_x_5[100];
double p_n_x_6[100], p_n_y_6[100];
int top, lim_sup, lim_inf, tam_vent;
int reng_act, num_reng;
int posi_columna;
int n1, n2, n3, n4, n5, n6;
char *guarda[100];
int tamven_1, tamven_2;
char nombre_archivo[14];
char intento[80];
char path_archivo[80];
char aux_path[100];
char aux_path[80];
char cad_temp[80];
int columna;
int pri_mouse[] =
{ 24, 72, 0, 65, 104, 136, 0, 68, 168, 232, 0, 73, NULL};
int sec_mouse[] =
{40, 72, 24, 78, 40, 72, 32, 65, 40, 88, 40, 71, 40, 56, 48, 68, 40, 80, 56,
66, 40, 72, 64, 83, NULL};
int sec_1_mouse[] =
{120, 216, 24, 80, 120, 176, 32, 69, 120, 288, 40, 70, 120, 192, 48, 77,
120, 224, 56, 78, 120, 272, 64, 67, NULL};
int ter_mouse[] =
{160, 472, 80, 67, 160, 472, 88, 83, NULL};
int coord_archivos[] =
{96, 192, 80, 1, 1, 13, 96, 192, 88, 1, 2, 13, 96, 192, 96, 1, 3, 13,
96, 192, 104, 1, 4, 13, 96, 192, 112, 1, 5, 13, 96, 192, 120, 1, 6, 13,
96, 192, 128, 1, 7, 13, 96, 192, 136, 1, 8, 13, 96, 208, 152, 0, 0, 201,
216, 312, 80, 2, 1, 13, 216, 312, 88, 2, 2, 13, 216, 312, 96, 2, 3, 13,
216, 312, 104, 2, 4, 13, 216, 312, 112, 2, 5, 13, 216, 312, 120, 2, 6, 13,
216, 312, 128, 2, 7, 13, 216, 312, 136, 2, 8, 13, 208, 312, 152, 0, 0, 209,
NULL};
int coord_directorio[] =
{96, 184, 88, 1, 1, 13, 96, 184, 96, 1, 2, 13, 96, 184, 104, 1, 3, 13,
96, 184, 112, 1, 4, 13, 96, 184, 120, 1, 5, 13, 96, 184, 128, 1, 6, 13,
96, 184, 136, 1, 7, 13, 96, 184, 144, 1, 8, 13, 96, 200, 160, 1, 1, 201,
224, 312, 88, 2, 1, 13, 224, 312, 96, 2, 2, 13, 224, 312, 104, 2, 3, 13,
224, 312, 112, 2, 4, 13, 224, 312, 120, 2, 5, 13, 224, 312, 128, 2, 6, 13,
224, 312, 136, 2, 7, 13, 224, 312, 144, 2, 8, 13, 208, 312, 160, 1, 1, 209,
NULL};
int coord_puntos_nodales[] =
{216, 272, 56, 1, 4, 2, 13, 320, 376, 56, 1, 4, 3, 13,
216, 272, 64, 2, 5, 2, 13, 320, 376, 64, 2, 5, 3, 13,
216, 272, 72, 3, 6, 2, 13, 320, 376, 72, 3, 6, 3, 13,
216, 272, 80, 4, 7, 2, 13, 320, 376, 80, 4, 7, 3, 13,
216, 272, 88, 5, 8, 2, 13, 320, 376, 88, 5, 8, 3, 13,
216, 272, 96, 6, 9, 2, 13, 320, 376, 96, 6, 9, 3, 13,
216, 272, 104, 7, 10, 2, 13, 320, 376, 104, 7, 10, 3, 13,
216, 272, 112, 8, 11, 2, 13, 320, 376, 112, 8, 11, 3, 13,
216, 272, 120, 9, 12, 2, 13, 320, 376, 120, 9, 12, 3, 13,
216, 272, 128, 10, 13, 2, 13, 320, 376, 128, 10, 13, 3, 13,
176, 224, 144, 0, 0, 0, 201, 240, 288, 144, 0, 0, 0, 209, NULL};
int coord_elementos[] =

```

```

{72,80,64,1,4,2,13,128,136,64,1,4,3,13,
184,192,64,1,4,4,13,240,248,64,1,4,5,13,
296,304,64,1,4,6,13,352,360,64,1,4,7,13,
408,416,64,1,4,8,13,464,472,64,1,4,9,13,
552,560,64,1,4,10,13,72,80,72,2,5,2,13,
128,136,72,2,5,3,13,184,192,72,2,5,4,13,
240,248,72,2,5,5,13,296,304,72,2,5,6,13,
352,360,72,2,5,7,13,408,416,72,2,5,8,13,
464,472,72,2,5,9,13,552,560,72,2,5,10,13,
72,80,80,3,6,2,13,128,136,80,3,6,3,13,
184,192,80,3,6,4,13,240,248,80,3,6,5,13,
296,304,80,3,6,6,13,352,360,80,3,6,7,13,
408,416,80,3,6,8,13,464,472,80,3,6,9,13,
552,560,80,3,6,10,13,72,80,88,4,7,2,13,
128,136,88,4,7,3,13,184,192,88,4,7,4,13,
240,248,88,4,7,5,13,296,304,88,4,7,6,13,
352,360,88,4,7,7,13,408,416,88,4,7,8,13,
464,472,88,4,7,9,13,552,560,88,4,7,10,13,
72,80,96,5,8,2,13,128,136,96,5,8,3,13,
184,192,96,5,8,4,13,240,248,96,5,8,5,13,
296,304,96,5,8,6,13,352,360,96,5,8,7,13,
408,416,96,5,8,8,13,464,472,96,5,8,9,13,
552,560,96,5,8,10,13,72,80,104,6,9,2,13,
128,136,104,6,9,3,13,184,192,104,6,9,4,13,
240,248,104,6,9,5,13,296,304,104,6,9,6,13,
352,360,104,6,9,7,13,408,416,104,6,9,8,13,
464,472,104,6,9,9,13,552,560,104,6,9,10,13,
72,80,112,7,10,2,13,128,136,112,7,10,3,13,
184,192,112,7,10,4,13,240,248,112,7,10,5,13,
296,304,112,7,10,6,13,352,360,112,7,10,7,13,
408,416,112,7,10,8,13,464,472,112,7,10,9,13,
552,560,112,7,10,10,13,72,80,120,8,11,2,13,
128,136,120,8,11,3,13,184,192,120,8,11,4,13,
240,248,120,8,11,5,13,296,304,120,8,11,6,13,
352,360,120,8,11,7,13,408,416,120,8,11,8,13,
464,472,120,8,11,9,13,552,560,120,8,11,10,13,
72,80,128,9,12,2,13,128,136,128,9,12,3,13,
184,192,128,9,12,4,13,240,248,128,9,12,5,13,
296,304,128,9,12,6,13,352,360,128,9,12,7,13,
408,416,128,9,12,8,13,464,472,128,9,12,9,13,
552,560,128,9,12,10,13,72,80,136,10,13,2,13,
128,136,136,10,13,3,13,184,192,136,10,13,4,13,
240,248,136,10,13,5,13,296,304,136,10,13,6,13,
352,360,136,10,13,7,13,408,416,136,10,13,8,13,
464,472,136,10,13,9,13,552,560,136,10,13,10,13,
40,264,152,0,0,0,201,280,504,152,0,0,0,209,NULL};
int coord_frontera[] =
{264,272,72,1,4,2,13,384,392,72,1,4,3,13,
264,272,80,2,5,2,13,384,392,80,2,5,3,13,
264,272,88,3,6,2,13,384,392,88,3,6,3,13,
264,272,96,4,7,2,13,384,392,96,4,7,3,13,
264,272,104,5,8,2,13,384,392,104,5,8,3,13,
264,272,112,6,9,2,13,384,392,112,6,9,3,13,
264,272,120,7,10,2,13,384,392,120,7,10,3,13,

```



```

textcolor( LIGHTGREEN );
gotoxy(1,25);
cprintf("Use: las fechas [ ] derecha y [ ] izquierda
para moverse y presione enter.      ");
textcolor( WHITE );
caracter=26;
gotoxy(18,25);
cprintf("%c",caracter);
caracter=27;
gotoxy(32,25);
cprintf("%c",caracter);
bloque=menuhori(2,1,bloque);
switch(bloque)
{
case 0:
break;
case 1: DEL MENU HORIZONTAL PRESENTAMOS --- ARCHIVO
textcolor( LIGHTGREEN );
gotoxy(1,25);
cprintf("Use: las fechas [ ] arriba y [ ] abajo
para moverse y presione enter.      ");
textcolor( WHITE );
caracter=24;
gotoxy(18,25);
cprintf("%c",caracter);
caracter=25;
gotoxy(31,25);
cprintf("%c",caracter);
abre_ventana(4,3,16,11);
blockel=1;
while ( blockel != 0 )
{
blockel=menuvert (men_arch,car_arch,teclarch,
letrarch,blockel);
switch(blockel)
{
case 0: /*Teclamos solamente un ESC */
/* continue; */
break;
case 1: /*nuevo*/
window(1,1,80,25);
textattr(atribut3);
gotoxy(1,25);
cprintf("N uevo | Borra de la memoria todos
los datos .      ");
gettext(1,1,80,25,buf2);
delay(2500);
nuevo();
delay(2500);
break;
case 2: /*abrir*/
window(1,1,80,25);
textattr(atribut3);
gotoxy(1,25);

```

```

        cprintf("Abrir | Carga en memoria datos de
un archivo *.DAT especifico. ");
        gettext(1,1,80,25,buf2);
        abre_ventana(9,6,55,22);
        gotoxy(12,1);
        printf(" ABRIR ARCHIVO ");
        abrir_archivo();
        break;
    case 3: /*guardar*/
        window(1,1,80,25);
        textattr(atribut3);
        gotoxy(1,25);
        cprintf("Guardar | Guarda en disco los
datos cargados en la memoria. ");
        gettext(1,1,80,25,buf2);
        abre_ventana(9,7,55,22);
        gotoxy(3,1);
        printf("GUARDAR EN DISCO UN ARCHIVO *.DAT");
        guardar();
        break;
    case 4: /*dir*/
        window(1,1,80,25);
        textattr(atribut3);
        gotoxy(1,25);
        cprintf("Dir | Muestra los archivos
contenidos en un subdirectorio. ");
        gettext(1,1,80,25,buf2);
        abre_ventana(9,8,55,23);
        gotoxy(12,1);
        printf(" D I R E C T O R I O ");
        dir();
        break;
    case 5: /*borrar*/
        window(1,1,80,25);
        textattr(atribut3);
        gotoxy(1,25);
        cprintf("Borrar | Borra físicamente un
archivo del disco de trabajo. ");
        gettext(1,1,80,25,buf2);
        abre_ventana(9,9,55,21);
        gotoxy(4,1);
        printf(" B O R R A R un archivo *.* del
DISCO ");
        borrar();
        break;
    case 6: /*salir*/
        textattr(atribut1);
        window(1,1,80,25);
        mousin.x.ax = 10;
        mousin.x.bx = 1;
        mousin.x.cx = 6;
        mousin.x.dx = 7;
        int86(0x33, &mousin, &mousout);
        clrscr();

```

```

        exit(0);
        break;
    } /*fin del switch() de archivo*/
    if(bloque1 == 0 ) break;
    window(1,1,80,25);
    puttext(1,1,80,25,buf2);
    textattr( atribut23 );
    gotoxy(1,25);
    cprintf("Use: las fechas [ ] arriba y [ ] abajo
para moverse y presione enter.  ");
    textattr( atribut9 );
    caracter=24;
    gotoxy(18,25);
    cprintf("%c", caracter);
    caracter=25;
    gotoxy(31,25);
    cprintf("%c", caracter);
    textattr(atribut4);
    window(4,3,80,24);
} /*fin del while del menu de archivo */
window(1,1,80,25);
textattr(atribut1);
puttext(1,2,80,25,buf);
break;
case 2: DEL MENU HORIZONTAL PRESENTAMOS --- DATOS --
    textattr(atribut23);
    gotoxy(1,25);
    cprintf("Use: las fechas [ ] arriba y [ ] abajo
para moverse y presione enter.  ");
    textattr(atribut9);
    caracter=24;
    gotoxy(18,25);
    cprintf("%c", caracter);
    caracter=25;
    gotoxy(31,25);
    cprintf("%c", caracter);
    abre_ventana(14,3,51,11);
    bloque2=1;
    while (bloque2 != 0)
    {
    bloque2=menuvert(men_datos,car_datos,tecl_datos,
        letr_datos,bloque2);
        switch(bloque2)
        {
            case 0: /*Teclamos solamente un ESC */
                break;
            case 1: /*Puntos nodales*/
                window(1,1,80,25);
                textattr(atribut3);
                gotoxy(1,25);
                cprintf("P untos nodales | número de nodos
que tiene la estructura.  ");
                textattr(atribut4);
                window(14,3,80,21);

```

```

gettext(1,1,80,25,buf1);
nl=dato_ini[bloque2-1];
nl=determina_tamano_size(buf1,men_dato,
    car_dato,dato_ini,bloque2,nl);
if ( nl != 0 && nl > 0)
{
    window(pun_nod.left,pun_nod.top,
        pun_nod.right-2,pun_nod.bottom+1);
    textattr(atribut4);
    clrscr();
    fondo( (pun_nod.right - pun_nod.left-
1), (pun_nod.bottom+2 - pun_nod.top) );
    textattr(atribut4);
    gotoxy(3,1);
    printf(" %s ",path_archivo);
    gotoxy(2,2);
    textattr(atribut8);
    cputs(pun_nod.titulo);
    textattr(atribut4);
    gotoxy(2,3);
    cputs(pun_nod.subtitulo);
    textattr(atribut17);
    gotoxy(3,15);
    cprintf("«#####»");
    textattr(atribut4);
    gotoxy(24,15);
    cprintf("PgUp/PgDn");
    if (nl <= 10 )
    {
        ii=0;
        ultimo=(nl*14) - 1;
        for(i=0;i<=ultimo;i++)
        {
            temp[ii]=coord_puntos_nodales[i];
            ii++;
        }
        temp[ii]=0;
        tamano_ventana(nl,&pun_nod,p_n_x_1,p_n_y_1,
p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,p_n_e_2,
p_n_f_2,bloque2,temp);
    }
    else
    {
        tamano_ventana(nl,&pun_nod,p_n_x_1,p_n_y_1,
p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,p_n_e_2,
p_n_f_2,bloque2,coord_puntos_nodales);
    }
}
break;
case 2: /*Elementos*/
    window(1,1,80,25);
    textattr(atribut3);
    gotoxy(1,25);

```



```

case 3: /*Condiciones de frontera*/
window(1,1,80,25);
textattr(atribut3);
gotoxy(1,25);
cprintf("C ondiciones de frontera | puntos
restringidos de desplazamiento. ");
textattr(atribut4);
window(14,3,80,21);
gettext(1,1,80,25,buf1);
n3=dato_ini[bloque2-1];
n3=determina_tamano_size(buf1,men_dato,car_dato,
dato_ini,bloque2,n3);
if ( n3 != 0 && n3 > 0)
{
window(frontera.left,frontera.top,
frontera.right-2,frontera.bottom+1);
textattr(atribut4);
clrscr();
fondo( (frontera.right - frontera.left-
1),(frontera.bottom+2 - frontera.top) );
textattr(atribut4);
gotoxy(3,1);
printf(" %s ",path_archivo);
gotoxy(2,2);
textattr(atribut8);
cputs(frontera.titulo);
textattr(atribut4);
gotoxy(2,3);
cputs(frontera.subtitulo);
textattr(atribut17);
gotoxy(4,15);
cprintf("«#####»");
textattr(atribut4);
gotoxy(24,15);
cprintf("PgUp/PgDn");
if (n3 <= 10 )
{
ii=0;
ultimo=(n3*14) - 1;
for(i=0;i<=ultimo;i++)
{
temp[ii]=coord_frontera[i];
ii++;
}
temp[ii]=0;
tamano_ventana(n3,&frontera,p_n_x_3,p_n_y_3,p_n_z_2,p_n_a_2,p
_n_b_2,p_n_c_2,p_n_d_2,p_n_e_2,p_n_f_2,bloque2,temp);
}
else
{
tamano_ventana(n3,&frontera,p_n_x_3,
p_n_y_3,p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
p_n_e_2,p_n_f_2,bloque2,coord_frontera);
}
}
}

```



```

        else
            tamano_ventana(n6, &cargas, p_n_x_6,
                p_n_y_6, p_n_z_2, p_n_a_2, p_n_b_2, p_n_c_2,
                p_n_d_2, p_n_e_2, p_n_f_2, bloque2, coord_cargas);
        }
        break;
    }
    if( bloque2 == 0 ) break;
    window(1,1,80,25);
    puttext(1,1,80,25,buf1);
    textattr(atribut23);
    gotoxy(1,25);
    cprintf("Use: las fechas [ ] arriba y [ ] abajo
para moverse y presione enter. ");
    textattr(atribut9);
    caracter=24;
    gotoxy(18,25);
    cprintf("%c", caracter);
    caracter=25;
    gotoxy(31,25);
    cprintf("%c", caracter);
    textattr(atribut4);
    window(14,3,80,21);
    textcolor(BLACK);
    gotoxy(33, (bloque2+1));
    cprintf("%2d", dato_ini[bloque2-1]);
} /*fin del while de datos */
textattr(atribut1);
window(1,1,80,25);
puttext(1,2,80,25,buf);
break;
case 3: DEL MENU HORIZONTAL PRESENTAMOS --- IMPRIMIR
window(1,1,80,25);
textattr(atribut3);
gotoxy(1,25);
cprintf("I mprimir | imprimir información de la
memoria o del disco. ");
textattr(atribut4);
abre_ventana(22,3,72,14);
textattr(atribut16);
gotoxy(16,1);
printf(" I M P R I M I R ");
if( strlen(path_archivo) == 0 )
{
    gotoxy(3,3);
    cprintf("          I M P R I M I E N D O");
    textattr(atribut4);
    gotoxy(3,6);
    cprintf(" \" INFORMACION DE LA MEMORIA \" ");
}
else
{
    gotoxy(3,3);
    cprintf("          I M P R I M I E N D O");

```

```

        textattr(atribut4);
        gotoxy(3,5);
        printf("INFORMACION DEL ARCHIVO:" );
        gotoxy(3,7);
        printf("%s",path_archivo);
    }

    imprimir(dato_ini);
    window(1,1,80,25);
    textattr(atribut1);
    clrscr();
    puttext(1,2,80,25,buf);
    break;
} /* fin del swtch() del men_prin */
}while ( blockel != 6 );
}
/** MENU HORIZONTAL **/
int menuhori( cursor_x,cursor_y,auxbloc)
int cursor_x,cursor_y;
int auxbloc;

{
    int i=0;int long_let=0;int tecla;
    int aux_bloc=1;int algo=0;
    cursor_x=4; cursor_y=1;

    while ( men_prin[i] != NULL )
    {
        textcolor(WHITE);
        cursor_x+=long_let;
        gotoxy(cursor_x,cursor_y);
        cprintf("%s",men_prin[i]);
        textcolor(YELLOW);
        gotoxy(cursor_x,cursor_y);
        cprintf("%s",car_prin[i]);
        long_let=strlen(men_prin[i])+3;
        i+=1;
    }
    if (auxbloc == 1)
    {
        cursor_x=4;cursor_y=1;
        textattr(atribut2);
        gotoxy(cursor_x,cursor_y);
        cprintf("%s",men_prin[0]);
        textattr(atribut3);
        gotoxy(cursor_x,cursor_y);
        cprintf("%s",car_prin[0]);
        aux_bloc=auxbloc;
    }
    if (auxbloc == 2)
    {
        cursor_x=14;cursor_y=1;
        textattr(atribut2);
        gotoxy(cursor_x,cursor_y);

```

```

cprintf("%s",men_prin[1]);
textattr(atribut3);
gotoxy(cursor_x,cursor_y);
cprintf("%s",car_prin[1]);
aux_bloc=auxbloc;
}
if (auxbloc == 3)
{
    cursor_x=22;cursor_y=1;
    textattr(atribut2);
    gotoxy(cursor_x,cursor_y);
    cprintf("%s",men_prin[2]);
    textattr(atribut3);
    gotoxy(cursor_x,cursor_y);
    cprintf("%s",car_prin[2]);
    aux_bloc=auxbloc;
}
i=0;
do
{
    tecla=fun_vali(tecla,teclprin,pri_mouse);
    if (tecla == 65 || tecla == 68 || tecla == 73)
    {
        aux_bloc=che_car(tecla,men_prin,car_prin,letrprin);
        tecla=13;
    }
    if (tecla == 205)
    {
        if (aux_bloc == 1) cursor_x=4;
        if (aux_bloc == 2) cursor_x=14;
        if (aux_bloc == 3) cursor_x=22;
        gotoxy(cursor_x,1);
        textcolor(WHITE);
        cprintf("%s",men_prin[aux_bloc-1]);
        textcolor(YELLOW);
        gotoxy(cursor_x,1);
        cprintf("%s",car_prin[aux_bloc-1]);
        if (aux_bloc == 1) algo=10;
        if (aux_bloc == 2) algo=8;
        if (aux_bloc == 3)
        {
            algo=-18;
            aux_bloc=0;
        }
        textattr(atribut2);
        gotoxy(cursor_x+algo,1);
        cprintf("%s",men_prin[aux_bloc]);
        textattr(atribut3);
        gotoxy(cursor_x+algo,1);
        cprintf("%s",car_prin[aux_bloc]);
        if (tecla != 13 && tecla != 27) aux_bloc += i;
    }
    if (tecla == 203)
    {

```

```

    if (aux_bloc == 1) cursor_x=4;
    if (aux_bloc == 2) cursor_x=14;
    if (aux_bloc == 3) cursor_x=22;
    gotoxy(cursor_x,1);
    textcolor(WHITE);
    cprintf("%s",men_prin[aux_bloc-1]);
    textcolor(YELLOW);
    gotoxy(cursor_x,1);
    cprintf("%s",car_prin[aux_bloc-1]);
    if (aux_bloc == 1)
    {
        algo=18;
        aux_bloc=4;
    }
    if (aux_bloc == 2) algo=-10;
    if (aux_bloc == 3) algo=-8;
    textattr(atribut2);
    gotoxy(cursor_x+algo,1);
    cprintf("%s",men_prin[aux_bloc-2]);
    textattr(atribut3);
    gotoxy(cursor_x+algo,1);
    cprintf("%s",car_prin[aux_bloc-2]);
    if (tecla != 13 && tecla != 27) aux_bloc -= 1;
}
} while (tecla != 13) ;
return(aux_bloc);
}

/** MENU: DE ARCHIVO Y DATOS **/
int menuvert (letre_ar, carac2, apunta1, apunta2, auxblocl)
int apunta1[];
int apunta2[];
char *letre_ar[];
char *carac2[];
int auxblocl;

{
    int i=0;int tecla;
    int aux_bloc=1;int algo=0;
    int cursor_x=3; int cursor_y=2;
    int entro=0;

    while ( letre_ar[i] != NULL )
    {
        textcolor(WHITE);
        gotoxy(cursor_x, cursor_y);
        cprintf("%s", letre_ar[i]);
        textcolor(YELLOW);
        gotoxy(cursor_x, cursor_y);
        cprintf("%s", carac2[i]);
        cursor_y += 1;
        i+=1;
    }
    if (auxblocl == 1)

```

```

{
  cursor_x=3;cursor_y=2;
  textattr(atribut2);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", letre_ar[0]);
  textattr(atribut5);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", carac2[0]);
  aux_bloc=auxbloc1;
}
if (auxbloc1 == 2)
{
  cursor_x=3;cursor_y=3;
  textattr(atribut2);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", letre_ar[1]);
  textattr(atribut5);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", carac2[1]);
  aux_bloc=auxbloc1;
}
if (auxbloc1 == 3)
{
  cursor_x=3;cursor_y=4;
  textattr(atribut2);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", letre_ar[2]);
  textattr(atribut5);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", carac2[2]);
  aux_bloc=auxbloc1;
}
if (auxbloc1 == 4)
{
  cursor_x=3;cursor_y=5;
  textattr(atribut2);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", letre_ar[3]);
  textattr(atribut5);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", carac2[3]);
  aux_bloc=auxbloc1;
}
if (auxbloc1 == 5)
{
  cursor_x=3;cursor_y=6;
  textattr(atribut2);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", letre_ar[4]);
  textattr(atribut5);
  gotoxy(cursor_x, cursor_y);
  cprintf("%s", carac2[4]);
  aux_bloc=auxbloc1;
}

```

```

if (auxbloc1 == 6)
{
    cursor_x=3;cursor_y=7;
    textattr(atribut2);
    gotoxy(cursor_x,cursor_y);
    cprintf("%s",letre_ar[5]);
    textattr(atribut5);
    gotoxy(cursor_x,cursor_y);
    cprintf("%s",carac2[5]);
    aux_bloc=auxbloc1;
}
i=0;
do
{
    if( *(carac2[0]) == 'P' )
        tecla=fun_vali(tecla,apunta1,sec_1_mouse);
    else tecla=fun_vali(tecla,apunta1,sec_mouse);
        if (tecla != ENTER && tecla != ESC && tecla != FLEABAJ
&& tecla != FLEARRI && tecla != FLEATRA && tecla != FLEADEL)
        {
            entro=1;
            aux_bloc=che_car2(tecla,letre_ar,carac2,apunta2);
            tecla=13;
        }

    if (tecla == FLEABAJ)
    {
        entro=1;
        if (aux_bloc == 1) cursor_y=2;
        if (aux_bloc == 2) cursor_y=3;
        if (aux_bloc == 3) cursor_y=4;
        if (aux_bloc == 4) cursor_y=5;
        if (aux_bloc == 5) cursor_y=6;
        if (aux_bloc == 6) cursor_y=7;
        gotoxy(3,cursor_y);
        textcolor(WHITE);
        cprintf("%s",letre_ar[aux_bloc-1]);
        textcolor(YELLOW);
        gotoxy(3,cursor_y);
        cprintf("%s",carac2[aux_bloc-1]);
        if (aux_bloc == 1) algo=1;
        if (aux_bloc == 2) algo=1;
        if (aux_bloc == 3) algo=1;
        if (aux_bloc == 4) algo=1;
        if (aux_bloc == 5) algo=1;
        if (aux_bloc == 6)
        {
            algo=0;
            cursor_y=2;
            aux_bloc=0;
        }
        textattr(atribut2);
        gotoxy(3,cursor_y+algo);
        cprintf("%s",letre_ar[aux_bloc]);
    }
}

```

```

    textattr(atribut5);
    gotoxy(3, cursor_y+algo);
    cprintf("%s", carac2[aux_bloc]);
    if (tecla != 13 && tecla != 27) aux_bloc += 1;
}
if (tecla == FLEARRI)
{
    entro=1;
    if (aux_bloc == 1) cursor_y=2;
    if (aux_bloc == 2) cursor_y=3;
    if (aux_bloc == 3) cursor_y=4;
    if (aux_bloc == 4) cursor_y=5;
    if (aux_bloc == 5) cursor_y=6;
    if (aux_bloc == 6) cursor_y=7;
    gotoxy(3, cursor_y);
    textcolor(WHITE);
    cprintf("%s", letre_ar[aux_bloc-1]);
    textcolor(YELLOW);
    gotoxy(3, cursor_y);
    cprintf("%s", carac2[aux_bloc-1]);
    if (aux_bloc == 1)
    {
        algo=-1;
        cursor_y=8;
        aux_bloc=7;
    }
    if (aux_bloc == 2) algo=-1;
    if (aux_bloc == 3) algo=-1;
    if (aux_bloc == 4) algo=-1;
    if (aux_bloc == 5) algo=-1;
    if (aux_bloc == 6) algo=-1;
    textattr(atribut2);
    gotoxy(3, cursor_y+algo);
    cprintf("%s", letre_ar[aux_bloc-2]);
    textattr(atribut5);
    gotoxy(3, cursor_y+algo);
    cprintf("%s", carac2[aux_bloc-2]);
    if (tecla != 13 && tecla != 27) aux_bloc -= 1;
}
} while (tecla != 27 && tecla != 13);
if ( entro == 0 && tecla == 27 ) aux_bloc--;
if ( tecla == 27 ) aux_bloc=0;
return(aux_bloc);
}

/** VALIDA LA TECLA PRESIONADA **/
int fun_vali(tecla, posi_act, aux_men1_mouse)
int tecla;
int posi_act [];
int aux_men1_mouse[];

{
    int error=0, i;
    tecla=1;

```



```

do
{
    .i=0;
    tecla=pres_1_tecla(tecla,aux_men1_mouse);
    tecla=toupper(tecla);
    while ( posi_act[i] != NULL )
    {
        if ( posi_act[i] == tecla )
        {
            error=tecla;
            break;
        }
        else
        {
            error=0;
        }
        i+=1;
    }
    } while (error != tecla);
return(tecla);
}

```

```

/** VALIDA LA TECLA SI ES UN CARACTER **/

```

```

int che_car(tecla,men_arch,car_arch,apunta)
int tecla;
int apunta[];
char *car_arch [];
char *men_arch [];

```

```

{
    int i=0;int j=0 ;
    int aux_bloc;
    while ( apunta[i] != NULL )
    {
        if ( tecla == apunta[i] )
        {
            textattr(atribut2);
            gotoxy(4+j,1);
            cprintf("%s",men_arch[i]);
            textattr(atribut3);
            gotoxy(4+j,1);
            cprintf("%s",car_arch[i]);
            aux_bloc=i+1;
        }
        else
        {
            gotoxy(4+j,1);
            textcolor(WHITE);
            cprintf("%s",men_arch[i]);
            textcolor(YELLOW);
            gotoxy(4+j,1);
            cprintf("%s",car_arch[i]);
        }
    }
}

```

```

        i+=1;
        j=10;
        if ( i == 2 ) j=18;
    }
    return(aux_bloc);
}

/** VALIDA LA TECLA SI ES UN CARACTER DE ARCHIVO Y DATOS **/
int che_car2(tecla,men_arch,car_arch,posi_act)
int tecla;
char *car_arch [];
char *men_arch [];
int posi_act [];

{
    int aux_bloc;
    int i=0;
    while ( posi_act[i] != NULL )
    {
        if (tecla == posi_act[i] )
        {
            textattr(atribut2);
            gotoxy(3,2+i);
            cprintf("%s",men_arch[i]);
            textattr(atribut5);
            gotoxy(3,2+i);
            cprintf("%s",car_arch[i]);
            aux_bloc=i+1;
        }
        else
        {
            gotoxy(3,2+i);
            textcolor(WHITE);
            cprintf("%s",men_arch[i]);
            textcolor(YELLOW);
            gotoxy(3,2+i);
            cprintf("%s",car_arch[i]);
        }
        i+=1;
    }
    return(aux_bloc);
}

```

PREPROCESADOR INCLUYE AL ARCHIVO: MOUSE.C

```

#include "libreria.h"
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <stat.h>
#include <dir.h>
#include <errno.h>

void mouse();
int pres_1_tecla(int tecla,int aux_men1_mouse[]);
int mouse1(int aux_men1_mouse[]);
int vali_1_click(int aux_men1_mouse[],int col_graf,
int reng_graf);
int pres_2_tecla(int tecla,char *aux_guarda[100],
int *apu_columna,int *apu_renglon,int tamven1_,int
tamven2_,int cord_archivos[]);
int mouse2(char *aux_guarda[100],int *apu_columna,
int *apu_renglon,int tamven1_,int tamven2_,
int cord_archivos[]);
int vali_2_click(char *aux_guarda[100],int *apu_columna,
int *apu_renglon,int tamven1_,int tamven2_,
int cord_archivos[],int col_graf,int reng_graf);
int pres_3_tecla(int teclaaa,int *apu_num_reng,
int *apu_renglon,int *apu_columna,struct ventana *vent,
int aux_top,int aux_tam_vent,double aux_t_x[],
double aux_t_y[],double aux_t_z[],double aux_t_a[],
double aux_t_b[],double aux_t_c[],double aux_t_d[],
double aux_t_e[],double aux_t_f_[],int cord_graf[],
int aux_lim_sup,int option);
int mouse3(int *apu_num_reng,int *apu_renglon,
int *apu_columna,struct ventana *vent,int aux_top,
int aux_tam_vent,double aux_t_x[],double aux_t_y[],
double aux_t_z[],double aux_t_a[],double aux_t_b[],
double aux_t_c[],double aux_t_d[],double aux_t_e[],
double aux_t_f_[],int cord_graf[],int aux_lim_sup,
int option);
int vali_3_click(int *apu_num_reng,int *apu_renglon,
int *apu_columna,struct ventana *vent,int aux_top,
int aux_tam_vent,double aux_t_x[],double aux_t_y[],
double aux_t_z[],double aux_t_a[],double aux_t_b[],
double aux_t_c[],double aux_t_d[],double aux_t_e[],
double aux_t_f_[],int cord_graf[],int col_graf,
int reng_graf,int aux_lim_sup,int option);

```

```

/** M O U S E **/

```

```

void mouse()
{
    union REGS rin,rout;

```

```

/**** INICIALIZAR RATON Y OBTENER SU ESTADO ****/
rin.x.ax = 0;
int86(0x33, &rin, &rout);
/** MOSTRAR EL CURSOR DEL RATON **/
rin.x.ax = 1;
int86(0x33, &rin, &rout);
/** MOSTRAMOS EL CURSOR CON SUS ATRIBUTOS AL LEER **/
rin.x.ax = 10;
rin.x.bx = 1;
rin.x.cx = 3;
rin.x.dx = 7;
int86(0x33, &rin, &rout);
}

/** PRESIONA UNA TECLA Y LA REGRESA MOUSE O KEYBOARD **/
int pres_1_tecla(tecla, aux_men1_mouse)
int tecla;
int aux_men1_mouse[];
{
    if( ( tecla=mousel(aux_men1_mouse) ) != 0 ) return(tecla);
    if( !(tecla = getch() ) )
        tecla = getch() | 128;
    return tecla & 255;
}

/** LECTURA DEL MOUSE1 **/
int mousel(aux_men1_mouse)
int aux_men1_mouse[];
{
    union REGS mousin, mousout;
    int columna_graf, renglon_graf;

    while(1)
    {
        if( kbhit() ) return(0);
        mousin.x.ax = 3 ;
        int86(0x33, &mousin, &mousout);
        if( mousout.x.bx == 1 )
        {
            columna_graf = mousout.x.cx;
            renglon_graf = mousout.x.dx;
            if( (columna_graf = vali_1_click(aux_men1_mouse,
                columna_graf, renglon_graf)) != 0 )
                break;
        }
        if( mousout.x.bx == 2 )
        {
            columna_graf=27;
            break;
        }
    }
    return(columna_graf);
}

```

```

/** VALIDA QUE EL CLIK ESTE EN UNA AREA VALIDA **/
int vali_1_click(aux_men1_mouse, col_graf, reng_graf)
int aux_men1_mouse[];
int col_graf, reng_graf;
{
    int i=0;
    while( *(aux_men1_mouse+i) != NULL )
    {
        if( (col_graf >= *(aux_men1_mouse+i) &&
(col_graf<=*(aux_men1_mouse+i+1)) ) && reng_graf ==
*(aux_men1_mouse+i+2) )
            return(*(aux_men1_mouse+i+3));
        else i=i+4;
    }
    return(0);
}

/** PRESIONA UNA TECLA PARA LOS ARCHIVOS **/
int pres_2_tecla(tecla, aux_guarda, apu_columna, apu_renglon,
tamven1_, tamven2_, cord_archivos)
int tecla;
char *aux_guarda[100];
int *apu_columna;
int *apu_renglon;
int tamven1_, tamven2_;
int cord_archivos[];

{
    if( ( tecla=mouse2(aux_guarda, apu_columna,
apu_renglon, tamven1_, tamven2_, cord_archivos) ) != 0 )
return(tecla);
    if( !(tecla = getch() ) )
        tecla = getch() | 128;
    return tecla & 255;
}

/** LECTURA DEL MOUSE2 **/
int mouse2(aux_guarda, apu_columna, apu_renglon, tamven1_,
tamven2_, cord_archivos)
char *aux_guarda[100];
int *apu_columna;
int *apu_renglon;
int tamven1_, tamven2_;
int cord_archivos[];
{
    union REGS mousin, mousout;
    int columna_graf, renglon_graf;

while(1)
{
    if( kbhit() ) return(0);
    mousin.x.ax = 3 ;
    int86(0x33, &mousin, &mousout);
}
}

```

```

if( mousout.x.bx == 1 )
{
    columna_graf = mousout.x.cx;
    renglon_graf = mousout.x.dx;
    if( (columna_graf = vali_2_click(aux_guarda,
        apu_columna,apu_renglon,tamven1,tamven2,
        cord_archivos,columna_graf,renglon_graf) ) != 0 );
        break;
}
}
return(columna_graf);
}
/** VALIDA QUE EL CLIK ESTE EN UNA AREA VALIDA **/
int vali_2_click(aux_guarda,apu_columna,apu_renglon,
    tamven1,tamven2,cord_archivos,col_graf,reng_graf)
char *aux_guarda[100];
int *apu_columna, *apu_renglon;
int tamven1,tamven2;
int cord_archivos[];
int col_graf,reng_graf;

{
    int i=0;
    int entra,col__temp;
    int num_reng;

    while( *(cord_archivos+i) != NULL )
    {
        if( (col_graf >= *(cord_archivos+i) &&
            (col_graf<=*(cord_archivos+i+1)) ) && reng_graf ==
            *(cord_archivos+i+2) )
        {
            if( (col_graf >= *(cord_archivos+i) &&
                (col_graf<=*(cord_archivos+i+1)) ) && 152 ==
                *(cord_archivos+i+2) )
                return( *(cord_archivos+i+5) );
            if( (col_graf >= *(cord_archivos+i) &&
                (col_graf<=*(cord_archivos+i+1)) ) && 152 ==
                *(cord_archivos+i+2) )
                return( *(cord_archivos+i+5) );
            if( (col_graf >= *(cord_archivos+i) &&
                (col_graf<=*(cord_archivos+i+1)) ) && 160 ==
                *(cord_archivos+i+2) )
                return( *(cord_archivos+i+5) );
            if( (col_graf >= *(cord_archivos+i) &&
                (col_graf<=*(cord_archivos+i+1)) ) && 160 ==
                *(cord_archivos+i+2) )
                return( *(cord_archivos+i+5) );
            entra=*(cord_archivos+i+4);
            while( entra != 0 )
            {
                if( *apu_columna == i )
                {
                    if( tamven1 + *apu_renglon - 1 == tamven2_ )

```

```

    {
        if( aux_guarda[tamven2_+1] == NULL )
        {
            entra--;
            continue;
        }
write_archivos(*apu_renglon,*apu_columna,aux_guarda,
                0,tamven1_);
        *apu_columna = 2;
        *apu_renglon = 1;
    }
    else
    {
        if(aux_guarda[tamven1_+ ( *apu_renglon + 1 ) - 1 ]
== NULL )
        {
            --entra;
            continue;
        }
write_archivos(*apu_renglon,*apu_columna,aux_guarda,
                0,tamven1_);
        ++(*apu_renglon);
    }
}
else
{
    if( tamven2_ + *apu_renglon == tamven2_ + 8 )
    {
        if ( aux_guarda[tamven2_ + ( *apu_renglon + 1 )]
== NULL ) continue;
        col_temp = *apu_columna;
        tamven1_ = tamven1_ + 8;
        *apu_renglon = 1;
        *apu_columna = 1;
        textattr(atribut12);
        clrscr();
        linea();
        textattr(atribut12);
        for(num_reng = 1 ; num_reng <= 8 ; num_reng++ )
        {
            if(aux_guarda[tamven1_ + *apu_renglon - 1 ]
== NULL) break;
write_archivos(*apu_renglon,*apu_columna,aux_guarda,
                0,tamven1_);
            ++(*apu_renglon);
        }
        tamven2_ = tamven2_ + 8;
        *apu_columna=2;
        *apu_renglon=1;
        for(num_reng = 1 ; num_reng <= 8 ; num_reng++ )
        {
            if(aux_guarda[tamven2_ + *apu_renglon] == NULL )
            break;
write_archivos(*apu_renglon,*apu_columna,aux_guarda,

```

```

        0,tamven2_);
        ++(*apu_renglon);
    }
    *apu_columna = col__temp;
    *apu_renglon = 1;
}
else
{
    if( aux_guarda[tamven2_ + *apu_renglon + 1 ] ==
        NULL ) continue;
    write_archivos(*apu_renglon,*apu_columna,aux_guarda,
        0,tamven2_);
    ++(*apu_renglon);
}
}
    entra--;
}
if(*(cord_archivos+i+3) == 1)
{
    write_archivos(*(cord_archivos+i+4),*(cord_archivos+i+3),
        aux_guarda,1,tamven1_);
    *apu_columna=*(cord_archivos+i+3);
    *apu_renglon=*(cord_archivos+i+4);
}
else
{
    write_archivos(*(cord_archivos+i+4),*(cord_archivos+i+3),
        aux_guarda,1,tamven2_);
    *apu_columna=*(cord_archivos+i+3);
    *apu_renglon=*(cord_archivos+i+4);
}
return( *(cord_archivos+i+5) );
}
else
    i=i+6;
}
return(0);
}

```

```

/** PRESIONA UNA TECLA PARA LOS ARCHIVOS **/
int pres_3_tecla(tecla,apu_num_reng,apu_renglon,
apu_columna,vent,aux_top,aux_tam_vent,aux_t_x,aux_t_y,
aux_t_z,aux_t_a,aux_t_b,aux_t_c,aux_t_d,aux_t_e,aux_t_f,
cord_graf,aux_lim_sup,option)
int tecla;
int *apu_num_reng,*apu_renglon,*apu_columna;
struct ventana *vent;
int aux_top;
int aux_tam_vent;
double aux_t_x[],aux_t_y[],aux_t_z[],aux_t_a[],aux_t_b[],
aux_t_c[],aux_t_d[];
double aux_t_e[],aux_t_f[];
int cord_graf[];
int aux_lim_sup;

```



```

int option;
{
    if( ( tecla=mouse3(apu_num_reng,apu_renglon,apu_columna,
                      vent,aux_top,aux_tam_vent,aux_t_x,
                      aux_t_y,aux_t_z,aux_t_a,aux_t_b,aux_t_c,
                      aux_t_d,aux_t_e,aux_t_f,cord_graf,aux_lim_sup,
                      option) ) != 0 ) return(tecla);
    if( !(tecla = getch() ) )
        tecla = getch() | 128;
    return tecla & 255;
}

/** LECTURA DEL MOUSE3 **/
int mouse3(apu_num_reng,apu_renglon,apu_columna,vent,
aux_top,aux_tam_vent,aux_t_x,aux_t_y,aux_t_z,aux_t_a,
aux_t_b,aux_t_c,aux_t_d,aux_t_e,aux_t_f,cord_graf,
aux_lim_sup,option)
int *apu_num_reng,*apu_renglon,*apu_columna;
struct ventana *vent;
int aux_top;
int aux_tam_vent;
double aux_t_x[],aux_t_y[],aux_t_z[],aux_t_a[],aux_t_b[],
aux_t_c[],aux_t_d[];
double aux_t_e[],aux_t_f[];
int cord_graf[];
int aux_lim_sup;
int option;
{
    union REGS mousin, mousout;
    int columna_graf, renglon_graf;

    while(1)
    {
        if( kbhit() ) return(0);
        mousin.x.ax = 3 ;
        int86(0x33,&mousin,&mousout);
        if( mousout.x.bx == 1 )
        {
            columna_graf = mousout.x.cx;
            renglon_graf = mousout.x.dx;
            if( (columna_graf = vali_3_click(apu_num_reng,
                                           apu_renglon,apu_columna,
                                           vent,aux_top,aux_tam_vent,
                                           aux_t_x,aux_t_y,aux_t_z,aux_t_a,
                                           aux_t_b,aux_t_c,aux_t_d,aux_t_e,
                                           aux_t_f,cord_graf,columna_graf,
                                           renglon_graf,aux_lim_sup,option) ) != 0 )
                break;
        }
    }
    return(columna_graf);
}

```

```

/** VALIDA EL CLICK DEL MOSE **/
int vali_3_click(apu_num_reng,apu_renglon,apu_columna,vent,
aux_top,aux_tam_vent,aux_t_x,aux_t_y,aux_t_z,aux_t_a,
aux_t_b,aux_t_c,aux_t_d,aux_t_e,aux_t_f, cord_graf, col_graf, reng_graf, aux_lim_sup, option)
int *apu_num_reng, *apu_renglon, *apu_columna;
struct ventana *vent;
int aux_top;
int aux_tam_vent;
double aux_t_x[],aux_t_y[],aux_t_z[],aux_t_a[],aux_t_b[],
aux_t_c[],aux_t_d[];
double aux_t_e[],aux_t_f[];
int cord_graf[];
int col_graf, reng_graf;
int aux_lim_sup;
int option;
{
    int i=0;
    int temp_reng;
        while( *(cord_graf+i) != NULL )
        {
            if( (col_graf >= *(cord_graf+i) &&
(col_graf<=*(cord_graf+i+1)) ) && reng_graf ==
*(cord_graf+i+2) )
            {
                if( option == 1 )
                {
                    if( (col_graf >= 176 && (col_graf<=224) ) &&
reng_graf == 144 ) return(201);
                    if( (col_graf >= 240 && (col_graf<=288) ) &&
reng_graf == 144 ) return(209);
                }
                if( option == 2 )
                {
                    if( (col_graf >= 40 && (col_graf<=264) ) &&
reng_graf == 152 ) return(201);
                    if( (col_graf >= 280 && (col_graf<=504) ) &&
reng_graf == 152 ) return(209);
                }
                if( option == 3 )
                {
                    if( (col_graf >= 192 && (col_graf<=240) ) &&
reng_graf == 160 ) return(201);
                    if( (col_graf >= 256 && (col_graf<=304) ) &&
reng_graf == 160 ) return(209);
                }
                if( option == 4 )
                {
                    if( (col_graf >= 134 && (col_graf<=280) ) &&
reng_graf == 168 ) return(201);
                    if( (col_graf >= 296 && (col_graf<=392) ) &&
reng_graf == 168 ) return(209);
                }
                if( option == 5 )

```

```

    {
        if( (col_graf >= 184 && (col_graf<=232) ) &&
            reng_graf == 176 ) return(201);
        if( (col_graf >= 248 && (col_graf<=296) ) &&
            reng_graf == 176 ) return(209);
    }
    if( option == 6 )
    {
        if( (col_graf >= 184 && (col_graf<=232) ) &&
            reng_graf == 184 ) return(201);
        if( (col_graf >= 248 && (col_graf<=296) ) &&
            reng_graf == 184 ) return(209);
    }
    temp_reng=4;
    for(*apu_num_reng =aux_top ; *apu_num_reng < aux_top +
        aux_tam_vent ; (*apu_num_reng)++ )
    {
        escribe_linea_ceros(temp_reng,*apu_num_reng,*apu_columna,
            vent,aux_t_x,aux_t_y,aux_t_z,
            aux_t_a,aux_t_b,aux_t_c,aux_t_d,
            aux_t_e,aux_t_f,0);
        ++(temp_reng);
    }
    *apu_num_reng= *(cord_graf+i+4) - aux_lim_sup +
aux_top;
    *apu_renglon=*(cord_graf+i+4);
    *apu_columna=*(cord_graf+i+5);
    escribe_linea_ceros(*(cord_graf+i+4),*apu_num_reng,
        *(cord_graf+i+5),vent,aux_t_x,aux_t_y,aux_t_z,
        aux_t_a,aux_t_b,aux_t_c,aux_t_d,aux_t_e,aux_t_f,1);
    return(13);
}
else
    i=i+7;
}
return(0);
}

```

PREPROCESADOR INCLUYE AL ARCHIVO: PRESENTA.C

```

#include "libreria.h"
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <stat.h>
#include <dir.h>
#include <errno.h>

void cursor();
void fondo(int cursor_x,int cursor_y);
void marco(int cursor_x,int cursor_y);
void presenta();
void sombra(int cursor_x,int cursor_y);
void sombra_2(int cursor_x,int cursor_y);
void abre_ventana(int cursor_x,int cursor_y,int cursor_z,int
cursor_w);

/** CURSOR **/
void cursor()
{
    union REGS rin, rout;

    rin.h.ah = 1;
    rin.h.ch = 0x20;
    int86(0x10, &rin, &rout);
}
/** FONDO **/
void fondo(cursor_x,cursor_y)
int cursor_x,cursor_y;
{
    int j,i;
    int inicio=2;
    clrscr();
    if ( cursor_x != 80 )
    {
        textattr(atribut4);
        inicio=1;
    }
    gotoxy(1, inicio);
    printf("%c", ESQSUIZQ);
    gotoxy(cursor_x, inicio);
    printf("%c", ESQSUDER);
    gotoxy(1, cursor_y);
    printf("%c", ESQINIZQ);
    gotoxy(cursor_x, cursor_y);
    printf("%c", ESQINDER);
    if ( cursor_x != 80 ) inicio=1;
    else inicio=2;
    for (j=2; j<=(cursor_x-1); j++)

```

```

    {
        gotoxy(j, inicio);
        printf("%c", LADOARAB);
        gotoxy(j, cursor_y);
        printf("%c", LADOARAB);
    }
    if ( cursor_x != 80 ) inicio=2;
    else inicio=3;
    for (j=inicio; j<=(cursor_y-1); j++)
    {
        gotoxy(1, j);
        printf("%c", LADOIZDE);
        gotoxy(cursor_x, j);
        printf("%c", LADOIZDE);
    }
    if ( cursor_x == 80)
    {
        for (i=3; i<=23; i++)
        {
            for (j=2; j<=79; j++)
            {
                gotoxy(j, i);
                printf("%c", TABIQUE);
            }
        }
    }
}

```

```

/** SOMBRA SENCILLA **/

```

```

void sombra(cursor_x, cursor_y)
int cursor_x, cursor_y;

```

```

{
    int i;

    highvideo();
    textattr(atribut1);
    gotoxy(1, cursor_y+1);
    cprintf("%c", TABIQUE);
    for (i=2; i<=cursor_x; i++)
    {
        textattr(atribut7);
        gotoxy(i, cursor_y+1);
        printf("%c", TABIQUE);
    }
    textattr(atribut1);
    gotoxy(cursor_x+1, 1);
    cprintf("%c", TABIQUE);
    for (i=2; i<=cursor_y+1; i++)
    {
        textattr(atribut7);
        gotoxy(cursor_x+1, i);
        printf("%c", TABIQUE);
    }
}

```

```

}

/** DOBLE SOMBRA **/
void sombra_2(cursor_x, cursor_y)
int cursor_x, cursor_y;

{
    int i;
    highvideo();
    textattr(atribut1);
    gotoxy(1, cursor_y+1);
    cprintf("%c", TABIQUE);
    gotoxy(1, cursor_y+2);
    cprintf("%c", TABIQUE);
    for (i=2; i<=cursor_x; i++)
    {
        textattr(atribut7);
        gotoxy(i, cursor_y+1);
        printf("%c", TABIQUE);
    }
    textattr(atribut1);
    gotoxy(cursor_x+1, 1);
    cprintf("%c", TABIQUE);
    gotoxy(cursor_x+2, 1);
    cprintf("%c", TABIQUE);
    for (i=2; i<=cursor_y+1; i++)
    {
        textattr(atribut7);
        gotoxy(cursor_x+1, i);
        printf("%c", TABIQUE);
        gotoxy(cursor_x+2, i);
        printf("%c", TABIQUE);
    }
}

/** ABRE VENTANA **/
void abre_ventana( cursor_x, cursor_y, cursor_z, cursor_w)
int cursor_x, cursor_y, cursor_z, cursor_w;
{
    int tecla0;

    window(cursor_x, cursor_y, cursor_z, cursor_w);
    textattr(atribut4);
    clrscr();
    fondo( (cursor_z - cursor_x-1), (cursor_w - cursor_y) );
    sombra_2((cursor_z - cursor_x-1), (cursor_w - cursor_y));
    textattr(atribut4);
}

/** PANTALLAS DE PRESENTACION **/
void presenta()
{
    char stop0[51*3*2]; char stop1[53*5*2];
    char stop2[31*7*2]; char stop3[33*9*2];
}

```

```

char stop4[35*11*2];char stop5[37*13*2];
char stop6[39*15*2];char stop7[41*17*2];
char stop8[43*19*2];char stop9[45*21*2];
int i,j;
struct text_info ti;
int letre_presenta[125];

```

```

letre_presenta[0]='U';letre_presenta[1]='N';
letre_presenta[2]='I';letre_presenta[3]='V';
letre_presenta[4]='E';letre_presenta[5]='R';
letre_presenta[6]='S';letre_presenta[7]='I';
letre_presenta[8]='D';letre_presenta[9]='A';
letre_presenta[10]='D';letre_presenta[11]='';
letre_presenta[12]='N';letre_presenta[13]='A';
letre_presenta[14]='C';letre_presenta[15]='I';
letre_presenta[16]='O';letre_presenta[17]='N';
letre_presenta[18]='A';letre_presenta[19]='L';
letre_presenta[21]=' ';letre_presenta[22]='A';
letre_presenta[23]='U';letre_presenta[24]='T';
letre_presenta[25]='O';letre_presenta[26]='N';
letre_presenta[27]='O';letre_presenta[28]='M';
letre_presenta[29]='A';letre_presenta[30]='';
letre_presenta[31]='D';letre_presenta[32]='E';
letre_presenta[33]=' ';letre_presenta[34]='M';
letre_presenta[35]='E';letre_presenta[36]='X';
letre_presenta[37]='I';letre_presenta[38]='C';
letre_presenta[39]='O';letre_presenta[40]='';
letre_presenta[41]='E';letre_presenta[42]='.';
letre_presenta[43]='N';letre_presenta[44]='.';
letre_presenta[45]='E';letre_presenta[46]='.';
letre_presenta[47]='P';letre_presenta[48]='.';
letre_presenta[49]=' ';letre_presenta[50]='-';
letre_presenta[51]=' ';letre_presenta[52]='A';
letre_presenta[53]='C';letre_presenta[54]='A';
letre_presenta[55]='T';letre_presenta[56]='L';
letre_presenta[57]='A';letre_presenta[58]='N';
letre_presenta[59]='P';letre_presenta[60]='';
letre_presenta[61]='R';letre_presenta[62]='';
letre_presenta[63]='E';letre_presenta[64]='';
letre_presenta[65]='P';letre_presenta[66]='';
letre_presenta[67]='R';letre_presenta[68]='.';
letre_presenta[69]='O';letre_presenta[70]='';
letre_presenta[71]='C';letre_presenta[72]='';
letre_presenta[73]='E';letre_presenta[74]='';
letre_presenta[75]='S';letre_presenta[76]='';
letre_presenta[77]='A';letre_presenta[78]='.';
letre_presenta[79]='D';letre_presenta[80]='';
letre_presenta[81]='O';letre_presenta[82]='';
letre_presenta[83]='R';letre_presenta[88]='P';
letre_presenta[89]='R';letre_presenta[90]='O';
letre_presenta[91]='G';letre_presenta[92]='R';
letre_presenta[93]='A';letre_presenta[94]='M';
letre_presenta[95]='A';letre_presenta[96]='D';
letre_presenta[97]='O';letre_presenta[98]='R';

```

```

letre_presenta[99]=': '; letre_presenta[100]='S';
letre_presenta[101]='E'; letre_presenta[102]='R';
letre_presenta[103]='G'; letre_presenta[104]='I';
letre_presenta[105]='O'; letre_presenta[106]=' ';
letre_presenta[107]='G'; letre_presenta[108]='O';
letre_presenta[109]='N'; letre_presenta[110]='Z';
letre_presenta[111]='A'; letre_presenta[112]='L';
letre_presenta[113]='E'; letre_presenta[114]='Z';
letre_presenta[115]=' '; letre_presenta[116]='C';
letre_presenta[117]='A'; letre_presenta[118]='S';
letre_presenta[119]='T'; letre_presenta[120]='A';
letre_presenta[121]='Ñ'; letre_presenta[122]='E';
letre_presenta[123]='D'; letre_presenta[124]='A';
textattr( atribut2 );
clrscr();
gettextinfo(&ti);
for( i=0 ; i<=9 ; i++ )
{
    textattr( atribut3 );
    window(39-i-24,12-i,41+i+24,14+i);
    clrscr();
    delay(150);
    if( i == 0 ) gettext(15,12,65,14,stop0);
    if( i == 1 ) gettext(14,11,66,15,stop1);
    if( i == 2 ) gettext(13,10,43,16,stop2);
    if( i == 3 ) gettext(12,9,44,17,stop3);
    if( i == 4 ) gettext(11,8,45,18,stop4);
    if( i == 5 ) gettext(10,7,46,19,stop5);
    if( i == 6 ) gettext(9,6,47,20,stop6);
    if( i == 7 ) gettext(8,5,48,21,stop7);
    if( i == 8 ) gettext(7,4,49,22,stop8);
    if( i == 9 ) gettext(6,3,74,23,stop9);
}
marco(74,23);
window(6,3,74,23);
j=4;
for(i=0;i<=29;i++)
{
    if(i==20) continue;
    textcolor( WHITE );
    gotoxy(i+j,3);
    cprintf(" %c ", letre_presenta[i]);
    delay(50);
    j++;
}
textcolor( WHITE );
gotoxy(30,5);
cprintf(" %c ", letre_presenta[31]);
delay(50);
textcolor( WHITE );
gotoxy(32,5);
cprintf(" %c ", letre_presenta[32]);
delay(50);
j=4;

```



```

for(i=34; i<=39; i++)
{
    textcolor( WHITE );
    gotoxy(i-12+j,7);
    cprintf(" %c ",letra_presenta[i]);
    delay(50);
    j++;
}
j=9;
for(i=41; i<=58; i++)
{
    textcolor( WHITE );
    gotoxy(i-36+j,9);
    cprintf(" %c ",letra_presenta[i]);
    delay(50);
    j++;
}
j=9;
for(i=59; i<=83; i++)
{
    textcolor( LIGHTBLUE );
    gotoxy(i-59+j,12);
    cprintf(" %c ",letra_presenta[i]);
    delay(50);
    j++;
}
j=7;
for(i=88; i<=99; i++)
{
    textcolor( LIGHTCYAN );
    gotoxy(i-88+j,16);
    cprintf(" %c ",letra_presenta[i]);
    delay(50);
    j++;
}
j=16;
for(i=100; i<=124; i++)
{
    textcolor( WHITE );
    gotoxy(i-100+j,18);
    cprintf("%c",letra_presenta[i]);
    delay(50);
    j++;
}
lowvideo();
textattr(atribut6);
gotoxy(35,21);
cprintf("... ESPERE UN          ...");
lowvideo();
textcolor( WHITE | BLINK );
gotoxy(54,21);
cprintf("MOMENTO");
delay(5000);
i=9;

```



```

    cprintf("      █ █ █ █ █ █ █ █ █ █");
    gotoxy(4,8);
    cprintf("      █ █ █ █ █ █ █ █ █ █");
    gotoxy(4,9);
    cprintf("      █ █ █ █ █ █ █ █ █ █");
    lowvideo();
    gotoxy(3,13);
    textcolor( LIGHTBLUE );
    cprintf("S I G L A S : ");
    gotoxy(3,15);
    textcolor( LIGHTGREEN );
    cprintf(" A N ");
    textcolor( LIGHTGRAY );
    gotoxy(12,15);
    cprintf(".- Análisis ");
    textcolor( LIGHTGREEN );
    gotoxy(3,16);
    cprintf(" E ");
    textcolor( LIGHTGRAY );
    gotoxy(12,16);
    cprintf(".- Estático por el método del");
    textcolor( LIGHTGREEN );
    gotoxy(3,17);
    cprintf(" ELEFIN ");
    textcolor( LIGHTGRAY );
    gotoxy(12,17);
    cprintf(".- Elemento Finito de 8 puntos. ");
    textattr(atribut6);
    gotoxy(35,21);
    cprintf("... ESPERE UN ...");
    lowvideo();
    textcolor( WHITE | BLINK );
    gotoxy(54,21);
    cprintf("MOMENTO");
    delay(5000);
    i=9;
    textattr(atribut2);
    window(1,1,80,24);
    clrscr();
    while(i>=0 )
    {
        if ( i == 0 ) puttext(39-i-24,12-i,41+i+24,14+i,stop0);
        if ( i == 1 ) puttext(39-i-24,12-i,41+i+24,14+i,stop1);
        if ( i == 2 ) puttext(39-i-24,12-i,41+i+24,14+i,stop2);
        if ( i == 3 ) puttext(39-i-24,12-i,41+i+24,14+i,stop3);
        if ( i == 4 ) puttext(39-i-24,12-i,41+i+24,14+i,stop4);
        if ( i == 5 ) puttext(39-i-24,12-i,41+i+24,14+i,stop5);
        if ( i == 6 ) puttext(39-i-24,12-i,41+i+24,14+i,stop6);
        if ( i == 7 ) puttext(39-i-24,12-i,41+i+24,14+i,stop7);
        if ( i == 8 ) puttext(39-i-24,12-i,41+i+24,14+i,stop8);
        if ( i == 9 ) puttext(39-i-24,12-i,41+i+24,14+i,stop9);
    }

```

```

        textattr(atribut2);
        window(1,1,80,24);
        delay(200);
        clrscr();
        --i;
    }
    textattr(atribut1);
    window(1,1,80,25);
    clrscr();
}

/** MARCO Y SOMBRA DE LAS PANTALLAS DE PRESENTACION **/
void marco(cursor_x, cursor_y)
int cursor_x, cursor_y;
{
    int j, i;
    int inicio=3;
    clrscr();
    textattr(atribut18);
    window(1,1,80,24);
    textattr(atribut19);
    gotoxy(6, cursor_y);
    cprintf("%c", ESPACIO);
    gotoxy(7, cursor_y);
    cprintf("%c", ESPACIO);
    textattr(atribut7);
    for (i=8; i<=cursor_x-2; i++)
    {
        gotoxy(i, cursor_y);
        cprintf("%c", ESPACIO);
    }
    textattr(atribut19);
    gotoxy(cursor_x-1, 3);
    cprintf("%c", ESPACIO);
    gotoxy(cursor_x, 3);
    cprintf("%c", ESPACIO);
    textattr(atribut7);
    for (i=4; i<=cursor_y; i++)
    {
        gotoxy(cursor_x-1, i);
        cprintf("%c", ESPACIO);
        gotoxy(cursor_x, i);
        cprintf("%c", ESPACIO);
    }
    textattr(atribut18);
    gotoxy(6, inicio);
    cprintf("%c", ESQSUIZQ);
    gotoxy(cursor_x-2, inicio);
    cprintf("%c", ESQSUDER);
    gotoxy(6, cursor_y-1);
    cprintf("%c", ESQINIZQ);
    gotoxy(cursor_x-2, cursor_y-1);
    cprintf("%c", ESQINDER);
    for (j=7; j<=(cursor_x-3); j++)

```

```
{
    gotoxy(j, inicio);
    cprintf("%c", LADOARAB);
    gotoxy(j, cursor_y-1);
    cprintf("%c", LADOARAB);
}
inicio=4;
for (j=inicio; j<=(cursor_y-2); j++)
{
    gotoxy(6, j);
    cprintf("%c", LADOIZDE);
    gotoxy(cursor_x-2, j);
    cprintf("%c", LADOIZDE);
}
}
```

PREPROCESADOR INCLUYE AL ARCHIVO: ARCHIVO.C

```

#include "libreria.h"
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <stat.h>
#include <dir.h>
#include <errno.h>

FILE *formato_archivo;
FILE *consulta;
extern int dato_ini[6] ;

void nuevo();
void current_directori(char aux_path[80]);
void abrir_archivo();
void abre_ventana(int cursor_x,int cursor_y,
int cursor_z,int cursor_w);
void ventana_archivos(int size,char *guarda[100],
int cord_graf_archivos[]);
void linea();
void write_archivos(int reng_act,int posi_columna,
char *aux_guards[100],int aux_pasada,int aux_tam_vent);
int columna_archivos(int posi_columna );
void traer_datos_arreglo(int fp, int bytes,char *arreglo);
void guardar();
void salvar_datos_arreglo(int fp, int bytes,char *arreglo);
void archi_extra(int dato_ini[]);
void arre_archi_extra(int size,struct ventana *vent,
double t_x[],double t_y[],double t_z[], double t_a[],
double t_b[],double t_c[], double t_d[], double t_e[],
double t_f[],int bloque);
void lin_archi_extra(int reng_act,int columna,
struct ventana *vent,double t_x[],double t_y[],
double t_z[], double t_a[], double t_b[],double t_c[], double
t_d[], double t_e[], double t_f[], int size,
int aux_bloque);
void consulta_archivo(int dato_ini[]);
void archivo_escritura(int size,struct ventana *vent,
double t_x[],double t_y[],double t_z[], double t_a[],
double t_b[],double t_c[], double t_d[], double t_e[], double
t_f[]);
void linea_escritura(int columna, struct ventana *vent,
double t_x[],double t_y[],double t_z[], double t_a[], double
t_b[], double t_c[], double t_d[], double t_e[], double
t_f[]);
char *formato(struct ventana *vent,int posi_columna);
void dir();
void borrar();

```



```

clrscr();
cad_temp[0]='\0';
aux_path[0]='\0';
path_archivo[0]='\0';
current_directori(aux_path);
strcpy(cad_temp,aux_path);
gotoxy(1,1);
printf("%s",aux_path);
gotoxy(1,1);
mousin.x.ax = 10;
mousin.x.bx = 1;
mousin.x.cx = 3;
mousin.x.dx = 7;
int86(0x33,&mousin,&mousout);
gets(aux_path);
rin.h.ah = 1;
rin.h.ch = 0x20;
int86(0x10,&rin,&rout);
mousin.x.ax = 10;
mousin.x.bx = 1;
mousin.x.cx = 3;
mousin.x.dx = 7;
int86(0x33,&mousin,&mousout);
if( strlen(aux_path) == 0 ) strcpy(aux_path,cad_temp);
if( chdir(aux_path) == -1 )
{
    textattr(atribut4);
    window(12,11,51,20);
    clrscr();
    gotoxy(1,2);
    printf("          ■ E R R O R ■ ");
    gotoxy(1,4);
    printf("SUBDIRECTORIO : ");
    gotoxy(1,5);
    printf("» %s ",aux_path);
    gotoxy(1,6);
    printf("INEXISTENTE   VERIFICAR ");
    gotoxy(1,7);
    printf("          DRIVE:\SUBDIRECTORIO... ");
    gotoxy(1,8);
    printf("NOTA: Presione cualquier tecla ");
    gotoxy(1,9);
    printf("          para continuar.");
    if (!getch())
        delay(2000);
    return;
}
strcpy(path_archivo,aux_path);
clrscr();
gotoxy(1,1);
printf("%s",aux_path);
gotoxy((strlen(aux_path)+1),1);
if( strlen(aux_path) == 3 ) printf("*.DAT");
if( strlen(aux_path) > 3 ) printf("\\*.DAT");

```



```

if( strlen(aux_path) == 3 ) strcat(aux_path,"*.DAT",6);
else
if( strlen(aux_path) > 3 )
    strcat(aux_path,"\\*.DAT",7);
done = findfirst( aux_path , &ffblk,0);
if( done == -1 )
{
    textattr(atribut12);
    window(12,11,40,18);
    clrscr();
    gotoxy(7,4);
    printf("No hay archivos");
    delay(2000);
    return;
}
i=0;
while(!done)
{
    guarda[i] = malloc(14);
    strcpy(guarda[i],ffblk.ff_name);
    done = findnext(&ffblk);
    i=i+1;
}
size = i;
textattr(atribut12);
window(12,11,40,18);
clrscr();
ventana_archivos(size,guarda,coord_archivos);
if( strlen(path_archivo) > 3 )
    strcat(path_archivo,"\\");
strncat(path_archivo,nombre_archivo,
        strlen(nombre_archivo));
if( ( fp = open( path_archivo , O_RDWR|O_CREAT|O_BINARY,
S_IREAD|S_IWRITE) ) == -1 )
{
    abre_ventana(22,13,60,19);
    textattr(atribut10);
    gotoxy(2,2);
    cprintf("                A V I S O :                ");
    gotoxy(2,4);
    cprintf(" ERROR EN LA APERTURA DE ARCHIVO " );
    gotoxy(2,5);
    cprintf(" ... OPCION INVALIDADA ... " );
    delay(2000);
    return;
}
if( read(fp, dato_ini , sizeof(dato_ini) ) !=
sizeof(dato_ini))
{
    abre_ventana(22,13,60,19);
    textattr(atribut4);
    gotoxy(2,2);
    cprintf("                A V I S O :                ");
    gotoxy(2,4);

```

```

    cprintf(" ERROR EN LA LECTURA DE BYTES " );
    gotoxy(2,5);
    cprintf(" .... OPCION INVALIDADA ...");
    delay(2000);
    return;
}

traer_datos_arreglo(fp, sizeof(p_n_x_1), (char *)p_n_x_1 );
traer_datos_arreglo(fp, sizeof(p_n_y_1), (char *)p_n_y_1 );
traer_datos_arreglo(fp, sizeof(p_n_x_2), (char *)p_n_x_2 );
traer_datos_arreglo(fp, sizeof(p_n_y_2), (char *)p_n_y_2 );
traer_datos_arreglo(fp, sizeof(p_n_z_2), (char *)p_n_z_2 );
traer_datos_arreglo(fp, sizeof(p_n_a_2), (char *)p_n_a_2 );
traer_datos_arreglo(fp, sizeof(p_n_b_2), (char *)p_n_b_2 );
traer_datos_arreglo(fp, sizeof(p_n_c_2), (char *)p_n_c_2 );
traer_datos_arreglo(fp, sizeof(p_n_d_2), (char *)p_n_d_2 );
traer_datos_arreglo(fp, sizeof(p_n_e_2), (char *)p_n_e_2 );
traer_datos_arreglo(fp, sizeof(p_n_f_2), (char *)p_n_f_2 );
traer_datos_arreglo(fp, sizeof(p_n_x_3), (char *)p_n_x_3 );
traer_datos_arreglo(fp, sizeof(p_n_y_3), (char *)p_n_y_3 );
traer_datos_arreglo(fp, sizeof(p_n_x_4), (char *)p_n_x_4 );
traer_datos_arreglo(fp, sizeof(p_n_y_4), (char *)p_n_y_4 );
traer_datos_arreglo(fp, sizeof(p_n_z_4), (char *)p_n_z_4 );
traer_datos_arreglo(fp, sizeof(p_n_x_5), (char *)p_n_x_5 );
traer_datos_arreglo(fp, sizeof(p_n_x_6), (char *)p_n_x_6 );
traer_datos_arreglo(fp, sizeof(p_n_y_6), (char *)p_n_y_6 );
close(fp);
}

/** DIRECTORIO ACTUAL **/
void current_directori(aux_path)
char aux_path[80];
{
    strcpy(aux_path, "X:\\");
    aux_path[0] = 'A' + getdisk();
    getcurdir(0, aux_path+3);
}

/**          ESTABLECEMOS LOS LIMITES DE LA VENTANA
                DE ARCHIVOS          **/
void ventana_archivos(aux_size, aux_guarda,
cord_graf_archivos)
int aux_size;
char *aux_guarda[100];
int cord_graf_archivos[];
{
    int teclaa=0;
    int col_temp;
    int col_reng;
    int reng_temp;
    int c;
    int *apu_columna=&posi_columna;
    int *apu_renglon=&reng_act;
    num_reng = 1;
    posi_columna = 1;

```



```

        -- posi_columna;
    }
}
if(teclaa == FLEABAJ )
{
    if( posi_columna == 1 )
    {
        if( tamven_1+reng_act-1 == tamven_2 )
        {
            if( aux_guarda[tamven_2+1] == NULL )
                continue;
            write_archivos(reng_act, posi_columna, aux_guarda,
                           0, tamven_1);
            posi_columna = 2;
            reng_act = 1;
        }
    }
}
else
{
    if(aux_guarda[tamven_1+ ( reng_act + 1 ) - 1 ] ==
NULL ) continue;
    write_archivos(reng_act, posi_columna, aux_guarda,
                   0, tamven_1);
    ++reng_act;
}
}
else
{
    if( tamven_2 + reng_act == tamven_2 + 8 )
    {
        if ( aux_guarda[tamven_2 + ( reng_act + 1 )] == NULL
) continue;
        col_temp = posi_columna;
        tamven_1 = tamven_1 + 8;
        reng_act = 1;
        posi_columna = 1;
        textattr(atribut12);
        clrscr();
        linea();
        textattr(atribut13);
        for(num_reng = 1 ; num_reng <= 8 ; num_reng++)
        {
            if(aux_guarda[tamven_1+reng_act-1] == NULL)
                break;
            write_archivos(reng_act, posi_columna, aux_guarda,
                           0, tamven_1);
            ++reng_act;
        }
        tamven_2 = tamven_2 + 8;
        posi_columna=2;
        reng_act=1;
        for(num_reng = 1 ; num_reng <= 8 ; num_reng++)
        {

```

```

        if( aux_guarda[tamven_2+reng_act] == NULL )
            break;
        write_archivos(reng_act, posi_columna, aux_guarda,
            0, tamven_2);
        ++reng_act;
    }
    posi_columna = col_temp;
    reng_act = 1;
}
else
{
    if( aux_guarda[tamven_2 + reng_act+1 ] == NULL )
continue;
    write_archivos(reng_act, posi_columna, aux_guarda,
        0, tamven_2);
    ++reng_act;
}
}
if( teclaa == FLEARRI )
{
    if( posi_columna == 2 )
    {
        if( tamven_2+reng_act == tamven_2 + 1 )
        {
            if( aux_guarda[tamven_2+reng_act-1] == NULL )
                continue;
            write_archivos(reng_act, posi_columna, aux_guarda,
                0, tamven_2);
            posi_columna = 1;
            reng_act = 8;
        }
        else
        {
            if( aux_guarda[tamven_2+reng_act -1] == NULL )
                continue;
            write_archivos(reng_act, posi_columna, aux_guarda,
                0, tamven_2);
            --reng_act;
        }
    }
}
else
{
    if( tamven_1+reng_act-1 == tamven_1 )
    {
        if( tamven_1 == 0 ) continue;
        if( aux_guarda[tamven_1+(reng_act-1)-1] == NULL )
            continue;
        col_temp = posi_columna;
        tamven_1 = tamven_1 - 8;
        reng_act = 1;
        posi_columna = 1;
        textattr(atribut12);
        clrscr();
    }
}
}

```

```

    linea();
    textattr(atribut12);
    for(num_reng = 1 ; num_reng <= 8 ; num_reng++)
    {
        if(aux_guarda[tamven_1+reng_act-1] == NULL)
            break;
        write_archivos(reng_act, posi_columna, aux_guarda,
            0, tamven_1);
        ++reng_act;
    }
    tamven_2 = tamven_2 - 8;
    posi_columna = 2;
    reng_act = 1;
    for(num_reng = 1 ; num_reng <= 8 ; num_reng++)
    {
        if(aux_guarda[tamven_2+reng_act] == NULL )
            break;
        write_archivos(reng_act, posi_columna, aux_guarda,
            0, tamven_2);
        ++reng_act;
    }
    posi_columna = col_temp;
    reng_act = 8;
}
else
{
    if( aux_guarda[tamven_1+(reng_act-1)-1] == NULL)
        continue;
    write_archivos(reng_act, posi_columna, aux_guarda,
        0, tamven_1);
    --reng_act;
}
}
if( teclaa == PGDN )
{
    if( ( tamven_2 + 8 ) < aux_size )
    {
        col_temp = posi_columna;
        col_reng = reng_act;
        tamven_1 = tamven_1 + 8;
        reng_act = 1;
        posi_columna = 1;
        textattr(atribut12);
        clrscr();
        linea();
        textattr(atribut12);
        for(num_reng = 1 ; num_reng <= 8 ; num_reng++ )
        {
            if(aux_guarda[tamven_1+reng_act-1] == NULL)
                break;
            write_archivos(reng_act, posi_columna, aux_guarda,
                0, tamven_1);
            ++reng_act;
        }
    }
}

```

```

}
tamven_2 = tamven_2 + 8;
posi_columna = 2;
reng_act = 1;
for(num_reng = 1 ; num_reng <= 8 ; num_reng++ )
{
    if(aux_guarda[tamven_2+reng_act] == NULL )
        break;
    write_archivos(reng_act, posi_columna, aux_guarda,
        0, tamven_2);
    ++reng_act;
}
posi_columna = col_temp;
reng_act = col_reng;
if(posi_columna == 1)
{
    if( aux_guarda[tamven_1+reng_act-1] == NULL )
        reng_act = 1;
}
else
{
    if( aux_guarda[tamven_2+reng_act] == NULL )
    {
        reng_act = 1;
        posi_columna = 1;
    }
}
}
if( teclaa == PGUP )
{
    if( tamven_1 != 0 )
    {
        col_temp = posi_columna;
        col_reng = reng_act;
        tamven_1 = tamven_1 - 8;
        reng_act = 1;
        posi_columna = 1;
        textattr( atribut12 );
        clrscr();
        linea();
        textattr( atribut12 );
        for(num_reng = 1 ; num_reng <= 8 ; num_reng++ )
        {
            if(aux_guarda[tamven_1+reng_act-1] == NULL)
                break;
            write_archivos(reng_act, posi_columna, aux_guarda,
                0, tamven_1);
            ++reng_act;
        }
        tamven_2 = tamven_2 - 8;
        posi_columna = 2;
        reng_act = 1;
        for(num_reng = 1 ; num_reng <= 8 ; num_reng++ )

```

```

{
    if(aux_guarda[tamven_2+reng_act] == NULL )
        break;
    write_archivos(reng_act, posi_columna, aux_guarda,
                   0, tamven_2);
    ++reng_act;
}
posi_columna = col_temp;
reng_act = col_reng;
}
}
if( teclaa == HOME )
{
    if( posi_columna == 1 )
    {
        write_archivos(reng_act, posi_columna, aux_guarda,
                       0, tamven_1);
        posi_columna = 1;
        reng_act = 1;
    }
    else
    {
        write_archivos(reng_act, posi_columna, aux_guarda,
                       0, tamven_2);
        posi_columna = 1;
        reng_act = 1;
    }
}
if( teclaa == END )
{
    if( posi_columna == 1 )
        write_archivos(reng_act, posi_columna, aux_guarda,
                       0, tamven_1);
    else
        write_archivos(reng_act, posi_columna, aux_guarda,
                       0, tamven_2);
    col_temp = 0;
    reng_temp = 0;
    posi_columna = 1;
    reng_act = 1;
    for( num_reng = 1 ; num_reng <= 8 ; num_reng++ )
    {
        if(aux_guarda[tamven_1+reng_act-1] == NULL)
        {
            col_temp = posi_columna;
            reng_temp = reng_act-1 ;
            break;
        }
        ++reng_act;
    }
    if( col_temp != 0 && reng_temp != 0 )
    {
        posi_columna = col_temp;
        reng_act = reng_temp;
    }
}

```



```

        continue;
    }
    posi_columna = 2;
    reng_act = 1;
    for(num_reng = 1 ; num_reng <= 8 ; num_reng++ )
    {
        if(aux_guarda[tamven_2+reng_act] == NULL )
        {
            col_temp = posi_columna;
            reng_temp = reng_act-1;
            break;
        }
        ++reng_act;
    }
    if( col_temp != 0 && reng_temp != 0 )
    {
        posi_columna = col_temp;
        reng_act = reng_temp;
        continue;
    }
    if( posi_columna == 1 )
    {
        reng_act = 8;
        posi_columna = 2;
    }
    else
    {
        reng_act = 8;
        posi_columna = 2;
    }
}
if( teclaa == ENTER )
{
    if( posi_columna == 1 )
    {
        strcpy(nombre_archivo,
                aux_guarda[tamven_1+reng_act-1]);
        teclaa = ESC;
    }
    else
    {
        strcpy(nombre_archivo,
                aux_guarda[tamven_2+reng_act]);
        teclaa = ESC;
    }
}
}
}

/** ESTA FUNCION LLENA EL NUMERO DE LINEA CON NOMBRES */
/** DE ARCHIVOS CON SOMBRA O SIN SOMBRA **/
void write_archivos(reng_act, posi_columna, aux_guarda,
aux_pasada, aux_tam_vent)
int reng_act, posi_columna;

```

```

int aux_pasada;
int aux_tam_vent;
char *aux_guarda[100];
{
    if( aux_pasada == 0 )
    {
        textattr(atribut12);
        if ( posi_columna == 1 )
        {
            gotoxy(columna_archivos(posi_columna), reng_act);
            cputs(aux_guarda[aux_tam_vent+reng_act-1]);
        }
        else
        {
            gotoxy(columna_archivos(posi_columna), reng_act);
            cputs(aux_guarda[aux_tam_vent+reng_act]);
        }
    }
    else
    {
        textattr(atribut13);
        if ( posi_columna == 1 )
        {
            gotoxy(columna_archivos(posi_columna), reng_act);
            cputs(aux_guarda[aux_tam_vent+reng_act-1]);
        }
        else
        {
            gotoxy(columna_archivos(posi_columna), reng_act);
            cputs(aux_guarda[aux_tam_vent+reng_act]);
        }
        textattr(atribut12);
    }
    return;
}

```

/* NOS DA DONDE ES LA POSICION DELOS CEROS A PINTAR */

```
int columna_archivos(posi_columna )
```

```
int posi_columna ;
```

```

{
    int pos;

    switch(posi_columna ) {
        case 1:
            pos = 2;
            break;
        case 2:
            pos = 17;
            break;
    }
    return (pos);
}

```

/** LINEA **/

```

void linea()
{
    int i=0;
    for(i=1; i<9 ; i++)
    {
        textattr(atribut14);
        gotoxy(15, i);
        cprintf("|");
    }
    textattr(atribut12);
}

/** TRAER DATOS AL ARREGLO **/
void traer_datos_arreglo(fp, bytes , arreglo)
int fp, bytes;
char *arreglo;
{
    if( read(fp, arreglo, bytes ) != bytes)
    {
        window(12,20,40,20);
        clrscr();
        gotoxy(1,1);
        printf(" ERROR en la lectura de bytes ");
        sleep(1);
        return;
    }
}

/** SALVAR UN ARCHIVO **/
void guardar()
{
    union REGS rin,rout;
    union REGS mousin,mousout;
    int fp;
    char opci;
    int te;
    char *c= "\\";

    strcpy(aux_path, path_archivo);
    gotoxy(4,2);
    printf("Subdirectorio a usar ...");
    gotoxy(4,5);
    printf("Nombre del archivo a guardar");
    textattr(atribut12);
    window(12,9,40,9);
    clrscr();
    cad_temp[0]='\0';
    aux_path[0]='\0';
    path_archivo[0]='\0';
    current_directori(aux_path);
    strcpy(cad_temp,aux_path);
    gotoxy(1,1);
    printf("%s",aux_path);
    gotoxy(1,1);
}

```

```

mousin.x.ax = 10;
mousin.x.bx = 1;
mousin.x.cx = 3;
mousin.x.dx = 7;
int86(0x33, &mousin, &mousout);
gets(aux_path);
if( strlen(aux_path) == 0 ) strcpy(aux_path, cad_temp);
if( chdir(aux_path) == -1 )
{
    textattr(atribut4);
    window(12,12,52,19);
    clrscr();
    gotoxy(1,2);
    printf("          ■ E R R O R ■ ");
    gotoxy(1,3);
    printf("SUBDIRECTORIO : ");
    gotoxy(1,4);
    printf("» %s ", aux_path);
    gotoxy(1,5);
    printf("INEXISTENTE VERIFICAR ");
    gotoxy(1,6);
    printf("          DRIVE:\\SUBDIRECTORIO");
    gotoxy(1,7);
    printf("NOTA: Presione cualquier tecla ");
    gotoxy(1,8);
    printf("          para continuar.");
    if (!getch())
        path_archivo[0]='\0';
    strcpy(path_archivo, aux__path);
    return;
}
strcpy(path_archivo, aux_path);
textattr(atribut12);
window(12,12,45,12);
clrscr();
gotoxy(1,1);
cad_temp[0]='\0';
strcpy(cad_temp, nombre_archivo);
printf("%s", nombre_archivo);
gotoxy(1,1);
gets(nombre_archivo);
if( strlen(nombre_archivo) == 0 )
    strcpy(nombre_archivo, cad_temp);
if( strlen(aux_path) > 3 ) strcat(path_archivo, c);
strncat(path_archivo, nombre_archivo,
strlen(nombre_archivo));
fp = open(path_archivo , O_BINARY | O_WRONLY | O_CREAT |
O_EXCL , S_IREAD | S_IWRITE);
if( errno == EEXIST )
{
    textattr(atribut4);
    window(12,15,50,15);
    clrscr();
    gotoxy(1,1);

```

```

printf("EL ARCHIVO EXITE SOBRESERIBIR S/N ");
gotoxy(37,1);
printf("[ ]");
opci=toupper(getch());
gotoxy(38,1);
printf("%c",opci);
if( opci == 'S' )
{
    fp = open(path_archivo , O_BINARY | O_WRONLY |
              O_CREAT | O_TRUNC , S_IREAD | S_IWRITE );
    if( fp == -1 )
    {
        window(12,13,50,13);
        clrscr();
        gotoxy(1,1);
        printf(" ERROR EN LA APERTURA DE ARCHIVO " );
        return;
    }
    if( write(fp, dato_ini , sizeof(dato_ini) ) !=
        sizeof(dato_ini))
    {
        window(12,14,50,14);
        clrscr();
        gotoxy(1,1);
        printf(" ERROR EN LA ESCRITURA DE BYTES " );
    }
    salvar_datos_arreglo(fp, sizeof(p_n_x_1), (char *)p_n_x_1);
    salvar_datos_arreglo(fp, sizeof(p_n_y_1), (char *)p_n_y_1);
    salvar_datos_arreglo(fp, sizeof(p_n_x_2), (char *)p_n_x_2);
    salvar_datos_arreglo(fp, sizeof(p_n_y_2), (char *)p_n_y_2);
    salvar_datos_arreglo(fp, sizeof(p_n_z_2), (char *)p_n_z_2);
    salvar_datos_arreglo(fp, sizeof(p_n_a_2), (char *)p_n_a_2);
    salvar_datos_arreglo(fp, sizeof(p_n_b_2), (char *)p_n_b_2);
    salvar_datos_arreglo(fp, sizeof(p_n_c_2), (char *)p_n_c_2);
    salvar_datos_arreglo(fp, sizeof(p_n_d_2), (char *)p_n_d_2);
    salvar_datos_arreglo(fp, sizeof(p_n_e_2), (char *)p_n_e_2);
    salvar_datos_arreglo(fp, sizeof(p_n_f_2), (char *)p_n_f_2);
    salvar_datos_arreglo(fp, sizeof(p_n_x_3), (char *)p_n_x_3);
    salvar_datos_arreglo(fp, sizeof(p_n_y_3), (char *)p_n_y_3);
    salvar_datos_arreglo(fp, sizeof(p_n_x_4), (char *)p_n_x_4);
    salvar_datos_arreglo(fp, sizeof(p_n_y_4), (char *)p_n_y_4);
    salvar_datos_arreglo(fp, sizeof(p_n_z_4), (char *)p_n_z_4);
    salvar_datos_arreglo(fp, sizeof(p_n_x_5), (char *)p_n_x_5);
    salvar_datos_arreglo(fp, sizeof(p_n_x_6), (char *)p_n_x_6);
    salvar_datos_arreglo(fp, sizeof(p_n_y_6), (char *)p_n_y_6);
    close(fp);
    archi_extra(dato_ini);
    consulta_archivo(dato_ini);
}
else
return;
}
else
{

```

```

if( write(fp, dato_ini , sizeof(dato_ini) ) !=
    sizeof(dato_ini))
{
    window(12,13,50,13);
    clrscr();
    gotoxy(1,1);
    printf(" ERROR EN LA APERTURA DE ARCHIVO " );
}

salvar_datos_arreglo(fp, sizeof(p_n_x_1), (char *)p_n_x_1);
salvar_datos_arreglo(fp, sizeof(p_n_y_1), (char *)p_n_y_1);
salvar_datos_arreglo(fp, sizeof(p_n_x_2), (char *)p_n_x_2);
salvar_datos_arreglo(fp, sizeof(p_n_y_2), (char *)p_n_y_2);
salvar_datos_arreglo(fp, sizeof(p_n_z_2), (char *)p_n_z_2);
salvar_datos_arreglo(fp, sizeof(p_n_a_2), (char *)p_n_a_2);
salvar_datos_arreglo(fp, sizeof(p_n_b_2), (char *)p_n_b_2);
salvar_datos_arreglo(fp, sizeof(p_n_c_2), (char *)p_n_c_2);
salvar_datos_arreglo(fp, sizeof(p_n_d_2), (char *)p_n_d_2);
salvar_datos_arreglo(fp, sizeof(p_n_e_2), (char *)p_n_e_2);
salvar_datos_arreglo(fp, sizeof(p_n_f_2), (char *)p_n_f_2);
salvar_datos_arreglo(fp, sizeof(p_n_x_3), (char *)p_n_x_3);
salvar_datos_arreglo(fp, sizeof(p_n_y_3), (char *)p_n_y_3);
salvar_datos_arreglo(fp, sizeof(p_n_x_4), (char *)p_n_x_4);
salvar_datos_arreglo(fp, sizeof(p_n_y_4), (char *)p_n_y_4);
salvar_datos_arreglo(fp, sizeof(p_n_z_4), (char *)p_n_z_4);
salvar_datos_arreglo(fp, sizeof(p_n_x_5), (char *)p_n_x_5);
salvar_datos_arreglo(fp, sizeof(p_n_x_6), (char *)p_n_x_6);
salvar_datos_arreglo(fp, sizeof(p_n_y_6), (char *)p_n_y_6);
    close(fp);
    archi_extra(dato_ini);
    consulta_archivo(dato_ini);
}

rin.h.ah = 1;
rin.h.ch = 0x20;
int86(0x10, &rin, &rout);
mousin.x.ax = 10;
mousin.x.bx = 1;
mousin.x.cx = 3;
mousin.x.dx = 7;
int86(0x33, &mousin, &mousout);
strcpy(path_archivo, aux_path);
}

/* SALVAR DATOS DEL ARREGLO */
void salvar_datos_arreglo(fp, bytes, arreglo)
int fp, bytes;
char *arreglo;
{
    if( write(fp, arreglo, bytes) != bytes)
    {
        window(12,14,50,14);
        gotoxy(1,1);
        printf(" ERROR en la escritura de bytes ");
        sleep(2);
    }
}

```

```

}

/** ARCHIVO EXTRA PARA EL PROCESADOR **/
void archi_extra(int dato_ini[])
{
    int i;

    i=0;
    textattr(atribut4);
    window(12,14,50,19);
    clrscr();
    gotoxy(1,1);
    printf("Creando archivo de INFORMACION * .FMT ");
    gotoxy(1,2);
    printf("para el funcionamiento del PROCESADOR:");
    aux_path[0]='\0';
    while( (path_archivo[i] != '.') )    i++;
    strncat(aux_path,path_archivo,i);
    if( strstr(aux_path, ".fmt") == NULL )
        strcat(aux_path, ".fmt");
    gotoxy(1,3);
    printf("%s", aux_path);
    delay(2000);
    unlink(aux_path);
    if( (formato_archivo = fopen (aux_path, "w") ) == NULL )
        return;
    fprintf(formato_archivo, "%2d\n", dato_ini[0]);
    fprintf(formato_archivo, "%2d\n", dato_ini[1]);
    fprintf(formato_archivo, "%2d\n", dato_ini[3]);
    fprintf(formato_archivo, "%2d\n", dato_ini[4]);
    fprintf(formato_archivo, "%2d\n", dato_ini[5]);
    if (dato_ini[0] > 0)
        arre_archi_extra(dato_ini[0], &pun_nod, p_n_x_1, p_n_y_1,
            p_n_z_2, p_n_a_2, p_n_b_2, p_n_c_2, p_n_d_2, p_n_e_2,
            p_n_f_2, 1);
    if (dato_ini[2] > 0)
        arre_archi_extra(dato_ini[2], &frontera, p_n_x_3, p_n_y_3,
            p_n_z_2, p_n_a_2, p_n_b_2, p_n_c_2, p_n_d_2, p_n_e_2,
            p_n_f_2, 2);
    if (dato_ini[3] > 0)
        arre_archi_extra(dato_ini[3], &materiales, p_n_x_4,
            p_n_y_4, p_n_z_4, p_n_a_2, p_n_b_2, p_n_c_2, p_n_d_2,
            p_n_e_2, p_n_f_2, 3);
    if (dato_ini[1] > 0)
        arre_archi_extra(dato_ini[1], &elementos, p_n_x_2,
            p_n_y_2, p_n_z_2, p_n_a_2, p_n_b_2, p_n_c_2,
            p_n_d_2, p_n_e_2, p_n_f_2, 1);
    if (dato_ini[4] > 0)
        arre_archi_extra(dato_ini[4], &nodos_cargados, p_n_x_5,
            p_n_y_2, p_n_z_2, p_n_a_2, p_n_b_2, p_n_c_2,
            p_n_d_2, p_n_e_2, p_n_f_2, 1);
    if (dato_ini[5] > 0)
        arre_archi_extra(dato_ini[5], &cargas, p_n_x_6, p_n_y_6,
            p_n_z_2, p_n_a_2, p_n_b_2, p_n_c_2, p_n_d_2, p_n_e_2,

```

```

        p_n_f_2,1);
fclose(formato_archivo);
}

/** ARREGLO DEL ARCHIVO EXTRA PARA EL PROCESADOR **/
void arre_archi_extra(size,vent,t_x, t_y, t_z, t_a,
t_b,t_c,t_d,t_e,t_f,bloccke)
int size;
int bloccke;
struct ventana *vent;
double t_x[], t_y[], t_z[], t_a[],
t_b[],t_c[],t_d[],t_e[],t_f[];
{
    int i;
    int reng_act = 1;
    int columna=2;
    if ( bloccke == 1 )
    {
        for (columna = 2; columna <= vent->columnas - 1 ;
            columna++)
        {
            lin_archi_extra(reng_act, columna, vent, t_x, t_y,
                t_z, t_a, t_b,t_c,t_d,t_e,t_f,size,1);
            ++reng_act;
        }
    }
    if ( bloccke == 2 )
    {
        for (reng_act = 1; reng_act <= size; reng_act++)
        {
            lin_archi_extra(reng_act, columna,vent, t_x, t_y,
                t_z, t_a, t_b,t_c,t_d,t_e,t_f,size,2);
        }
    }
    if ( bloccke == 3 )
    {
        for (reng_act = 1; reng_act <= size; reng_act++)
        {
            lin_archi_extra(reng_act, columna,vent, t_x, t_y,
                t_z, t_a, t_b,t_c,t_d,t_e,t_f,size,3);
        }
    }
}

/** LINEA EXTRA PARA EL ARCHIVO DEL PROCESADOR **/
void lin_archi_extra(reng_act, columna, vent, t_x, t_y, t_z,
t_a, t_b,t_c,t_d,t_e,t_f,size,aux_bloccke)
int reng_act, columna;
struct ventana *vent;
double t_x[], t_y[], t_z[], t_a[],
t_b[],t_c[],t_d[],t_e[],t_f[];
int size;
int aux_bloccke;

```



```

{
    int col_act;
    int i;
    char *line, *fmt;
    double *t[9];

    t[0] = t_x;
    t[1] = t_y;
    t[2] = t_z;
    t[3] = t_a;
    t[4] = t_b;
    t[5] = t_c;
    t[6] = t_d;
    t[7] = t_e;
    t[8] = t_f;
    if ( aux_bloccke == 1 )
    {
        for(reng_act = 1; reng_act <= size ; reng_act++)
        {
            fmt = formato(vent,columna);
            sprintf(line, fmt, *(t[ columna - 2 ] + reng_act));
            fprintf(formato_archivo, "%s\n", line);
            free(fmt);
        }
    }
    if (aux_bloccke == 2 )
    {
        line = malloc(136);
        for(col_act = 2; col_act < vent->columnas; col_act++)
        {
            fmt = formato(vent,col_act);
            sprintf(line, fmt, *(t[ col_act - 2 ] + reng_act));
            if( vent->columnas == 4 )
                fprintf(formato_archivo, "%s\n",line);
            if( vent->columnas == 11 )
                fprintf(formato_archivo, "%s\n",line);
            if( vent->columnas == 5 )
                fprintf(formato_archivo, "%s\n",line);
            if( vent->columnas == 3 )
                fprintf(formato_archivo, "%s\n",line);
            free(fmt);
        }
    }
    if (aux_bloccke == 3 )
    {
        line = malloc(136);
        sprintf(line, "%2d", reng_act);
        if ( aux_bloccke == 3 )
        {
            strcat(line, ".000");
        }
        fprintf(formato_archivo, "%s\n", line);
        for(col_act = 2; col_act < vent->columnas; col_act++)
    }
}

```

```

        fmt = formato(vent,col_act);
        sprintf(line, fmt, *(t[ col_act - 2 ] +
reng_act));
        if( vent->columnas == 4 )
            fprintf(formato_archivo, "%s\n",line);
        if( vent->columnas == 11 )
            fprintf(formato_archivo, "%s\n",line);
        if( vent->columnas == 5 )
            fprintf(formato_archivo, "%s\n",line);
        if( vent->columnas == 3 )
            fprintf(formato_archivo,"%s\n",line);
        free(fmt);
    }
}

/** ELABORAR UN PROGRAMA DE CONSULTA **/
void consulta_archivo(int dato_ini[])
{
    int i;

    i=0;
    textattr(atribut4);
    gotoxy(1,4);
    printf("Creando archivo de CONSULTA *.CNS ");
    gotoxy(1,5);
    printf("en el disco para el 'u s u a r i o'");
    aux_path[0]='\0';
    while( (path_archivo[i] != '.') )    i++;
    strncpy(aux_path,path_archivo,i);
    if( strstr(aux_path,".cns") == NULL )
        strcat(aux_path,".cns");
    gotoxy(1,6);
    printf("%s",aux_path);
    delay(2000);
    unlink(aux_path);
    if( (consulta = fopen (aux_path,"w") ) == NULL )
        return;
    fprintf(consulta,"ARCHIVO EN CONSULTA:
        %s\n\r",aux_path);
    if (dato_ini[0] > 0)
        archivo_escritura(dato_ini[0], &pun_nod, p_n_x_1,
            p_n_y_1,p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
            p_n_e_2,p_n_f_2);
    if (dato_ini[1] > 0)
    {
        elementos.subtitulo=" ELE.  NOD_I  NOD_J  NOD_K
            NOD_L  NOD_M  NOD_N  NOD_O  NOD_P  MATERIAL ";
        archivo_escritura(dato_ini[1], &elementos,p_n_x_2,
            p_n_y_2,p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
            p_n_e_2,p_n_f_2);
    }
    if (dato_ini[2] > 0)
    {

```

```

    frontera.subtitulo=" No.    DX    DY ",
    archivo_escritura(dato_ini[2], &frontera, p_n_x_3,
        p_n_y_3,p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
        p_n_e_2,p_n_f_2);
}
if (dato_ini[3] > 0)
{
    materiales.subtitulo=" No.          E          v
        Espesor",
    archivo_escritura(dato_ini[3],&materiales,p_n_x_4,
        p_n_y_4,p_n_z_4,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
        p_n_e_2,p_n_f_2);
}
if (dato_ini[4] > 0)
{
    nodos_cargados.subtitulo=" CARGA    NODO ",
    archivo_escritura(dato_ini[4], &nodos_cargados,
        p_n_x_5,p_n_y_2,p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,
        p_n_d_2,p_n_e_2,p_n_f_2);
}
if (dato_ini[5] > 0)
{
    cargas.subtitulo=" No.          FX          FY ",
    archivo_escritura(dato_ini[5], &cargas, p_n_x_6,
        p_n_y_6,p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
        p_n_e_2,p_n_f_2);
}
}

/* ARCHIVO DE ESCRITURA PARA CONSULTA DE DATOS EN DISCO */
void archivo_escritura(size, vent, t_x, t_y, t_z, t_a,
    t_b,t_c,t_d,t_e,t_f)
int size;
struct ventana *vent;
double t_x[], t_y[], t_z[], t_a[],
    t_b[],t_c[],t_d[],t_e[],t_f[];
{
    int i;
    int ren = 1;

    fprintf(consulta,"%s\n\r", vent->titulo);
    fprintf(consulta,"%s\n\r", vent->subtitulo);
    for (i = 1; i <= size; i++) {
        linea_escritura(i, vent, t_x, t_y, t_z, t_a,
            t_b,t_c,t_d,t_e,t_f);
        ++ren;
    }
    fprintf(consulta,"\n");
}

/* LINEA DE ESCRITURA PARA ARCHIVO DE CONSULTA EN DISCO */
void linea_escritura(linea, vent, t_x, t_y, t_z, t_a,
    t_b,t_c,t_d,t_e,t_f)
int linea;

```

```

struct ventana *vent;
double t_x[], t_y[], t_z[], t_a[],
t_b[], t_c[], t_d[], t_e[], t_f[];
{
    int col_act;
    int i;
    char *line, *fmt;
    double *t[9];

    t[0] = t_x;
    t[1] = t_y;
    t[2] = t_z;
    t[3] = t_a;
    t[4] = t_b;
    t[5] = t_c;
    t[6] = t_d;
    t[7] = t_e;
    t[8] = t_f;
    line = malloc(136);
    sprintf(line, "%2d", linea);
    fprintf(consulta, " %s ", line);
    for(col_act = 2; col_act < vent->columnas; col_act++)
    {
        fmt = formato(vent, col_act);
        sprintf(line, fmt, *(t[ col_act - 2 ] + linea));
        if( vent->columnas == 4 )
            fprintf(consulta, " %s ", line);
        if( vent->columnas == 11 )
            fprintf(consulta, " %s ", line);
        if( vent->columnas == 5 )
            fprintf(consulta, " %s ", line);
        if( vent->columnas == 3 )
            fprintf(formato_archivo, " %s", line);
        free(fmt);
    }
    fprintf(consulta, "\n");
    free(line);
}

/** DIRECTORIO **/
void dir()
{
    struct ffblk ffblk;
    union REGS mousin, mousout;
    union REGS rin, rout;
    char aux_archivo[80];
    int done;
    int i;
    char cur_dir[80];
    int size=0;
    int fp;
    int te;

    i=0;

```



```

printf("» %s  ", aux_path);
gotoxy(1,5);
printf("INEXISTENTE   VERIFICAR ");
gotoxy(1,6);
printf("                DRIVE/SUBDIRECTORIO... ");
gotoxy(1,7);
printf("NOTA: Presione cualquier tecla ");
gotoxy(1,8);
printf("                para continuar.");
if (!(getch()))
path_archivo[0]='\0';
strcpy(path_archivo,aux_path);
return;
}
strcpy(path_archivo,aux_path);
clrscr();
gotoxy(1,1);
printf("%s",aux_path);
gotoxy((strlen(aux_path)+1),1);
if( strlen(aux_path) == 3 )
{
printf("*.");
strncat(aux_path,"*. ",5);
}
else
{
printf("\\*.");
strncat(aux_path,"\\*. ",7);
}
done = findfirst( aux_path , &ffblk, 0);
if( done == -1 )
{
textattr(atribut12);
window(12,13,40,20);
clrscr();
gotoxy(1,1);
printf("No hay archivos");
return;
}
i=0;
while(!done)
{
guarda[i] = malloc(14);
strcpy(guarda[i],ffblk.ff_name);
done = findnext(&ffblk);
i=i+1;
}
size = i;
textattr(atribut12);
window(12,12,40,19);
clrscr();
ventana_archivos(size,guarda,coord_directorio);
strcpy(nombre_archivo,aux_archivo);
if( (strlen(path_archivo) == 3 ) )

```

```

        strcat(path_archivo,nombre_archivo,
               strlen(nombre_archivo));
if( (strlen(path_archivo) > 3 ) )
{
    strcat(path_archivo,"\\");
    strcat(path_archivo,nombre_archivo,
           strlen(nombre_archivo));
}
path_archivo[0]='\0';
strcpy(path_archivo,aux__path);
}

```

```

/** BORRAR UN ARCHIVO **/
void borrar()

```

```

{
    union REGS rin,rout;
    union REGS mousin,mousout;
    struct fblk ffblk;
    int done;
    int i;
    char cur_dir[80];
    int fp;
    int te;
    char opci;

    strcpy(aux__path,path_archivo);
    gotoxy(4,2);
    printf("Subdirectorío a consultar ...");
    gotoxy(4,5);
    printf("Archivo ...");
    textattr(atribut12);
    window(12,11,40,11);
    clrscr();
    cad_temp[0]='\0';
    aux_path[0]='\0';
    path_archivo[0]='\0';
    current_directori(aux_path);
    strcpy(cad_temp,aux_path);
    gotoxy(1,1);
    printf("%s",aux_path);
    gotoxy(1,1);
    mousin.x.ax = 10;
    mousin.x.bx = 1;
    mousin.x.cx = 3;
    mousin.x.dx = 7;
    int86(0x33,&mousin,&mousout);
    gets(aux_path);
    rin.h.ah = 1;
    rin.h.ch = 0x20;
    int86(0x10,&rin,&rout);
    mousin.x.ax = 10;
    mousin.x.bx = 1;
    mousin.x.cx = 3;
    mousin.x.dx = 7;
}

```

```

int86(0x33, &mousin, &mousout);
if( strlen(aux_path) == 0 ) strcpy(aux_path, cad_temp);
if( chdir(aux_path) == -1 )
{
    textattr(atribut4);
    window(12,12,50,19);
    clrscr();
    gotoxy(1,2);
    printf("      ■ E R R O R ■   ");
    gotoxy(1,3);
    printf("SUBDIRECTORIO : ");
    gotoxy(1,4);
    printf("» %s  ", aux_path);
    gotoxy(1,5);
    printf("INEXISTENTE   VERIFICAR ");
    gotoxy(1,6);
    printf("          DRIVE/SUBDIRECTORIO... ");
    gotoxy(1,7);
    printf("NOTA: Presione cualquier tecla ");
    gotoxy(1,8);
    printf("      para continuar.");
    if (!(getch()))
        path_archivo[0]='\0';
    strcpy(path_archivo, aux_path);
    return;
}
strcpy(path_archivo, aux_path);
clrscr();
gotoxy(1,1);
printf("%s", aux_path);
textattr(atribut12);
window(12,14,35,14);
clrscr();
cad_temp[0]='\0';
strcpy(cad_temp, nombre_archivo);
printf("%s", nombre_archivo);
gotoxy(1,1);
gets(nombre_archivo);
if( strlen(nombre_archivo) == 0 )
    strcpy(nombre_archivo, cad_temp);
if( (strlen(aux_path) > 3 ) ) strcat(aux_path, "\\");
strncat(aux_path, nombre_archivo, strlen(nombre_archivo));
done = findfirst( aux_path, &ffblk, 0);
if( done == -1 )
{
    textattr(atribut4);
    window(12,16,50,18);
    clrscr();
    gotoxy(1,1);
    printf("No hay ningun archivo con ese nombre");
    gotoxy(1,2);
    printf("%s", aux_path);
    delay(2000);
    return;
}

```



```

}
textattr(atribut4);
window(12,16,45,19);
clrscr();
gotoxy(1,1);
printf("Confirme si desea borrar :");
gotoxy(1,2);
printf(" %s ",aux_path);
gotoxy(1,3);
printf("del disco ( S / N ) [ ].");
opci=' ';
while ( opci !='S' && opci !='N' )
{
    opci=toupper(getch());
}
gotoxy(25,3);
printf("%c",opci);
delay(2000);
if( opci == 'S')
{
    unlink(aux_path);
    return;
}
rin.h.ah = 1;
rin.h.ch = 0x20;
int86(0x10, &rin, &rout);
mousin.x.ax = 10;
mousin.x.bx = 1;
mousin.x.cx = 3;
mousin.x.dx = 7;
int86(0x33, &mousin, &mousout);
strcpy(path_archivo,aux_path);
}

```

PREPROCESADOR INCLUYE AL ARCHIVO: DATOS.C

```
#include "libreria.h"
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <stat.h>
#include <dir.h>
#include <errno.h>
```

```
extern int dato_ini[] ;
```

```
int determina_tamano_size(char *buf1,char *letre_ar[],
char *carac2[],int dat_ini2[],int aux_blocke2,int aux_n);
void tamano_ventana(int size, struct ventana *vent,
double t_x[],double t_y[],double t_z[],double t_a[],
double t_b[],double t_c[],double t_d[],double t_e[],
double t_f[],int aux_blocke2,int cord_graf[]);
void escribe_linea_ceros(int reng_act,int num_reng,
int posi_columna,struct ventana *vent, double t_xx[],
double t_yy[],double t_zz[],double t_aa[],
double t_bb[],double t_cc[],double t_dd[],
double t_ee[],double t_ff[],int aux_pasada);
char *formato(struct ventana *vent,int posi_columna);
int posicion(struct ventana *vent,int posi_columna);
int tamano(char *fmt);
```

```
/** ESTA FUNCION PIDE EL TAMANO DEL CONTENIDO **/
/** DE LA VENTANA Y LA ABRE **/
int determina_tamano_size(buf1,letre_ar,carac2,dat_ini2,
aux_blocke2,aux_n)
char buf1[81*25*2];
char *letre_ar[];
char *carac2[];
int dat_ini2[];
int aux_blocke2;
int aux_n;
```

```
{
    int c,i;
    char vals[4];
    int n;

    textcolor(WHITE);
    gotoxy(3,(aux_blocke2+1));
    cprintf("%s",letre_ar[aux_blocke2-1]);
    textcolor(YELLOW);
    gotoxy(3,(aux_blocke2+1));
    cprintf("%s",carac2[aux_blocke2-1]);
    gotoxy(33,(aux_blocke2+1));
    textattr(atribut2);
```

```

cprintf("%2d",dat_ini2[aux_bloque2-1]);
sprintf(vals,"%2d",dat_ini2[aux_bloque2-1]);
i=0;
do
{
    if (!(c=getch())) c = getch() | 128 ;
    if( c == ENTER && i == 0) break;
    if( c == ENTER && i == 1)
    {
        vals[i++] = '\0';
        gotoxy(32+i, (aux_bloque2+1));
        putchar(' ');
        gotoxy(32+(++i), (aux_bloque2+1));
        putchar(c);
    }

    if( c == ENTER && i == 2)
    {
        vals[i++] = '\0';
        gotoxy(32+i, (aux_bloque2+1));
        putchar(' ');
        gotoxy(32+(++i), (aux_bloque2+1));
        putchar(c);
    }

    if( c == ESC )
    {
        i=0;
        break;
    }

    if ( isdigit(c) )
    {
        vals[i++] = c;
        gotoxy(32+i, (aux_bloque2+1));
        putchar(c);
    }
}while ( i < 3 );
if( c == ENTER && i == 0 )
{
    gotoxy(33, (aux_bloque2+1));
    cprintf("%2d",dat_ini2[aux_bloque2-1]);
    return(dat_ini2[aux_bloque2-1]);
}
if( i != 0 && i > 0 )
{
    dat_ini2[aux_bloque2-1]=atoi(vals);
    aux_n=dat_ini2[aux_bloque2-1];
    return(aux_n);
}
if( c == ESC )
{
    dat_ini2[aux_bloque2-1]=aux_n;
    gotoxy(33, (aux_bloque2+1));
    cprintf("%2d",dat_ini2[aux_bloque2-1]);
    gettext(1,1,80,24,buf1);
}

```

```

        return(0);
    }
    return(0);
}

/* ESTABLECEMOS LOS LIMITES DE LA VENTANA DE PUN.NODALES */
void tamano_ventana(size,vent,t_x,t_y,t_z,t_a,t_b,t_c,t_d,
t_e,t_f,aux_bloque2, cord_graf)
int size;
struct ventana *vent;
double t_x[],t_y[],t_z[],t_a[],t_b[],t_c[],t_d[],t_e[],
t_f[];
int aux_bloque2;
int cord_graf[];
{
    int i;
    int voy_reng,voy_col;
    int ren_temp,num_temp; /* ,col_temp; */
    int teclaa=0;
    char *buf3;
    int *apu_num_reng=&num_reng;
    int *apu_renglon=&reng_act;
    int *apu_columna=&posi_columna;
    char *s;
    char *blank;
    char *fmt;
    int tam;
    int c;
    double *t[9];

    top=1;
    lim_sup = vent -> top - aux_bloque2 ;
    lim_inf = vent -> bottom - 6 - aux_bloque2 + 1 ;
    reng_act = lim_sup;
    tam_vent = ( lim_inf - lim_sup )+1;
    num_reng = 1;
    posi_columna = 1;

    if ( size < tam_vent )
    {
        lim_inf = ( lim_sup + size ) - 1;
        tam_vent = ( lim_inf - lim_sup ) + 1;
    }
    ren_temp = 1;
    for(num_reng = top ; num_reng < top + tam_vent ;
        num_reng++)
    {
        escribe_linea_ceros(reng_act,num_reng,posi_columna,
            vent,t_x,t_y,t_z,t_a,t_b,t_c,t_d,t_e,t_f,0);
        ++reng_act;
    }
    reng_act = 4;
    num_reng = 1;
    posi_columna = 2;
}

```

```

while ( teclaa != ESC)
{
    num_reng = reng_act - lim_sup + top;
    escribe_linea_ceros(reng_act,num_reng, posi_columna,
        vent,t_x,t_y,t_z,t_a,t_b,t_c,t_d,t_e,t_f,1);
    teclaa=pres_3_tecla(teclaa,apu_num_reng,apu_renglon,
        apu_columna,vent,top,tam_vent,t_x,t_y,
        t_z,t_a,t_b,t_c,t_d,t_e,t_f, cord_graf,
        lim_sup,aux_bloque2);
    teclaa=toupper(teclaa);
    if(teclaa == FLEADEL)
    {
        if (posi_columna < vent->columnas-1)
            ++posi_columna;
    }
    if(teclaa == FLEATRA)
    {
        if (posi_columna > 2)
            --posi_columna;
    }
    if(teclaa == PGDN )
    {
        if( (top + tam_vent) <= size )
        {
            ren_temp = reng_act;
            reng_act = lim_sup;
            if( ( top + (tam_vent * 2 ) - 1 ) <= size )
                top = top + tam_vent;
            else
                top = size - tam_vent + 1;
            for(num_reng = top ; num_reng < top + tam_vent ;
                num_reng++)
            {
                escribe_linea_ceros(reng_act,num_reng,
                    posi_columna,vent,t_x,t_y,t_z,t_a,t_b,
                    t_c,t_d,t_e,t_f,0);
                ++reng_act;
            }
            reng_act=ren_temp;
        }
    }
    if( teclaa == PGUP)
    {
        if(top > 1)
        {
            ren_temp = reng_act;
            reng_act = lim_sup;
            if( (top-tam_vent) >= 1 )
                top = top - tam_vent;
            else
                top = 1;
            for(num_reng = top ; num_reng < top + tam_vent ;
                num_reng++)
            {

```

```

        escribe_linea_ceros(reng_act,num_reng,
                           posi_columna,vent,t_x,t_y,t_z,t_a,
                           t_b,t_c,t_d,t_e,t_f,0);
        ++reng_act;
    }
    reng_act=ren_temp;
}
}
if( teclaa == FLEARRI )
{
    num_reng = reng_act - lim_sup + top;
    buf3 = malloc( ( (vent->bottom-2)-(vent->top+3)+1 )
                  * ( (vent->right-2)-(vent->left+2)+1)*2);
    if( num_reng > 1 )
    {
        escribe_linea_ceros(reng_act,num_reng,
                           posi_columna,vent,t_x,t_y,t_z,t_a,t_b,
                           t_c,t_d,t_e,t_f,0);
        if ( reng_act > lim_sup )
            --reng_act;
        else
        {
            gettext(vent->left+2,vent->top+3,vent->right-
2,vent->bottom-3,buf3);
            puttext(vent->left+2,vent->top+4,vent->right-
2,vent->bottom-2,buf3);
            --top;
        }
    }
    free(buf3);
}
if( teclaa == FLEABAJ )
{
    num_reng = reng_act - lim_sup + top;
    buf3 = malloc(((vent->bottom-2)-(vent->top+3)+1 )
                  * ( ((vent->right-2)-(vent->left+2)+1)*2));
    if( num_reng < size )
    {
        escribe_linea_ceros(reng_act,num_reng,
                           posi_columna,vent,t_x,t_y,t_z,t_a,
                           t_b,t_c,t_d,t_e,t_f,0);
        if ( reng_act < lim_inf )
            ++reng_act;
        else
        {
            gettext(vent->left+2,vent->top+4,vent->right-
2,vent->bottom-2,buf3);
            puttext(vent->left+2,vent->top+3,vent->right-
2,vent->bottom-3,buf3);
            ++top;
        }
    }
    free(buf3);
}
}

```

```

if( teclaa == HOME )
{
    if( (top-tam_vent) >= 1 || reng_act != lim_sup )
    {
        reng_act = lim_sup;
        ren_temp = reng_act;
        top=1;
        i=4;
        for(num_reng = top ; num_reng < top + tam_vent;
            num_reng++ )
        {
            escribe_linea_ceros(reng_act,num_reng,
                posi_columna,vent,t_x,t_y,t_z,t_a,
                t_b,t_c,t_d,t_e,t_f,0);
            ++reng_act;
            ++i;
        }
        reng_act = ren_temp;
    }
}
if( teclaa == END )
{
    if((top + tam_vent)<= size || reng_act != lim_inf)
    {
        reng_act = lim_sup;
        top = size - tam_vent + 1;
        for(num_reng = top ; num_reng < top +tam_vent-1;
            num_reng++ )
        {
            escribe_linea_ceros(reng_act,num_reng,
                posi_columna,vent,t_x,t_y,t_z,t_a,
                t_b,t_c,t_d,t_e,t_f,0);
            ++reng_act;
            ++i;
        }
    }
}
if(teclaa == ENTER)
{
    t[0]=t_x;
    t[1]=t_y;
    t[2]=t_z;
    t[3]=t_a;
    t[4]=t_b;
    t[5]=t_c;
    t[6]=t_d;
    t[7]=t_e;
    t[8]=t_f;
    s=malloc(10);
    fmt=formato(vent,posi_columna);
    tam=tamano(fmt);
    sprintf(s,fmt,*(t[posi_columna-2]+num_reng));
    blank=malloc(tam);
    for(i=0;i<tam;i++) blank[i]=' ';
}

```

```

blank[tam]='\0';
if(strlen(s) >= tam) s[tam]='\0';
gotoxy( posicion(vent, posi_columna), reng_act );
textattr( atribut9 );
cputs( s );
gotoxy( posicion(vent, posi_columna), reng_act );
i=0;
do
{
    if ( !(c=getch()) ) c = getch() | 128 ;
    if( ( c == ESC ) )
    {
        textattr( atribut2 );
        gotoxy( posicion(vent, posi_columna), reng_act );
        cputs( s );
        free( blank );
        textattr( atribut4 );
        i=0;
        break;
    }
    if( ( c == ENTER ) && ( i == 0 ) )
    {
        textattr( atribut2 );
        gotoxy( posicion(vent, posi_columna), reng_act );
        cputs( s );
        free( blank );
        textattr( atribut4 );
        break;
    }
    if( c == ENTER ) break;
    if( c == 8 )
    {
        if( i > 0 )
        {
            --i;
            putch( c );
            s[i]='\0';
            putch( ' ' );
            putch( c );
        }
        else
            s[i]='\0';
    }
    else
    {
        if( isdigit( c ) || ( c == '.' ) || ( c == '-' ) )
        {
            s[i++]=c;
            putch( c );
        }
    }
} while( i < ( tam ) );
if( i > 0 )
{

```



```

        s[i]='\0';
        textattr(atribut2);
        gotoxy(posicion(vent, posi_columna), reng_act);
        cputs(s);
        *(t[posi_columna-2]+num_reng)=atof(s);
        textattr(atribut4);
        free(blank);
        free(fmt);
        free(s);
    }
}
}

/** ESTA FUNCION LLENA EL NUMERO DE LINEA CON CEROS **/
/** CON SOMBRA O SIN SOMBRA **/
void escribe_linea_ceros(reng_act, num_reng, posi_columna,
vent, t_xx, t_yy, t_zz, t_aa, t_bb, t_cc, t_dd, t_ee, t_ff,
aux_pasada)
int reng_act, num_reng, posi_columna;
int aux_pasada;
struct ventana *vent;
double
t_xx[], t_yy[], t_zz[], t_aa[], t_bb[], t_cc[], t_dd[], t_ee[],
t_ff[];
{
    int i, col_temp;
    char *line, *fmt;
    double *t[9];

    t[0] = t_xx;
    t[1] = t_yy;
    t[2] = t_zz;
    t[3] = t_aa;
    t[4] = t_bb;
    t[5] = t_cc;
    t[6] = t_dd;
    t[7] = t_ee;
    t[8] = t_ff;
    line = malloc(20);
    if( aux_pasada > 0 )
    {
        textattr(atribut2);
        sprintf(line, "%3d", num_reng);
        gotoxy(posicion(vent, 1), reng_act);
        cputs(line);
        textattr(atribut4);
    }
    else
    {
        sprintf(line, "%3d", num_reng);
        gotoxy(posicion(vent, 1), reng_act);
        cputs(line);
    }
}

```

```

col_temp = posi_columna;
for(posi_columna = 2;posi_columna < vent->columnas;
    posi_columna++)
{
    fmt = formato(vent, posi_columna);
    sprintf(line, fmt, *(t[ posi_columna - 2 ] +
        num_reng));
    free(fmt);
    gotoxy(posicion(vent, posi_columna), reng_act);
    cputs(line);
}
posi_columna = col_temp;
if (aux_pasada > 0 )
{
    fmt = formato(vent, posi_columna);
    sprintf(line, fmt, *(t[ posi_columna - 2 ] +
        num_reng));
    free(fmt);
    textattr(atribut2);
    gotoxy(posicion(vent, posi_columna), reng_act);
    cputs(line);
    textattr(atribut4);
}
free(line);
}

/* EN ESTA FUNCION DAMOS EL FOMATO DEL NUMERO A ESCRIBIR */
char *formato(vent, posi_columna)
struct ventana *vent;
int posi_columna ;
{
    char *pos;
    char *fmt;
    int letra, i;

    fmt = malloc(10);
    strcpy(fmt, "      ");
    switch(posi_columna)
    {
        case 2:
            pos = strchr(vent->formato, 'a');
            break;
        case 3:
            pos = strchr(vent->formato, 'b');
            break;
        case 4:
            pos = strchr(vent->formato, 'c');
            break;
        case 5:
            pos = strchr(vent->formato, 'd');
            break;
        case 6:
            pos = strchr(vent->formato, 'e');
            break;
    }
}

```

```

    case 7:
        pos = strchr(vent->formato, 'g');
        break;
    case 8:
        pos = strchr(vent->formato, 'h');
        break;
    case 9:
        pos = strchr(vent->formato, 'i');
        break;
    case 10:
        pos = strchr(vent->formato, 'j');
        break;
}
letra = (*pos++) + 1;
i = 0;
while((*pos) && *pos != letra)
    fmt[i++] = *pos++;
if (letra == 'f')
    fmt[i++] = 'f';
fmt[i] = '\0';
return (fmt);
}

/* NOS DA DONDE ES LA POSICION DE LOS CEROS A PINTAR */
int posicion(vent, posi_columna )
int posi_columna ;
struct ventana *vent;
{
    char *pos;
    char valor[3];

    strcpy(valor, " ");
    switch(posi_columna ) {
        case 1:
            pos = strchr(vent->posiciones, 'n');
            break;
        case 2:
            pos = strchr(vent->posiciones, 'x');
            break;
        case 3:
            pos = strchr(vent->posiciones, 'y');
            break;
        case 4:
            pos = strchr(vent->posiciones, 'z');
            break;
        case 5:
            pos = strchr(vent->posiciones, 'a');
            break;
        case 6:
            pos = strchr(vent->posiciones, 'b');
            break;
        case 7:
            pos = strchr(vent->posiciones, 'c');
            break;
    }
}

```

```

    case 8:
        pos = strchr(vent->posiciones, 'd');
        break;
    case 9:
        pos = strchr(vent->posiciones, 'e');
        break;
    case 10:
        pos = strchr(vent->posiciones, 'f');
        break;
}
strncpy(valor, pos+1, 2);
valor[2] = '\0';
return (atoi(valor));
}

/* DETERMINA EL TAMANO DE ENTREROS "8" QUE VA A TENER EL */
/* NUMERO */
int tamano(fmt)
char *fmt;
{
    int i;
    char *tam;

    tam = malloc(5);
    i = 0;
    while((*++fmt) && *fmt != '.')
        *(tam+i++) = *fmt;
    *(tam+i) = '\0';
    return (atoi(tam));
}

```

PREPROCESADOR INCLUYE AL ARCHIVO: IMPRIMIR.C

```

#include "libreria.h"
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <string.h>
#include <stdlib.h>
#include <io.h>
#include <fcntl.h>
#include <stat.h>
#include <dir.h>
#include <errno.h>

void imprimir(int dato_ini[]);
void arreglo_impreso(int size,struct ventana *vent,
double t_x[],double t_y[],double t_z[], double t_a[], double
t_b[],double t_c[], double t_d[], double t_e[], double
t_f[]);
void linea_impresa(int columna, struct ventana *vent,
double t_x[],double t_y[],double t_z[], double t_a[], double
t_b[],double t_c[], double t_d[], double t_e[], double
t_f[]);
char *formato(struct ventana *vent,int posi_columna);

/** IMPRESION DE DATOS EN MEMORIA **/
void imprimir(int dato_ini[])
{
if (dato_ini[0] > 0)
arreglo_impreso(dato_ini[0], &pun_nod, p_n_x_1,p_n_y_1,
p_n_z_2, p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,p_n_e_2,
p_n_f_2);
if (dato_ini[1] > 0)
{
elementos.subtitulo=" ELE. NOD_I NOD_J NOD_K NOD_L
NOD_M NOD_N NOD_O NOD_P MATERIAL ";
arreglo_impreso(dato_ini[1], &elementos, p_n_x_2,
p_n_y_2,p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
p_n_e_2,p_n_f_2);
}
if (dato_ini[2] > 0)
{
frontera.subtitulo=" No. DX DY ",
arreglo_impreso(dato_ini[2],&frontera,p_n_x_3,p_n_y_3,
p_n_z_2, p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,p_n_e_2,
p_n_f_2);
}
if (dato_ini[3] > 0)
{
materiales.subtitulo=" No. E v
Espesor",
arreglo_impreso(dato_ini[3], &materiales, p_n_x_4,
p_n_y_4,p_n_z_4,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
p_n_e_2,p_n_f_2);
}
}

```

```

if (dato_ini[4] > 0)
{
    nodos_cargados.subtitulo="  CARGA      NODO  ",
    arreglo_impreso(dato_ini[4], &nodos_cargados, p_n_x_5,
    p_n_y_2,p_n_z_2,p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
    p_n_e_2,p_n_f_2);
}
if (dato_ini[5] > 0)
{
    cargas.subtitulo="  No.      FX      FY  ",
    arreglo_impreso(dato_ini[5], &cargas, p_n_x_6,p_n_y_6,
    p_n_z_2, p_n_a_2,p_n_b_2,p_n_c_2,p_n_d_2,
    p_n_e_2,p_n_f_2);
}
fprintf(stdprn, "\f");
}

/* ARREGLO DE IMPRESION */
void arreglo_impreso(size, vent, t_x, t_y, t_z, t_a,
t_b,t_c,t_d,t_e,t_f)
int size;
struct ventana *vent;
double t_x[], t_y[], t_z[], t_a[],
t_b[],t_c[],t_d[],t_e[],t_f[];
{
    int i;
    int ren = 1;

    fprintf(stdprn,"%s\n\r", vent->titulo);
    fprintf(stdprn,"%s\n\r", vent->subtitulo);
    for (i = 1; i <= size; i++) {
        linea_impresa(i, vent, t_x, t_y, t_z, t_a, t_b,t_c,
        t_d,t_e,t_f);
        ++ren;
    }
    fprintf(stdprn, "\n\r");
}

/* LINEA DE IMPRESION */
void linea_impresa(linea, vent, t_x, t_y, t_z, t_a,t_b,t_c,
t_d,t_e,t_f)
int linea;
struct ventana *vent;
double t_x[], t_y[], t_z[], t_a[],
t_b[],t_c[],t_d[],t_e[],t_f[];
{
    int col_act;
    int i;
    char *line, *fmt;
    double *t[9];

    t[0] = t_x;
    t[1] = t_y;
    t[2] = t_z;

```

```
t[3] = t_a;
t[4] = t_b;
t[5] = t_c;
t[6] = t_d;
t[7] = t_e;
t[8] = t_f;
line = malloc(136);
sprintf(line, "%2d", linea);
fprintf(stdprn, " %s ", line);
for(col_act = 2; col_act < vent->columnas; col_act++)
{
    fmt = formato(vent, col_act);
    sprintf(line, fmt, *(t[ col_act - 2 ] + linea));
    if( vent->columnas == 4 )
        fprintf(stdprn, "%s      ", line);
    if( vent->columnas == 11 )
        fprintf(stdprn, " %s ", line);
    if( vent->columnas == 5 )
        fprintf(stdprn, " %s      ", line);
    if( vent->columnas == 3 )
        fprintf(stdprn, " %s      ", line);
    free(fmt);
}
fprintf(stdprn, "\n\r");
free(line);
}
```

PREPROCESADOR INCLUYE LA LIBRERIA DEFINIDA POR EL USUARIO:

LIBRERIA.H

```
#define atribut1 WHITE + (BLUE << 4)
#define atribut2 BLACK + (CYAN << 4)
#define atribut3 YELLOW + (BLUE << 4)
#define atribut4 BLACK + (LIGHTGRAY << 4)
#define atribut5 YELLOW + (LIGHTGRAY << 4)
#define atribut6 WHITE + (BLACK << 4)
#define atribut7 BLACK + (BLACK << 4)
#define atribut8 YELLOW + (GREEN << 4)
#define atribut9 WHITE + (BLUE << 4)
#define atribut10 RED + (LIGHTGRAY << 4)
#define atribut11 YELLOW + (BLACK << 4)
#define atribut12 YELLOW + (BLUE << 4)
#define atribut13 WHITE + (CYAN << 4)
#define atribut14 GREEN + (BLUE << 4)
#define atribut15 WHITE + (RED << 4)
#define atribut16 BLACK + (WHITE << 4)
#define atribut17 WHITE + (GREEN << 4)
#define atribut18 CYAN + (BLUE << 4)
#define atribut19 GREEN + (CYAN << 4)
#define atribut20 BLACK + (BLUE << 4)
#define atribut21 RED + (BLUE << 4)
#define atribut22 LIGHTBLUE + (BLUE << 4)
#define atribut23 LIGHTGREEN + (BLUE << 4)
#define ENTER 13
#define ESC 27
#define FLEADEL 205
#define FLEATRA 203
#define FLEARRI 200
#define FLEABAJ 208
#define ESQSUIZQ 201
#define ESQINIZQ 200
#define ESQSUDER 187
#define ESQINDER 188
#define LADOIZDE 186
#define LADOARAB 205
#define TABIQUE 177
#define ESPACIO 32
#define HOME 199
#define PGUP 201
#define PGDN 209
#define END 207
#define INS 210
#define DEL 211
#define NULL 0
```

```
struct ventana {
    int left;
    int top;
    int right;
    int bottom;
    char *titulo;
    char *subtitulo;
```



```

int columnas;
char *formato;
char *posiciones;
};

extern int bloque;
extern int cursor_x, cursor_y;
extern char *men_prin[] ;
extern char *car_prin[] ;
extern char *men_arch[] ;
extern char *car_arch[] ;
extern char *men_dato[] ;
extern char *men_carga[] ;
extern char *car_dato[] ;
extern char *car_carga[] ;
extern int teclprin[] ;
extern int letrprin[] ;
extern int teclarch[] ;
extern int letrarch[] ;
extern int tecldato[] ;
extern int letrdato[] ;
extern int teclcarga[] ;
extern int letrcarga[] ;
extern struct ventana pun_nod ;
extern struct ventana elementos ;
extern struct ventana frontera ;
extern struct ventana materiales ;
extern struct ventana nodos_cargados ;
extern struct ventana cargas ;
extern double p_n_x_1[100], p_n_y_1[100];
extern double p_n_x_2[100], p_n_y_2[100], p_n_z_2[100],
p_n_a_2[100], p_n_b_2[100], p_n_c_2[100], p_n_d_2[100],
p_n_e_2[100], p_n_f_2[100];
extern double p_n_x_3[100], p_n_y_3[100];
extern double p_n_x_4[100], p_n_y_4[100], p_n_z_4[100];
extern double p_n_x_5[100];
extern double p_n_x_6[100], p_n_y_6[100];
extern int top, lim_sup, lim_inf, tam_vent;
extern int reng_act, num_reng;
extern int posi_columna;
extern int n1, n2, n3, n4, n5, n6;
extern char *guarda[100];
extern int tamven_1, tamven_2;
extern char nombre_archivo[14];
extern char intento[80];
extern char path_archivo[80];
extern char aux_path[100];
extern char aux_path[80];
extern char cad_temp[80];
extern int columna;
extern int pri_mouse[] ;
extern int sec_mouse[] ;
extern int sec_1_mouse[] ;
extern int ter_mouse[] ;

```

```
extern int coord_archivos[] ;
extern int coord_directorio[] ;
extern int coord_puntos_nodales[] ;
extern int coord_elementos[] ;
extern int coord_frontera[] ;
extern int coord_materiales[] ;
extern int coord_nodos_cargados[] ;
extern int temp[] ;
extern int temp1[] ;
extern int temp2[] ;
extern dato_ini[] ;
```

```

PROCESADOR INCLUYE AL ARCHIVO: PROCESA.FOR
C   PROGRAMADOR: SERGIO GONZALEZ CASTAÑEDA.
C   ASESORADO POR: ING. Ma. DE LA GRACIA BARQUERO DIAZ
C   BARRIGA.
C   PROGRAMA ELEMENTO FINITO DE 8 PUNTOS
C   PROGRAMA PRINCIPAL
C
PROGRAM ELEFIN8
DIMENSION A(18000)
CHARACTER NOMBRE1*12, DRI1*1, NOMAR1*15
CHARACTER RESULTA*12, DRI2*1, NOMAR2*15
CHARACTER AUXILIAR*12, DRI3*1, NOMAR3*15
NA=18000
CALL PRESENTA
C   ABRIR EL ARCHIVO DE DATOS
CALL ABRIRA(NOMBRE1, DRI1, NOMAR1, 9, 2)
C   LECTURA DE DATOS INICIALES
CALL DATINIC(NPUN, NELEM, NMAT, NNOD, NCARGA, 9)
C   ABRIR UN ARCHIVO PARA IMPRIMIR RESULTADOS
CALL ABRIRA(RESULTA, DRI2, NOMAR2, 8, 1)
C   OPEN (8, FILE= 'RESULTA', STATUS= 'NEW')
C   ABRIR ARCHIVO PARA GUARDAR IE, DB, PG
CALL ABRIRA(AUXILIAR, DRI3, NOMAR3, 7, 1)
C   OPEN (7, FILE= 'AYUDA', STATUS= 'NEW')
C   DIMENSIONAMIENTO PARA MEMORIA DINAMICA
N1=1
C   COORD. 2-X, 3-Y
N2=N1+NPUN
N3=N2+NPUN
C   INDICADOR DE ECUACION
N4=N3+2*NPUN
C   MATERIALES
N5=N4+4*NMAT
C   NODOS (6-i, 7-j, 8-k, 9-l, 10-m, 11-n, 12-o, 13-p)
N6=N5+NELEM
N7=N6+NELEM
N8=N7+NELEM
N9=N8+NELEM
N10=N9+NELEM
N11=N10+NELEM
N12=N11+NELEM
N13=N12+NELEM
C   TIPO DE MATERIAL
N14=N13+NELEM
C   CARGAS (15-NODO CARGADO, 16-CARGA x, 17 CARGA y)
N15=N14+NCARGA
N16=N15+NCARGA
N17=N16+NCARGA
C   MATRIZ DE RIGIDEZ
NT=N17-1
C   REvisa EL DIMENSIONAMIENTO
CALL REvisa(NA, NT)
C   LECTURA DE DATOS
CALL LECAR(A, N1, N17, NT, 9)

```

```

C      CERRAR ARCHIVO
      CALL FINAR(9)
C      CALCULO DEL INDICADOR DE ECUACION
      CALL IND (A(N3), NPUN, 2, NC, NP, NN)
      N18=N17+NC*NC
      N19=N18+NC
      NT=N19-1
      WRITE (*, 6100) NC
C      REvisa EL DIMENSIONAMIENTO
      CALL REvisa (NA, NT)
C      CALCULO DE LA MATRIZ DE RIGIDECES DE LA ESTRUCTURA
      CALL MATK8 (A(N1), A(N2), A(N3), A(N4), A(N5), A(N6), A(N7)
*      , A(N8), A(N9), A(N10), A(N11), A(N12), A(N13), NPUN, NELEM,
*      *NMAT, NC, A(N17))
C      CALCULO DEL VECTOR DE CARGAS
      CALL FUERZA (A(N18), A(N3), A(N14), A(N15), A(N16),
*      *NPUN, NCARGA, NC)
C      CIERRA EL ARCHIVO GUARDA IE, DB, PG
      CLOSE (7)
C      CALCULO DE LOS DESPLAZAMIENTOS
      CALL CROUT(A(N17), A(N18), NC)
C      SE ABRE EL ARCHIVO QUE GUARDA IE, DB, PG
C      CALL ABRIRA (AUXILIAR, DRI3, NOMAR3, 7, 2)
      OPEN (7, FILE=AUXILIAR, STATUS='OLD')
C      CALCULO DE ESFUERZOS
      CALL ESFUER8 (A(N18), NC, NELEM, IE)
C      CIERRA EL ARCHIVO DE IMPRESION DE RESULTADOS
      CLOSE (8)
C      CIERRA EL ARCHIVO QUE GUARDA IE, DB, PG
      CLOSE (7)
6100  FORMAT(/, 2X, 'NC = ', I10)
      WRITE(*, *)
      WRITE(*, *) ' F I N           D E L           P R O C E S O '
      END

      SUBROUTINE PRESENTA
      WRITE(*, *)
      WRITE(*, *)
      WRITE(*, *)
      WRITE(*, *) ' U N I V E R S I D A D      N A C
$ I O N A L A U T O N O M A '
      WRITE(*, *) '
      WRITE(*, *) '
      WRITE(*, *) '
      $ C O '
      WRITE(*, *)
      WRITE(*, *)
      WRITE(*, *) '
      $ A C A T L A N '
      WRITE(*, *)
      WRITE(*, *)
      WRITE(*, *) '
      $ A D O R '
      WRITE(*, *)

```



```

SUBROUTINE ABRIRA(AUX_NOMBRE,AUX_DRI,AUX_NOMAR,
                 AUX_NAR,AUX_IND)
INTEGER AUX_IND,AUX_NAR
CHARACTER AUX_NOMBRE*12,AUX_DRI*1,AUX_NOMAR*15
IF( AUX_NAR .EQ. 9 ) THEN
WRITE(*,*) ' ARCHIVO DE DATOS *.FMT : '
WRITE(*,7000)
READ(*,7100)AUX_DRI
WRITE(*,7200)
READ(*,7300)AUX_NOMBRE
AUX_NOMAR(1:1)=AUX_DRI
AUX_NOMAR(2:1+1)=CHAR(58)
AUX_NOMAR(3:2+12)=AUX_NOMBRE
WRITE(*,7400)AUX_NOMAR
7400  FORMAT(2X,'ABRIENDO EL ARCHIVO: ',A15)
IF (AUX_IND.EQ.1)THEN
OPEN(AUX_NAR,FILE=AUX_NOMAR,STATUS='NEW')
ELSE
OPEN(AUX_NAR,FILE=AUX_NOMAR,STATUS='OLD')
ENDIF
7100  FORMAT(A1)
7300  FORMAT(A12)
7000  FORMAT(2X,'DRIVE DONDE SE ENCUENTRA EL ARCHIVO DE
$DATOS (A O B O C):')
7200  FORMAT(2X,'NOMBRE DEL ARCHIVO MAXIMO 12 LETRAS CON O
$SIN EXTENSION:')
ENDIF
IF( AUX_NAR .EQ. 8 ) THEN
WRITE(*,*) ' ARCHIVO DE RESULTADOS *.* : '
WRITE(*,8000)
READ(*,8100)AUX_DRI
WRITE(*,8200)
READ(*,8300)AUX_NOMBRE
AUX_NOMAR(1:1)=AUX_DRI
AUX_NOMAR(2:1+1)=CHAR(58)
AUX_NOMAR(3:2+12)=AUX_NOMBRE
WRITE(*,8400)AUX_NOMAR
8400  FORMAT(2X,'ABRIENDO EL ARCHIVO: ',A15)
IF (AUX_IND.EQ.1)THEN
OPEN(AUX_NAR,FILE=AUX_NOMAR,STATUS='NEW')
ELSE
OPEN(AUX_NAR,FILE=AUX_NOMAR,STATUS='OLD')
ENDIF
8100  FORMAT(A1)
8300  FORMAT(A12)
8000  FORMAT(2X,'DRIVE DONDE SE GUARDARA EL ARCHIVO DE
$RESULTADOS (A O B O C):')
8200  FORMAT(2X,'NOMBRE DEL ARCHIVO MAXIMO 12 LETRAS CON O
$SIN EXTENSION:')
ENDIF
IF( AUX_NAR .EQ. 7 ) THEN
WRITE(*,*) ' ARCHIVO AUXILIAR *.* : '
WRITE(*,9000)

```

```

READ(*,9100)AUX_DRI
WRITE(*,9200)
READ(*,9300)AUX_NOMBRE
AUX_NOMAR(1:1)=AUX_DRI
AUX_NOMAR(2:1+1)=CHAR(58)
AUX_NOMAR(3:2+12)=AUX_NOMBRE
WRITE(*,9400)AUX_NOMAR
9400  FORMAT(2X'ABRIENDO EL ARCHIVO: ',A15)
      IF (AUX_IND.EQ.1)THEN
      OPEN(AUX_NAR,FILE=AUX_NOMAR,STATUS='NEW')
      ELSE
      OPEN(AUX_NAR,FILE=AUX_NOMAR,STATUS='OLD')
      ENDIF
9100  FORMAT(A1)
9300  FORMAT(A12)
9000  FORMAT(2X,'DRIVE DONDE SE GUARDARA EL ARCHIVO
$AUXILIAR (A ó B ó C):')
9200  FORMAT(2X,'NOMBRE DEL ARCHIVO MAXIMO 12 LETRAS CON O
$$SIN EXTENSION:')
      ENDIF
      RETURN
      END

```

```

SUBROUTINE DATINIC(NPUN,NELEM,NMAT,NNOD,NCARGA,NAR)
READ(NAR,*)NPUN
READ(NAR,*)NELEM
READ(NAR,*)NMAT
READ(NAR,*)NNOD
READ(NAR,*)NCARGA
RETURN
END

```

```

SUBROUTINE REVISA(NA,NT)
IF(NA.LT.NT)THEN
WRITE(*,6000)NA,NT
STOP
ELSE
WRITE(*,*)
WRITE(*,*)'INFORMACION : '
WRITE(*,6000)NA,NT
WRITE(*,*)
WRITE(*,*)' PRESIONE RETURN PARA PODER CONTINUAR... '
READ(*,5000)NE
ENDIF
5000  FORMAT(I1)
6000  FORMAT(/,2X,'DIMENSIONAMIENTO DE A= ',I10,
*,2X,'DIMENSIONAMIENTO USADO = ',I10)
RETURN
END

```

```

SUBROUTINE LECAR(A,NI,NF,NT,NAR)
DIMENSION A(NT)
DO 100 I=NI,NF-1
READ(NAR,*)A(I)

```

```

100 CONTINUE
RETURN
END

SUBROUTINE FINAR (NAR)
CLOSE (NAR)
RETURN
END

SUBROUTINE IND (XIND, NNOD, NGL, NC, NP, NN)
DIMENSION XIND (NGL, NNOD)
NC=0
NN=0
NP=0
DO 500 J=1, NNOD
DO 400 I=1, NGL
IN= INT (XIND (I, J))
IF (IN) 100, 200, 300
100 CONTINUE
NP=NP+1
XIND (I, J) =NP
GOTO 400
200 CONTINUE
NC=NC+1
XIND (I, J) =NC
GOTO 400
300 CONTINUE
NN=NN+1
XIND (I, J) =0
400 CONTINUE
500 CONTINUE
RETURN
END

SUBROUTINE MATK8 (XX, YY, XIND, XMAT1, XIEL,
XJEL, XKEL, XLEL, XMEL, XNEL, XOEL, XPEL,
XIMAT, NPUN, NELEM, NMAT, NC, RK)
C MATRIZ DE RIGIDECES GLOBAL
DIMENSION XX (NPUN), YY (NPUN), XIND (2, NPUN)
*, XMAT1 (NMAT, 4), XIEL (NELEM), XJEL (NELEM),
XKEL (NELEM), XLEL (NELEM), XMEL (NELEM),
*, XNEL (NELEM), XOEL (NELEM), XPEL (NELEM)
*, XIMAT (NELEM), RG (16, 16), X (8), Y (8), IE (16), RK (NC, NC)
DO L=1, NELEM
C SE IMPRIME UN LETRERO DE NUMERO DE ELEMENTO
WRITE (8, *) 'ELEMENTO NO. '
WRITE (8, *) L
X (1) =XX (INT (XIEL (L)))
X (2) =XX (INT (XJEL (L)))
X (3) =XX (INT (XKEL (L)))
X (4) =XX (INT (XLEL (L)))
X (5) =XX (INT (XMEL (L)))
X (6) =XX (INT (XNEL (L)))
X (7) =XX (INT (XOEL (L)))

```



```

X(8)=XX(INT(XPEL(L)))
Y(1)=YY(INT(XIEL(L)))
Y(2)=YY(INT(XJEL(L)))
Y(3)=YY(INT(XKEL(L)))
Y(4)=YY(INT(XLEL(L)))
Y(5)=YY(INT(XMEL(L)))
Y(6)=YY(INT(XNEL(L)))
Y(7)=YY(INT(XOEL(L)))
Y(8)=YY(INT(XPEL(L)))
C
MATERIALES
E=XMAT1(INT(XIMAT(L)),2)
XNU=XMAT1(INT(XIMAT(L)),3)
C
AQUI FALTA DEFINIR EL ESPESOR
T=.30
C
INDICADOR DE ECUACION
IE(1)=INT(XIND(1,INT(XIEL(L))))
IE(2)=INT(XIND(2,INT(XIEL(L))))
IE(3)=INT(XIND(1,INT(XJEL(L))))
IE(4)=INT(XIND(2,INT(XJEL(L))))
IE(5)=INT(XIND(1,INT(XKEL(L))))
IE(6)=INT(XIND(2,INT(XKEL(L))))
IE(7)=INT(XIND(1,INT(XLEL(L))))
IE(8)=INT(XIND(2,INT(XLEL(L))))
IE(9)=INT(XIND(1,INT(XMEL(L))))
IE(10)=INT(XIND(2,INT(XMEL(L))))
IE(11)=INT(XIND(1,INT(XNEL(L))))
IE(12)=INT(XIND(2,INT(XNEL(L))))
IE(13)=INT(XIND(1,INT(XOEL(L))))
IE(14)=INT(XIND(2,INT(XOEL(L))))
IE(15)=INT(XIND(1,INT(XPEL(L))))
IE(16)=INT(XIND(2,INT(XPEL(L))))
C
GUARDAR EN UN ARCHIVO "IND"
DO LL=1,16
WRITE(7,*)IE(LL)
ENDDO
C
CALCULA LA MATRIZ DE RIGIDECES DE CADA ELEMENTO
CALL MATKE8(E,XNU,X,Y,RG,T)
C
SE IMPRIME EL RESULTADO DE LA MATRIZ RIGIDECES
CALL IMATR2(RG,16,16)
C
SE ENSAMBLA LA MATRIZ DE RIGIDECES DEL ELEMENTO
C
CON LA MATRIZ DE RIGIDECES DE LA ESTRUCTURA
CALL ENSAM(RK,RG,IE,NC,16)
END DO
C
SE IMPRIME LA MATRIZ DE RIGIDECES DE LA ESTRUCTURA
WRITE(8,*)'MATRIZ DE RIGIDECES DE LA ESTRUCTURA'
CALL IMATR2(RK,NC,NC)
RETURN
END

SUBROUTINE ENSAM(RK,RKE,IE,NC,NGLEL)
DIMENSION RK(NC,NC),RKE(NGLEL,NGLEL),IE(NGLEL)
DO 200 I=1,NGLEL
IF(IE(I).LE.0)GOTO 200
DO 100 J=1,NGLEL

```

```

      IF (IE(J).LE.0) GOTO 100
      IR=IE(I)
      JR=IE(J)
      RK(IR, JR)=RK(IR, JR)+RKE(I, J)
100   CONTINUE
200   CONTINUE
      RETURN
      END

      SUBROUTINE MATKE8 (E,XNU,X,Y, RG,T)
C     MATRIZ DE RIGIDECES DE CADA ELEMENTO
      DIMENSION H(2), SI(4), ANU(4), D(3,3), B(3,16)
*, BT(16,3), RIG(16,16), DB(3,16), X(8), Y(8), RG(16,16)
*, PGX(4), PGY(4)
      H(1)=1.0
      H(2)=1.0
      SI(1)=-0.57735026918963
      SI(2)=-SI(1)
      SI(3)=-SI(1)
      SI(4)=SI(1)
      ANU(1)=SI(1)
      ANU(2)=SI(1)
      ANU(3)=-SI(1)
      ANU(4)=-SI(1)
      IG=0
      M=2
      N=2
C     INICIALIZANDO
      DO I=1,16
      DO J=1,16
      RIG(I,J)=0.0
      RG(I,J)=0.0
      END DO
      END DO
      DO I=1,M
      DO J=1,N
      IG=IG+1
      SA=SI(IG)
      EA=ANU(IG)
C     CALL MATB8 (SA,EA,X,Y,DET,B,IG,PGX,PGY)
C     IMPRESION DE LA MATRIZ B
      WRITE(8,*) 'MATRIZ B'
      CALL IMATR2 (B,3,16)
      CALL CD (D,E,XNU)
      CALL MULMAT (D,B,DB,3,3,3,16)
C     GUARDAR EN UN ARCHIVO EL PUNTO GAUSSIANO Y DB
      DO I2=1,3
      DO J2=1,16
      WRITE(7,*)DB(I2,J2)
      ENDDO
      ENDDO
      WRITE(7,*) PGX(IG)
      WRITE(7,*) PGY(IG)
      CALL MATRAN (B,BT,3,16)

```

```

CALL MULMAT (BT,DB,RIG,16,3,3,16)
DO I1=1,16
DO J1=1,16
RG(I1,J1)=RG(I1,J1)+(T*DET*RIG(I1,J1))
END DO
END DO
END DO
END DO
RETURN
END

SUBROUTINE IMATR2(A, NR, NC)
DIMENSION A(NR, NC)
DO 100 L=1,NC,5
M = L + 4
IF (M.GT.NC) M=NC
WRITE(8,6000)(K,K=L,M)
DO 100 I=1,NR
WRITE(8,6010)I,(A(I,J),J=L,M)
100 CONTINUE
RETURN
6000 FORMAT(/,9X,I4,4(10X,I4))
6010 FORMAT(1X,I4,5(1PE14.6))
RETURN
END

SUBROUTINE FUERZA (F,XIND,XNODCA,CARX,CARY
*,NPUN,NCARGA,NC)
DIMENSION F(NC),XIND(2,NPUN),XNODCA(NCARGA)
*,CARX(NCARGA),CARY(NCARGA)
C INICIALIZAR
DO I=1,NC
F(I)=0
END DO
IF (NCARGA.EQ.0)GOTO 300
DO I=1,NCARGA
IND1=INT(XIND(1,INT(XNODCA(I))))
IND2=INT(XIND(2,INT(XNODCA(I))))
IF (IND1.EQ.0) GOTO 100
F(IND1)=F(IND1)+CARX(I)
100 IF (IND2.EQ.0) GOTO 200
F(IND2)=F(IND2)+CARY(I)
200 CONTINUE
END DO
300 CONTINUE
C IMPRESION EN ARCHIVO DEL VECTOR DE CARGA
WRITE(8,*)'VECTOR DE CARGA'
CALL IMATR2 (F,NC,1)
RETURN
END

SUBROUTINE CROUT(A,B,N)
DIMENSION A(N,N),B(N)
A(1,2)=A(1,2)/A(1,1)

```

```

A(2,2)=A(2,2)-A(1,1)*A(1,2)**2
IF(N.EQ.2) GOTO 50
DO J=3,N
  IS=J-1
  A(1,J)=A(1,J)/A(1,1)
  DO I=2, IS
    S=0
    KS=I-1
    DO K=1, KS
      S=S+A(K, I)*A(K, K)*A(K, J)
    END DO
    A(I, J)=(A(I, J)-S)/A(I, I)
  END DO
  S1=0
  DO K=1, IS
    S1=S1+A(K, K)*A(K, J)**2
  END DO
  A(J, J)=A(J, J)-S1
END DO
C  PROGRAMA QUE HACE LAS SUSTITUCIONES PARA EL
C  METODO DE GAUSS-CROUT
DO I=2,N
  KS=I-1
  S=0
  DO K=1, KS
    S=S+A(K, I)*B(K)
  END DO
  B(I)=B(I)-S
END DO
B(N)=B(N)/A(N, N)
IS=N-1
DO I=1, IS
  IA=N-I
  KI=IA+1
  S=0
  DO K=KI, N
    S=S+A(IA, K)*B(K)
  END DO
  B(IA)=B(IA)/A(IA, IA)-S
END DO
CALL IMATR2(B, N, 1)
RETURN
50  END

SUBROUTINE ESFUERS(F, NC, NELEM, IE)
DIMENSION DB(3, 16), DES(16), IE(16), F(NC), ESF(3)
C  INICIALIZAR
DO I= 1, NELEM
C  LECTURA DEL ARCHIVO DE LA INFORMACION
C  LECTURA DEL INDICADOR DE ECUACION
DO II=1, 16
  READ(7, *) IE(II)
ENDDO
C  CALCULO DEL VECTOR DE DESPLAZAMIENTO PARA

```

```

C      CADA ELEMENTO
      DO J=1,16
          IF (IE(J).EQ.0) THEN
              DES(J)=0.0
          ELSE
              DES(J)=F(IE(J))
          ENDIF
      ENDDO
C      LETRERO DEL ARCHIVO DE DATOS DEL ELEMENTO
      WRITE(8,*)'NUMERO DE ELEMENTO'
      WRITE(8,*)I
      DO JJ=1,4
C      SE CALCULAN LOS ESFUERZOS PARA 4 PUNTOS
C      GAUSSIANOS POR ELEMENTO
C      LECTURA DEL ARCHIVO DB
      DO IJJ=1,3
      DO JJJ=1,16
      READ(7,*)DB(IJJ,JJJ)
      ENDDO
      ENDDO
C      LECTURA DE LOS PUNTOS GAUSSIANOS
      READ(7,*)PGX
      READ(7,*)PGY
      CALL MULMAT (DB,DES,ESF,3,16,16,1)
C      ARCHIVO PARA GUARDAR LOS ESFUERZOS
C      Y PUNTOS GAUSSIANOS
      WRITE(8,*)'PUNTO GAUSSIANO X'
      WRITE(8,*)PGX
      WRITE(8,*)'PUNTO GAUSSIANO Y'
      WRITE(8,*)PGY
      WRITE(8,*)'ESFUERZOS'
      CALL IMATR2 (ESF,3,1)
      ENDDO
      ENDDO
      RETURN
      END

```

```

SUBROUTINE MULMAT (A,B,C,NRA,NCA,NRB,NCB)
DIMENSION A(NRA,NCA),B(NRB,NCB),C(NRA,NCB)
DO 300 I=1,NRA
DO 200 J=1,NCB
XX=0.0
DO 100 K=1,NCA
XX=XX+A(I,K)*B(K,J)
100 CONTINUE
C(I,J)=XX
200 CONTINUE
300 CONTINUE
RETURN
END

```

```

SUBROUTINE MATB8 (SA,EA,X,Y,DET,B,IG,PGX,PGY)
DIMENSION FN(8),DFNS(8),DFNN(8),X(8),Y(8),DFNX(8),
*DFNY(8),B(3,16),PGX(4),PGY(4)

```

C FUNCIONES DE FORMA
 $FN(1) = 0.25 * (1-SA) * (1-EA) * (-SA-EA-1)$
 $FN(2) = 0.50 * (1-SA**2) * (1-EA)$
 $FN(3) = 0.25 * (1+SA) * (1-EA) * (SA-EA-1)$
 $FN(4) = 0.50 * (1+SA) * (1-EA**2)$
 $FN(5) = 0.25 * (1+SA) * (1+EA) * (SA+EA-1)$
 $FN(6) = 0.50 * (1-SA**2) * (1+EA)$
 $FN(7) = 0.25 * (1-SA) * (1+EA) * (-SA+EA-1)$
 $FN(8) = 0.5 * (1-SA) * (1-EA**2)$

C DERIVADAS DE LA FUNCIONES DE FORMA EN REF. LOCAL
 $DFNS(1) = 0.25 * (1-EA) * (EA+2*SA)$
 $DFNS(2) = SA * (EA-1)$
 $DFNS(3) = 0.25 * (1-EA) * (2*SA-EA)$
 $DFNS(4) = 0.5 * (1-EA**2)$
 $DFNS(5) = 0.25 * (1+EA) * (2*SA+EA)$
 $DFNS(6) = -SA * (EA+1)$
 $DFNS(7) = 0.25 * (1+EA) * (2*SA-EA)$
 $DFNS(8) = 0.5 * (EA**2-1)$
 $DFNN(1) = 0.25 * (1-SA) * (SA+2*EA)$
 $DFNN(2) = 0.5 * (SA**2-1)$
 $DFNN(3) = 0.25 * (1+SA) * (2*EA-SA)$
 $DFNN(4) = -EA * (1+SA)$
 $DFNN(5) = 0.25 * (1+SA) * (2*EA+SA)$
 $DFNN(6) = 0.5 * (1-SA**2)$
 $DFNN(7) = 0.25 * (1-SA) * (2*EA-SA)$
 $DFNN(8) = -EA * (1-SA)$

C PUNTOS GAUSSIANOS
 $PGX(IG) = FN(1) * X(1) + FN(2) * X(2) + FN(3) * X(3) + FN(4) * X(4)$
 $+ FN(5) * X(5) + FN(6) * X(6) + FN(7) * X(7) + FN(8) * X(8)$
 $PGY(IG) = FN(1) * Y(1) + FN(2) * Y(2) + FN(3) * Y(3) + FN(4) * Y(4)$
 $+ FN(5) * Y(5) + FN(6) * Y(6) + FN(7) * Y(7) + FN(8) * Y(8)$

C DERIVADAS DE LAS CORD. GLOBALES EN TERMINOS DE LAS
 C LOCALES.
 $DXS = DFNS(1) * X(1) + DFNS(2) * X(2) + DFNS(3) * X(3) +$
 $DFNS(4) * X(4) + DFNS(5) * X(5) + DFNS(6) * X(6) +$
 $DFNS(7) * X(7) + DFNS(8) * X(8)$
 $DXN = DFNN(1) * X(1) + DFNN(2) * X(2) + DFNN(3) * X(3) +$
 $DFNN(4) * X(4) + DFNN(5) * X(5) + DFNN(6) * X(6) +$
 $DFNN(7) * X(7) + DFNN(8) * X(8)$
 $DYS = DFNS(1) * Y(1) + DFNS(2) * Y(2) + DFNS(3) * Y(3) +$
 $DFNS(4) * Y(4) + DFNS(5) * Y(5) + DFNS(6) * Y(6) +$
 $DFNS(7) * Y(7) + DFNS(8) * Y(8)$
 $DYN = DFNN(1) * Y(1) + DFNN(2) * Y(2) + DFNN(3) * Y(3) +$
 $DFNN(4) * Y(4) + DFNN(5) * Y(5) + DFNN(6) * Y(6) +$
 $DFNN(7) * Y(7) + DFNN(8) * Y(8)$
 $DET = DXS * DYN - DYS * DXN$

C DERIVADAS DE LAS FUNCIONES CON RESPECTO A X y Y
 $DFNX(1) = (DYN * DFNS(1) - DYS * DFNN(1)) / DET$
 $DFNX(2) = (DYN * DFNS(2) - DYS * DFNN(2)) / DET$
 $DFNX(3) = (DYN * DFNS(3) - DYS * DFNN(3)) / DET$
 $DFNX(4) = (DYN * DFNS(4) - DYS * DFNN(4)) / DET$
 $DFNX(5) = (DYN * DFNS(5) - DYS * DFNN(5)) / DET$
 $DFNX(6) = (DYN * DFNS(6) - DYS * DFNN(6)) / DET$
 $DFNX(7) = (DYN * DFNS(7) - DYS * DFNN(7)) / DET$

```

DFNX (8) = (DYN*DFNS (8) -DYS*DFNN (8) ) /DET
DFNY (1) = (-DXN*DFNS (1) +DXS*DFNN (1) ) /DET
DFNY (2) = (-DXN*DFNS (2) +DXS*DFNN (2) ) /DET
DFNY (3) = (-DXN*DFNS (3) +DXS*DFNN (3) ) /DET
DFNY (4) = (-DXN*DFNS (4) +DXS*DFNN (4) ) /DET
DFNY (5) = (-DXN*DFNS (5) +DXS*DFNN (5) ) /DET
DFNY (6) = (-DXN*DFNS (6) +DXS*DFNN (6) ) /DET
DFNY (7) = (-DXN*DFNS (7) +DXS*DFNN (7) ) /DET
DFNY (8) = (-DXN*DFNS (8) +DXS*DFNN (8) ) /DET

```

C

```

CALCULO DE B
B (1, 1) =DFNX (1)
B (1, 3) =DFNX (2)
B (1, 5) =DFNX (3)
B (1, 7) =DFNX (4)
B (1, 9) =DFNX (5)
B (1, 11) =DFNX (6)
B (1, 13) =DFNX (7)
B (1, 15) =DFNX (8)
B (2, 2) =DFNY (1)
B (2, 4) =DFNY (2)
B (2, 6) =DFNY (3)
B (2, 8) =DFNY (4)
B (2, 10) =DFNY (5)
B (2, 12) =DFNY (6)
B (2, 14) =DFNY (7)
B (2, 16) =DFNY (8)
B (3, 1) =DFNY (1)
B (3, 2) =DFNX (1)
B (3, 3) =DFNY (2)
B (3, 4) =DFNX (2)
B (3, 5) =DFNY (3)
B (3, 6) =DFNX (3)
B (3, 7) =DFNY (4)
B (3, 8) =DFNX (4)
B (3, 9) =DFNY (5)
B (3, 10) =DFNX (5)
B (3, 11) =DFNY (6)
B (3, 12) =DFNX (6)
B (3, 13) =DFNY (7)
B (3, 14) =DFNX (7)
B (3, 15) =DFNY (8)
B (3, 16) =DFNX (8)
RETURN
END

```

```

SUBROUTINE CD (D, E, XNU)
DIMENSION D (3, 3)
T=E/(1-XNU**2)
D (1, 1) =T
D (1, 2) =XNU*T
D (1, 3) =0
D (2, 1) =XNU*T
D (2, 2) =T
D (2, 3) =0

```

```
D(3,1)=0
D(3,2)=0
D(3,3)=(1-XNU)*T/2
RETURN
END
```

```
SUBROUTINE MATRAN (A,AT,NR,NC)
DIMENSION A(NR,NC),AT(NC,NR)
DO I=1,NR
DO J=1,NC
AT(J,I)=A(I,J)
END DO
END DO
RETURN
END
```

```
SUBROUTINE IMATR1(A, NR, NC)
DIMENSION A(NR, NC)
DO 100 L=1,NC,5
M = L + 4
IF (M.GT.NC) M=NC
WRITE(*,6000) (K,K=L,M)
DO 100 I=1,NR
WRITE(*,6010) I, (A(I,J),J=L,M)
100 CONTINUE
RETURN
6000 FORMAT(/,9X,I4,4(10X,I4))
6010 FORMAT(1X,I4,5(1PE14.6))
RETURN
END
```


A P E N D I C E B
LISTADO DE ARCHIVOS
GENERADOS
POR EL :
-PREPROCESADOR.
-PROCESADOR.

ARCHIVO B.1. DATOS PARA CONSULTA EN PANTALLA:
 A:\EJEMPLO1.CNS
 GENERADO POR EL PREPROCESADOR.

PUNTOS NODALES		
No.	X	Y
1	8.00	1.00
2	8.00	0.50
3	8.00	0.00
4	7.00	1.00
5	7.00	0.00
6	6.00	1.00
7	6.00	0.50
8	6.00	0.00
9	5.00	1.00
10	5.00	0.00
11	4.00	1.00
12	4.00	0.50
13	4.00	0.00
14	3.00	1.00
15	3.00	0.00
16	2.00	1.00
17	2.00	0.50
18	2.00	0.00
19	1.00	1.00
20	1.00	0.00
21	0.00	1.00
22	0.00	0.50
23	0.00	0.00

ELEMENTOS									
ELE.	NOD I	NOD J	NOD K	NOD L	NOD M	NOD N	NOD O	NOD P	MATERIAL
1	8	5	3	2	1	4	6	7	1
2	13	10	8	7	6	9	11	12	1
3	18	15	13	12	11	14	16	17	1
4	23	20	18	17	16	19	21	22	1

CONDICIONES DE FRONTERA		
No.	DX	DY
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0

14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	1	1
22	1	1
23	1	1

MATERIALES			
No.	E	v	Espesor
1	2000000.00	0.20	0.30

NODOS CARGADOS	
CARGA	NODO
1	1

CARGAS		
No.	FX	FY
1	0.00	-10.00

ARCHIVO B.2. DATOS PARA EL PROCESADOR:
A:\EJEMPLO1.FMT
GENERADO POR EL PREPROCESADOR.

23

4

1

1

1

8.00

8.00

8.00

7.00

7.00

6.00

6.00

6.00

5.00

5.00

4.00

4.00

4.00

3.00

3.00

2.00

2.00

2.00

1.00

1.00

0.00

0.00

0.00

1.00

0.50

0.00

1.00

0.00

1.00

0.50

0.00

1.00

0.00

1.00

0.50

0.00

1.00

0.00

1.00

0.50

0.00

1.00

0.00

1.00

0.50

0.00

5
10
15
20
3
8
13
18
2
7
12
17
1
6
11
16
4
9
14
19
6
11
16
21
7
12
17
22
1
1
1
1
1
0.00
-10.00

ARCHIVO B.3. DATOS PARA EL USUARIO:

A:\RESULTAI

GENERADO POR EL PROCESADOR.

ELEMENTO NO.- 1

MATRIZ B

	1	2	3	4	5
1	-6.83013E-01	0.00000E+00	9.10684E-01	0.00000E+00	-2.27671E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-6.83013E-01	-6.66667E-01	9.10684E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-1.83013E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-2.27671E-01	4.88034E-01	3.33333E-01	-3.66025E-01	-1.83013E-01
	11	12	13	14	15
1	2.44017E-01	0.00000E+00	-6.10042E-02	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	2.44017E-01	-4.55342E-01	-6.10042E-02	1.82137E+00
	16				
1	0.00000E+00				
2	1.82137E+00				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	2.27671E-01	0.00000E+00	-9.10684E-01	0.00000E+00	6.83013E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	2.27671E-01	-6.66667E-01	-9.10684E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.10042E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	6.83013E-01	1.82137E+00	3.33333E-01	-4.55342E-01	6.10042E-02
	11	12	13	14	15
1	-2.44017E-01	0.00000E+00	1.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-2.44017E-01	-3.66025E-01	1.83013E-01	4.88034E-01
	16				
1	0.00000E+00				
2	4.88034E-01				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	1.83013E-01	0.00000E+00	-2.44017E-01	0.00000E+00	6.10042E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.83013E-01	-6.66667E-01	-2.44017E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.83013E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	6.10042E-02	-1.82137E+00	3.33333E-01	1.36603E+00	6.83013E-01
	11	12	13	14	15
1	-9.10684E-01	0.00000E+00	2.27671E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-9.10684E-01	1.22009E-01	2.27671E-01	-4.88034E-01

16
 1 0.00000E+00
 2 -4.88034E-01
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	-6.10042E-02	0.00000E+00	2.44017E-01	0.00000E+00	-1.83013E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-6.10042E-02	-6.66667E-01	2.44017E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-2.27671E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.83013E-01	-4.88034E-01	3.33333E-01	1.22009E-01	-2.27671E-01
	11	12	13	14	15
1	9.10684E-01	0.00000E+00	-6.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	9.10684E-01	1.36603E+00	-6.83013E-01	-1.82137E+00

16
 1 0.00000E+00
 2 -1.82137E+00
 3 -3.33333E-01

MATRIZ DE RIGIDECES DEL ELEMENTO 1

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-2.22222E+05	-1.25000E+05	1.87500E+05
2	1.77083E+05	7.63889E+05	-4.16667E+04	2.77778E+04	-1.04167E+04
3	-2.22222E+05	-4.16667E+04	7.77778E+05	-2.19152E-02	-2.22222E+05
4	-1.25000E+05	2.77778E+04	-2.42589E-02	7.77778E+05	1.25000E+05
5	1.87500E+05	-1.04167E+04	-2.22222E+05	1.25000E+05	4.51389E+05
6	1.04167E+04	2.50000E+05	4.16667E+04	2.77777E+04	-1.77083E+05
7	-2.56945E+05	-4.16667E+04	2.45011E-02	-1.66667E+05	-4.09722E+05
8	-4.16667E+04	-5.69445E+05	-1.66667E+05	5.60067E-02	4.16667E+04
9	2.25695E+05	7.29167E+04	-1.94445E+05	-4.16667E+04	2.18750E+05
10	7.29167E+04	3.81945E+05	-4.16667E+04	-1.94445E+05	1.04167E+04
11	-1.94445E+05	-4.16667E+04	5.55556E+04	-3.04353E-03	-1.94445E+05
12	-4.16667E+04	-1.94445E+05	-4.17410E-03	-4.44444E+05	4.16667E+04
13	2.18750E+05	1.04167E+04	-1.94445E+05	4.16667E+04	2.25695E+05
14	-1.04167E+04	4.37500E+05	4.16667E+04	-1.94445E+05	-7.29167E+04
15	-4.09722E+05	-1.25000E+05	4.09587E-02	1.66667E+05	-2.56945E+05
16	-4.16667E+04	-1.09722E+06	1.66667E+05	-4.07597E-03	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05	7.29167E+04
2	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04	3.81945E+05
3	4.16667E+04	2.76261E-02	-1.66667E+05	-1.94445E+05	-4.16667E+04
4	2.77777E+04	-1.66667E+05	8.72567E-02	-4.16667E+04	-1.94445E+05
5	-1.77083E+05	-4.09722E+05	4.16667E+04	2.18750E+05	1.04167E+04
6	7.63889E+05	1.25000E+05	-1.09722E+06	-1.04167E+04	4.37500E+05
7	1.25000E+05	1.02778E+06	-4.30795E-04	-4.09722E+05	-1.25000E+05
8	-1.09722E+06	-4.30795E-04	2.27778E+06	-4.16667E+04	-1.09722E+06
9	-1.04167E+04	-4.09722E+05	-4.16667E+04	4.51389E+05	1.77083E+05
10	4.37500E+05	-1.25000E+05	-1.09722E+06	1.77083E+05	7.63889E+05
11	4.16667E+04	4.19972E-02	1.66667E+05	-2.22222E+05	-4.16667E+04
12	-1.94445E+05	1.66667E+05	8.41864E-02	-1.25000E+05	2.77777E+04

13	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05	-1.04167E+04
14	3.81945E+05	4.16667E+04	-5.69444E+05	1.04167E+04	2.50000E+05
15	4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05	-4.16667E+04
16	-5.69445E+05	2.42523E-03	1.05556E+06	-4.16667E+04	-5.69445E+05
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05
2	-4.16667E+04	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05
3	5.55556E+04	5.13406E-04	-1.94445E+05	4.16667E+04	3.15837E-02
4	4.37835E-03	-4.44444E+05	4.16667E+04	-1.94445E+05	1.66667E+05
5	-1.94445E+05	4.16667E+04	2.25695E+05	-7.29167E+04	-2.56945E+05
6	4.16667E+04	-1.94445E+05	-7.29167E+04	3.81945E+05	4.16667E+04
7	3.73097E-02	1.66667E+05	-2.56945E+05	4.16667E+04	3.05556E+05
8	1.66667E+05	7.94989E-02	4.16667E+04	-5.69445E+05	3.63841E-03
9	-2.22222E+05	-1.25000E+05	1.87500E+05	1.04167E+04	-2.56945E+05
10	-4.16667E+04	2.77777E+04	-1.04167E+04	2.50000E+05	-4.16667E+04
11	7.77778E+05	2.63984E-02	-2.22222E+05	4.16666E+04	2.45743E-02
12	2.63984E-02	7.77778E+05	1.25000E+05	2.77778E+04	-1.66667E+05
13	-2.22222E+05	1.25000E+05	4.51389E+05	-1.77083E+05	-4.09722E+05
14	4.16667E+04	2.77778E+04	-1.77083E+05	7.63889E+05	1.25000E+05
15	4.01993E-02	-1.66667E+05	-4.09722E+05	1.25000E+05	1.02778E+06
16	-1.66667E+05	-4.08585E-03	4.16667E+04	-1.09722E+06	7.64841E-03
	16				
1	-4.16667E+04				
2	-1.09722E+06				
3	1.66667E+05				
4	-4.07597E-03				
5	4.16667E+04				
6	-5.69445E+05				
7	-1.09040E-03				
8	1.05556E+06				
9	-4.16667E+04				
10	-5.69445E+05				
11	-1.66667E+05				
12	-4.08585E-03				
13	4.16667E+04				
14	-1.09722E+06				
15	1.39841E-03				
16	2.27778E+06				

ELEMENTO NO. - 2
MATRIZ B

	1	2	3	4	5
1	-6.83013E-01	0.00000E+00	9.10684E-01	0.00000E+00	-2.27671E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-6.83013E-01	-6.66667E-01	9.10684E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-1.83013E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-2.27671E-01	4.88034E-01	3.33333E-01	-3.66025E-01	-1.83013E-01
	11	12	13	14	15
1	2.44017E-01	0.00000E+00	-6.10042E-02	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	2.44017E-01	-4.55342E-01	-6.10042E-02	1.82137E+00

16
 1 0.00000E+00
 2 1.82137E+00
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	2.27671E-01	0.00000E+00	-9.10684E-01	0.00000E+00	6.83013E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	2.27671E-01	-6.66667E-01	-9.10684E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.10042E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	6.83013E-01	1.82137E+00	3.33333E-01	-4.55342E-01	6.10042E-02
	11	12	13	14	15
1	-2.44017E-01	0.00000E+00	1.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-2.44017E-01	-3.66025E-01	1.83013E-01	4.88034E-01
	16				
1	0.00000E+00				
2	4.88034E-01				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	1.83013E-01	0.00000E+00	-2.44017E-01	0.00000E+00	6.10042E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.83013E-01	-6.66667E-01	-2.44017E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.83013E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	6.10042E-02	-1.82137E+00	3.33333E-01	1.36603E+00	6.83013E-01
	11	12	13	14	15
1	-9.10684E-01	0.00000E+00	2.27671E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-9.10684E-01	1.22009E-01	2.27671E-01	-4.88034E-01
	16				
1	0.00000E+00				
2	-4.88034E-01				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	-6.10042E-02	0.00000E+00	2.44017E-01	0.00000E+00	-1.83013E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-6.10042E-02	-6.66667E-01	2.44017E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-2.27671E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.83013E-01	-4.88034E-01	3.33333E-01	1.22009E-01	-2.27671E-01
	11	12	13	14	15
1	9.10684E-01	0.00000E+00	-6.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	9.10684E-01	1.36603E+00	-6.83013E-01	-1.82137E+00

16

1 0.00000E+00
 2 -1.82137E+00
 3 -3.33333E-01

MATRIZ DE RIGIDECES DEL ELEMENTO 2

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-2.22222E+05	-1.25000E+05	1.87500E+05
2	1.77083E+05	7.63889E+05	-4.16667E+04	2.77778E+04	-1.04167E+04
3	-2.22222E+05	-4.16667E+04	7.77778E+05	-1.21496E-02	-2.22222E+05
4	-1.25000E+05	2.77778E+04	-1.44933E-02	7.77778E+05	1.25000E+05
5	1.87500E+05	-1.04167E+04	-2.22222E+05	1.25000E+05	4.51389E+05
6	1.04167E+04	2.50000E+05	4.16667E+04	2.77778E+04	-1.77083E+05
7	-2.56945E+05	-4.16667E+04	8.87610E-03	-1.66667E+05	-4.09722E+05
8	-4.16667E+04	-5.69445E+05	-1.66667E+05	2.47567E-02	4.16667E+04
9	2.25695E+05	7.29167E+04	-1.94445E+05	-4.16667E+04	2.18750E+05
10	7.29167E+04	3.81945E+05	-4.16667E+04	-1.94445E+05	1.04167E+04
11	-1.94445E+05	-4.16667E+04	5.55556E+04	-3.04353E-03	-1.94445E+05
12	-4.16667E+04	-1.94445E+05	3.63841E-03	-4.44445E+05	4.16667E+04
13	2.18750E+05	1.04167E+04	-1.94445E+05	4.16667E+04	2.25695E+05
14	-1.04167E+04	4.37500E+05	4.16667E+04	-1.94445E+05	-7.29167E+04
15	-4.09722E+05	-1.25000E+05	4.09587E-02	1.66667E+05	-2.56945E+05
16	-4.16667E+04	-1.09722E+06	1.66667E+05	-4.07597E-03	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05	7.29167E+04
2	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04	3.81945E+05
3	4.16667E+04	1.20011E-02	-1.66667E+05	-1.94445E+05	-4.16667E+04
4	2.77778E+04	-1.66667E+05	2.47567E-02	-4.16667E+04	-1.94445E+05
5	-1.77083E+05	-4.09722E+05	4.16667E+04	2.18750E+05	1.04167E+04
6	7.63889E+05	1.25000E+05	-1.09722E+06	-1.04167E+04	4.37500E+05
7	1.25000E+05	1.02778E+06	-4.30795E-04	-4.09722E+05	-1.25000E+05
8	-1.09722E+06	-4.30795E-04	2.27778E+06	-4.16667E+04	-1.09722E+06
9	-1.04167E+04	-4.09722E+05	-4.16667E+04	4.51389E+05	1.77083E+05
10	4.37500E+05	-1.25000E+05	-1.09722E+06	1.77083E+05	7.63889E+05
11	4.16667E+04	2.24660E-02	1.66667E+05	-2.22222E+05	-4.16667E+04
12	-1.94445E+05	1.66667E+05	4.51239E-02	-1.25000E+05	2.77778E+04
13	-7.29167E+04	-2.56945E+05	4.16667E-04	1.87500E+05	-1.04167E+04
14	3.81945E+05	4.16667E+04	-5.69444E+05	1.04167E+04	2.50000E+05
15	4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05	-4.16667E+04
16	-5.69445E+05	2.42523E-03	1.05556E+06	-4.16667E+04	-5.69445E+05
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05
2	-4.16667E+04	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05
3	5.55556E+04	5.13406E-04	-1.94445E+05	4.16667E+04	3.15837E-02
4	2.42523E-03	-4.44445E+05	4.16667E+04	-1.94445E+05	1.66667E+05
5	-1.94445E+05	4.16667E+04	2.25695E+05	-7.29167E+04	-2.56945E+05
6	4.16667E+04	-1.94445E+05	-7.29167E+04	3.81945E+05	4.16667E+04
7	1.77785E-02	1.66667E+05	-2.56945E+05	4.16667E+04	3.05556E+05
8	1.66667E+05	4.04364E-02	4.16667E+04	-5.69445E+05	3.63841E-03
9	-2.22222E+05	-1.25000E+05	1.87500E+05	1.04167E+04	-2.56945E+05
10	-4.16667E+04	2.77778E+04	-1.04167E+04	2.50000E+05	-4.16667E+04
11	7.77778E+05	1.85859E-02	-2.22222E+05	4.16667E+04	2.45743E-02
12	1.85859E-02	7.77778E+05	1.25000E+05	2.77778E+04	-1.66663E+05
13	-2.22222E+05	1.25000E+05	4.51389E+05	-1.77083E+05	-4.09722E+05

14	4.16667E+04	2.77778E+04	-1.77083E+05	7.63889E+05	1.25000E+05
15	4.01993E-02	-1.66667E+05	-4.09722E+05	1.25000E+05	1.02778E+06
16	-1.66667E+05	-4.08585E-03	4.16667E+04	-1.09722E+06	7.64841E-03

16

1	-4.16667E+04
2	-1.09722E+06
3	1.66667E+05
4	-4.07597E-03
5	4.16667E+04
6	-5.69445E+05
7	-1.09040E-03
8	1.05556E+06
9	-4.16667E+04
10	-5.69445E+05
11	-1.66667E+05
12	-4.08585E-03
13	4.16667E+04
14	-1.09722E+06
15	1.39841E-03
16	2.27778E+06

ELEMENTO NO. - 3

MATRIZ B

	1	2	3	4	5
1	-6.83013E-01	0.00000E+00	9.10684E-01	0.00000E+00	-2.27671E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-6.83013E-01	-6.66667E-01	9.10684E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-1.83013E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-2.27671E-01	4.88034E-01	3.33333E-01	-3.66025E-01	-1.83013E-01
	11	12	13	14	15
1	2.44017E-01	0.00000E+00	-6.10042E-02	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	2.44017E-01	-4.55342E-01	-6.10042E-02	1.82137E+00
	16				
1	0.00000E+00				
2	1.82137E+00				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	2.27671E-01	0.00000E+00	-9.10684E-01	0.00000E+00	6.83013E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	2.27671E-01	-6.66667E-01	-9.10684E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.10042E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	6.83013E-01	1.82137E+00	3.33333E-01	-4.55342E-01	6.10042E-02
	11	12	13	14	15
1	-2.44017E-01	0.00000E+00	1.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-2.44017E-01	-3.66025E-01	1.83013E-01	4.88034E-01

16
 1 0.00000E+00
 2 4.88034E-01
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	1.83013E-01	0.00000E+00	-2.44017E-01	0.00000E+00	6.10042E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.83013E-01	-6.66667E-01	-2.44017E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.83013E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	6.10042E-02	-1.82137E+00	3.33333E-01	1.36603E+00	6.83013E-01
	11	12	13	14	15
1	-9.10684E-01	0.00000E+00	2.27671E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-9.10684E-01	1.22009E-01	2.27671E-01	-4.88034E-01

16
 1 0.00000E+00
 2 -4.88034E-01
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	-6.10042E-02	0.00000E+00	2.44017E-01	0.00000E+00	-1.83013E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-6.10042E-02	-6.66667E-01	2.44017E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-2.27671E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.83013E-01	-4.88034E-01	3.33333E-01	1.22009E-01	-2.27671E-01
	11	12	13	14	15
1	9.10684E-01	0.00000E+00	-6.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	9.10684E-01	1.36603E+00	-6.83013E-01	-1.82137E+00

16
 1 0.00000E+00
 2 -1.82137E+00
 3 -3.33333E-01

MATRIZ DE RIGIDECES DEL ELEMENTO 3

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-2.22222E+05	-1.25000E+05	1.87500E+05
2	1.77083E+05	7.63889E+05	-4.16667E+04	2.77778E+04	-1.04167E+04
3	-2.22222E+05	-4.16667E+04	7.77778E+05	-1.60558E-02	-2.22222E+05
4	-1.25000E+05	2.77778E+04	-6.68080E-03	7.77778E+05	1.25000E+05
5	1.87500E+05	-1.04167E+04	-2.22222E+05	1.25000E+05	4.51389E+05
6	1.04167E+04	2.50000E+05	4.16667E+04	2.77777E+04	-1.77083E+05
7	-2.56945E+05	-4.16667E+04	4.01261E-02	-1.66667E+05	-4.09722E+05
8	-4.16667E+04	-5.69445E+05	-1.66667E+05	9.13167E-03	4.16667E+04
9	2.25695E+05	7.29167E+04	-1.94445E+05	-4.16667E+04	2.18750E+05
10	7.29167E+04	3.81945E+05	-4.16667E+04	-1.94445E+05	1.04167E+04
11	-1.94445E+05	-4.16667E+04	5.55555E+04	-3.04353E-03	-1.94445E+05

12	-4.16667E+04	-1.94445E+05	3.63841E-03	-4.44445E+05	4.16667E+04
13	2.18750E+05	1.04167E+04	-1.94445E+05	4.16667E+04	2.25695E+05
14	-1.04167E+04	4.37500E+05	4.16667E+04	-1.94445E+05	-7.29167E+04
15	-4.09722E+05	-1.25000E+05	2.53337E-02	1.66667E+05	-2.56945E+05
16	-4.16667E+04	-1.09722E+06	1.66667E+05	2.71740E-02	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05	7.29167E+04
2	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04	3.81945E+05
3	4.16667E+04	2.76261E-02	-1.66667E+05	-1.94445E+05	-4.16667E+04
4	2.77777E+04	-1.66667E+05	4.03817E-02	-4.16667E+04	-1.94445E+05
5	-1.77083E+05	-4.09722E+05	4.16667E+04	2.18750E+05	1.04167E+04
6	7.63889E+05	1.25000E+05	-1.09722E+06	-1.04167E+04	4.37500E+05
7	1.25000E+05	1.02778E+06	1.52233E-03	-4.09722E+05	-1.25000E+05
8	-1.09722E+06	-4.33705E-03	2.27778E+06	-4.16667E+04	-1.09722E+06
9	-1.04167E+04	-4.09722E+05	-4.16667E+04	4.51389E+05	1.77083E+05
10	4.37500E+05	-1.25000E+05	-1.09722E+06	1.77083E+05	7.63889E+05
11	4.16667E+04	2.63722E-02	1.66667E+05	-2.22222E+05	-4.16667E+04
12	-1.94445E+05	1.66667E+05	2.16864E-02	-1.25000E+05	2.77777E+04
13	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05	-1.04167E+04
14	3.81945E+05	4.16667E+04	-5.69445E+05	1.04167E+04	2.50000E+05
15	4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05	-4.16667E+04
16	-5.69445E+05	4.72100E-04	1.05556E+06	-4.16667E+04	-5.69445E+05
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05
2	-4.16667E+04	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05
3	5.55555E+04	5.13406E-04	-1.94445E+05	4.16667E+04	1.59587E-02
4	4.72101E-04	-4.44445E+05	4.16667E+04	-1.94445E+05	1.66667E+05
5	-1.94445E+05	4.16667E+04	2.25695E+05	-7.29167E+04	-2.56945E+05
6	4.16667E+04	-1.94445E+05	-7.29167E+04	3.81945E+05	4.16667E+04
7	3.73097E-02	1.66667E+05	-2.56945E+05	4.16667E+04	3.05556E+05
8	1.66667E+05	1.69989E-02	4.16667E+04	-5.69445E+05	3.63841E-03
9	-2.22222E+05	-1.25000E+05	1.87500E+05	1.04167E+04	-2.56945E+05
10	-4.16667E+04	2.77777E+04	-1.04167E+04	2.50000E+05	-4.16667E+04
11	7.77778E+05	1.85859E-02	-2.22222E+05	4.16667E+04	8.94927E-03
12	1.07734E-02	7.77778E+05	1.25000E+05	2.77778E+04	-1.66667E+05
13	-2.22222E+05	1.25000E+05	4.51389E+05	-1.77083E+05	-4.09722E+05
14	4.16667E+04	2.77778E+04	-1.77083E+05	7.63889E+05	1.25000E+05
15	2.45743E-02	-1.66667E+05	-4.09722E+05	1.25000E+05	1.02778E+06
16	-1.66667E+05	2.71642E-02	4.16667E+04	-1.09722E+06	7.64841E-03
	16				
1	-4.16667E+04				
2	-1.09722E+06				
3	1.66667E+05				
4	2.71740E-02				
5	4.16667E+04				
6	-5.69445E+05				
7	8.62725E-04				
8	1.05556E+06				
9	-4.16667E+04				
10	-5.69445E+05				
11	-1.66667E+05				
12	2.71642E-02				
13	4.16667E+04				
14	-1.09722E+06				

15 1.39841E-03
16 2.27778E+06

ELEMENTO NO.- 4
MATRIZ B

	1	2	3	4	5
1	-6.83013E-01	0.00000E+00	9.10684E-01	0.00000E+00	-2.27671E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-6.83013E-01	-6.66667E-01	9.10684E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-1.83013E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-2.27671E-01	4.88034E-01	3.33333E-01	-3.66025E-01	-1.83013E-01
	11	12	13	14	15
1	2.44017E-01	0.00000E+00	-6.10042E-02	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	2.44017E-01	-4.55342E-01	-6.10042E-02	1.82137E+00
	16				
1	0.00000E+00				
2	1.82137E+00				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	2.27671E-01	0.00000E+00	-9.10684E-01	0.00000E+00	6.83013E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	2.27671E-01	-6.66667E-01	-9.10684E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.10042E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	6.83013E-01	1.82137E+00	3.33333E-01	-4.55342E-01	6.10042E-02
	11	12	13	14	15
1	-2.44017E-01	0.00000E+00	1.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-2.44017E-01	-3.66025E-01	1.83013E-01	4.88034E-01
	16				
1	0.00000E+00				
2	4.88034E-01				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	1.83013E-01	0.00000E+00	-2.44017E-01	0.00000E+00	6.10042E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.83013E-01	-6.66667E-01	-2.44017E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.83013E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	6.10042E-02	-1.82137E+00	3.33333E-01	1.36603E+00	6.83013E-01
	11	12	13	14	15
1	-9.10684E-01	0.00000E+00	2.27671E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-9.10684E-01	1.22009E-01	2.27671E-01	-4.88034E-01

16
 1 0.00000E+00
 2 -4.88034E-01
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	-6.10042E-02	0.00000E+00	2.44017E-01	0.00000E+00	-1.83013E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-6.10042E-02	-6.66667E-01	2.44017E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-2.27671E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.83013E-01	-4.88034E-01	3.33333E-01	1.22009E-01	-2.27671E-01
	11	12	13	14	15
1	9.10684E-01	0.00000E+00	-6.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	9.10684E-01	1.36603E+00	-6.83013E-01	-1.82137E+00
	16				
1	0.00000E+00				
2	-1.82137E+00				
3	-3.33333E-01				

MATRIZ DE RIGIDECES DEL ELEMENTO 4

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-2.22222E+05	-1.25000E+05	1.87500E+05
2	1.77083E+05	7.63889E+05	-4.16667E+04	2.77778E+04	-1.04167E+04
3	-2.22222E+05	-4.16667E+04	7.77778E+05	-4.30795E-04	-2.22222E+05
4	-1.25000E+05	2.77778E+04	1.13171E-03	7.77778E+05	1.25000E+05
5	1.87500E+05	-1.04167E+04	-2.22222E+05	1.25000E+05	4.51389E+05
6	1.04167E+04	2.50000E+05	4.16667E+04	2.77778E+04	-1.77083E+05
7	-2.56945E+05	-4.16667E+04	4.01261E-02	-1.66667E+05	-4.09722E+05
8	-4.16667E+04	-5.69445E+05	-1.66667E+05	4.03817E-02	4.16667E+04
9	2.25695E+05	7.29167E+04	-1.94445E+05	-4.16667E+04	2.18750E+05
10	7.29167E+04	3.81945E+05	-4.16667E+04	-1.94445E+05	1.04167E+04
11	-1.94445E+05	-4.16667E+04	5.55556E+04	3.59710E-03	-1.94445E+05
12	-4.16667E+04	-1.94445E+05	-1.04910E-03	-4.44445E+05	4.16667E+04
13	2.18750E+05	1.04167E+04	-1.94445E+05	4.16667E+04	2.25695E+05
14	-1.04167E+04	4.37500E+05	4.16667E+04	-1.94445E+05	-7.29167E+04
15	-4.09722E+05	-1.25000E+05	4.09587E-02	1.66667E+05	-2.56945E+05
16	-4.16667E+04	-1.09722E+06	1.66667E+05	5.84240E-02	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05	7.29167E+04
2	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04	3.81945E+05
3	4.16667E+04	3.23136E-02	-1.66667E+05	-1.94445E+05	-4.16667E+04
4	2.77778E+04	-1.66667E+05	4.03817E-02	-4.16667E+04	-1.94445E+05
5	-1.77083E+05	-4.09722E+05	4.16667E+04	2.18750E+05	1.04167E+04
6	7.63889E+05	1.25000E+05	-1.09722E+06	-1.04167E+04	4.37500E+05
7	1.25000E+05	1.02778E+06	-8.21421E-04	-4.09722E+05	-1.25000E+05
8	-1.09722E+06	-4.33705E-03	2.27778E+06	-4.16667E+04	-1.09722E+06
9	-1.04167E+04	-4.09722E+05	-4.16667E+04	4.51389E+05	1.77083E+05
10	4.37500E+05	-1.25000E+05	-1.09722E+06	1.77083E+05	7.63889E+05
11	4.16667E+04	2.63722E-02	1.66667E+05	-2.22222E+05	-4.16667E+04
12	-1.94445E+05	1.66667E+05	2.16864E-02	-1.25000E+05	2.77778E+04

13	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05	-1.04167E+04
14	3.81945E+05	4.16667E+04	-5.69445E+05	1.04167E+04	2.50000E+05
15	4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05	-4.16667E+04
16	-5.69445E+05	-2.26228E-03	1.05556E+06	-4.16667E+04	-5.69445E+05
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05
2	-4.16667E+04	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05
3	5.55556E+04	-4.17410E-03	-1.94445E+05	4.16667E+04	3.15837E-02
4	-1.87165E-03	-4.44445E+05	4.16667E+04	-1.94445E+05	1.66667E+05
5	-1.94445E+05	4.16667E+04	2.25695E+05	-7.29167E+04	-2.56945E+05
6	4.16667E+04	-1.94445E+05	-7.29167E+04	3.81945E+05	4.16667E+04
7	4.19972E-02	1.66667E+05	-2.56945E+05	4.16667E+04	3.05556E+05
8	1.66667E+05	2.16864E-02	4.16667E+04	-5.69445E+05	3.63841E-03
9	-2.22222E+05	-1.25000E+05	1.87500E+05	1.04167E+04	-2.56945E+05
10	-4.16667E+04	2.77777E+04	-1.04167E+04	2.50000E+05	-4.16667E+04
11	7.77778E+05	-1.64086E-04	-2.22222E+05	4.16667E+04	2.76993E-02
12	1.39841E-03	7.77778E+05	1.25000E+05	2.77778E+04	-1.66667E+05
13	-2.22222E+05	1.25000E+05	4.51389E+05	-1.77083E+05	-4.09722E+05
14	4.16667E+04	2.77778E+04	-1.77083E+05	7.63889E+05	1.25000E+05
15	4.33243E-02	-1.66667E+05	-4.09722E+05	1.25000E+05	1.02778E+06
16	-1.66667E+05	6.46642E-02	4.16667E+04	-1.09722E+06	7.64841E-03
	16				
1	-4.16667E+04				
2	-1.09722E+06				
3	1.66667E+05				
4	5.84240E-02				
5	4.16667E+04				
6	-5.69445E+05				
7	2.03460E-03				
8	1.05556E+06				
9	-4.16667E+04				
10	-5.69445E+05				
11	-1.66667E+05				
12	6.46642E-02				
13	4.16667E+04				
14	-1.09722E+06				
15	1.39841E-03				
16	2.27778E+06				

MATRIZ DE RIGIDECES DE LA ESTRUCTURA

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-4.09722E+05	-4.16667E+04	2.18750E+05
2	1.77083E+05	7.63889E+05	-1.25000E+05	-1.09722E+06	1.04167E+04
3	-4.09722E+05	-1.25000E+05	1.02778E+06	-4.30795E-04	-4.09722E+05
4	-4.16667E+04	-1.09722E+06	-4.30795E-04	2.27778E+06	4.16667E+04
5	2.18750E+05	1.04167E+04	-4.09722E+05	4.16667E+04	4.51389E+05
6	-1.04167E+04	4.37500E+05	1.25000E+05	-1.09722E+06	-1.77083E+05
7	-2.22222E+05	-4.16667E+04	4.19972E-02	1.66667E+05	-1.94445E+05
8	-1.25000E+05	2.77777E+04	1.66667E+05	8.41864E-02	4.16667E+04
9	-1.94445E+05	-4.16667E+04	2.76261E-02	-1.66667E+05	-2.22222E+05
10	-4.16667E+04	-1.94445E+05	-1.66667E+05	8.72567E-02	1.25000E+05
11	1.87500E+05	-1.04167E+04	-2.56945E+05	4.16667E+04	2.25695E+05
12	1.04167E+04	2.50000E+05	4.16667E+04	-5.69444E+05	-7.29167E+04
13	-2.56945E+05	-4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05

14	-4.16667E+04	-5.69445E+05	2.42523E-03	1.05556E+06	4.16667E+04
15	2.25695E+05	7.29167E+04	-2.56945E+05	-4.16667E+04	1.87500E+05
16	7.29167E+04	3.81945E+05	-4.16667E+04	-5.69445E+05	-1.04167E+04
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
22	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
23	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
24	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
25	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
26	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	6	7	8	9	10
1	-1.04167E+04	-2.22222E+05	-1.25000E+05	-1.94445E+05	-4.16667E+04
2	4.37500E+05	-4.16667E+04	2.77777E+04	-4.16667E+04	-1.94445E+05
3	1.25000E+05	3.73097E-02	1.66667E+05	2.45011E-02	-1.66667E+05
4	-1.09722E+06	1.66667E+05	7.94989E-02	-1.66667E+05	5.60067E-02
5	-1.77083E+05	-1.94445E+05	4.16667E+04	-2.22222E+05	1.25000E+05
6	7.63889E+05	4.16667E+04	-1.94445E+05	4.16667E+04	2.77777E+04
7	4.16667E+04	7.77778E+05	2.63984E-02	5.55556E+04	-3.04353E-03
8	-1.94445E+05	2.63984E-02	7.77778E+05	-4.17410E-03	-4.44444E+05
9	4.16667E+04	5.55556E+04	5.13406E-04	7.77778E+05	-2.19152E-02
10	2.77777E+04	4.37835E-03	-4.44444E+05	-2.42589E-02	7.77778E+05
11	-7.29167E+04	-2.22222E+05	1.25000E+05	-1.94445E+05	4.16667E+04
12	3.81945E+05	4.16667E+04	2.77778E+04	4.16667E+04	-1.94445E+05
13	4.16667E+04	4.01993E-02	-1.66667E+05	4.09587E-02	1.66667E+05
14	-5.69445E+05	-1.66667E+05	-4.08585E-03	1.66667E+05	-4.07597E-03
15	1.04167E+04	-1.94445E+05	-4.16667E+04	-2.22222E+05	-1.25000E+05
16	2.50000E+05	-4.16667E+04	-1.94445E+05	-4.16667E+04	2.77778E+04
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
22	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
23	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
24	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
25	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
26	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	11	12	13	14	15
1	1.87500E+05	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05
2	-1.04167E+04	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04
3	-2.56945E+05	4.16667E+04	3.05556E+05	-1.09040E-03	-2.56945E+05
4	4.16667E+04	-5.69445E+05	3.63841E-03	1.05556E+06	-4.16667E+04
5	2.25695E+05	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05
6	-7.29167E+04	3.81945E+05	4.16667E+04	-5.69445E+05	1.04167E+04
7	-2.22222E+05	4.16667E+04	2.45743E-02	-1.66667E+05	-1.94445E+05
8	1.25000E+05	2.77778E+04	-1.66667E+05	-4.08585E-03	-4.16667E+04
9	-1.94445E+05	4.16667E+04	3.15837E-02	1.66667E+05	-2.22222E+05
10	4.16667E+04	-1.94445E+05	1.66667E+05	-4.07597E-03	-1.25000E+05
11	9.02778E+05	3.12500E-02	-8.19445E+05	7.81250E-03	4.37500E+05
12	3.12500E-02	1.52778E+06	0.00000E+00	-2.19445E+06	3.90625E-03
13	-8.19445E+05	0.00000E+00	2.05556E+06	9.67619E-04	-8.19445E+05
14	3.90625E-03	-2.19445E+06	7.21762E-03	4.55556E+06	0.00000E+00
15	4.37500E+05	-9.76563E-04	-8.19445E+05	-3.90625E-03	9.02778E+05
16	-2.92969E-03	8.75000E+05	-1.56250E-02	-2.19445E+06	-1.56250E-02
17	-2.22222E+05	-4.16667E+04	2.24660E-02	1.66667E+05	-1.94445E+05
18	-1.25000E+05	2.77778E+04	1.66667E+05	4.51239E-02	4.16667E+04
19	-1.94445E+05	-4.16667E+04	1.20011E-02	-1.66667E+05	-2.22222E+05
20	-4.16667E+04	-1.94445E+05	-1.66667E+05	2.47567E-02	1.25000E+05
21	1.87500E+05	-1.04167E+04	-2.56945E+05	4.16667E+04	2.25695E+05
22	1.04167E+04	2.50000E+05	4.16667E+04	-5.69444E+05	-7.29167E+04
23	-2.56945E+05	-4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05
24	-4.16667E+04	-5.69445E+05	2.42523E-03	1.05556E+06	4.16667E+04
25	2.25695E+05	7.29167E+04	-2.56945E+05	-4.16667E+04	1.87500E+05
26	7.29167E+04	3.81945E+05	-4.16667E+04	-5.69445E+05	-1.04167E+04
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	16	17	18	19	20
1	7.29167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	3.81945E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	-5.69445E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	-1.04167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	2.50000E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	-1.94445E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	2.77778E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	-1.95313E-03	-2.22222E+05	-1.25000E+05	-1.94445E+05	-4.16667E+04
12	8.75000E+05	-4.16667E+04	2.77778E+04	-4.16667E+04	-1.94445E+05
13	-7.81250E-03	1.77778E-02	1.66667E+05	8.87610E-03	-1.66667E+05
14	-2.19445E+06	1.66667E+05	4.04364E-02	-1.66667E+05	2.47567E-02
15	-1.56250E-02	-1.94445E+05	4.16667E+04	-2.22222E+05	1.25000E+05
16	1.52778E+06	4.16667E+04	-1.94445E+05	4.16667E+04	2.77778E+04
17	4.16667E+04	7.77778E+05	1.85859E-02	5.55556E+04	-3.04353E-03
18	-1.94445E+05	1.85859E-02	7.77778E+05	3.63841E-03	-4.44445E+05
19	4.16667E+04	5.55556E+04	5.13406E-04	7.77778E+05	-1.21496E-02
20	2.77778E+04	2.42523E-03	-4.44445E+05	-1.44933E-02	7.77778E+05
21	-7.29167E+04	-2.22222E+05	1.25000E+05	-1.94445E+05	4.16667E+04
22	3.81945E+05	4.16667E+04	2.77778E+04	4.16667E+04	-1.94445E+05
23	4.16667E+04	4.01993E-02	-1.66667E+05	4.09587E-02	1.66667E+05
24	-5.69445E+05	-1.66667E+05	-4.08585E-03	1.66667E+05	-4.07597E-03
25	1.04167E+04	-1.94445E+05	-4.16667E+04	-2.22222E+05	-1.25000E+05
26	2.50000E+05	-4.16667E+04	-1.94445E+05	-4.16667E+04	2.77778E+04
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	21	22	23	24	25
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	1.87500E+05	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05

12	-1.04167E+04	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04
13	-2.56945E+05	4.16667E+04	3.05556E+05	-1.09040E-03	-2.56945E+05
14	4.16667E+04	-5.69445E+05	3.63841E-03	1.05556E+06	-4.16667E+04
15	2.25695E+05	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05
16	-7.29167E+04	3.81945E+05	4.16667E+04	-5.69445E+05	1.04167E+04
17	-2.22222E+05	4.16667E+04	2.45743E-02	-1.66667E+05	-1.94445E+05
18	1.25000E+05	2.77778E+04	-1.66667E+05	-4.08585E-03	-4.16667E+04
19	-1.94445E+05	4.16667E+04	3.15837E-02	1.66667E+05	-2.22222E+05
20	4.16667E+04	-1.94445E+05	1.66667E+05	-4.07597E-03	-1.25000E+05
21	9.02778E+05	0.00000E+00	-8.19445E+05	3.90625E-03	4.37500E+05
22	0.00000E+00	1.52778E+06	0.00000E+00	-2.19445E+06	2.92969E-03
23	-8.19445E+05	0.00000E+00	2.05556E+06	2.92074E-03	-8.19445E+05
24	0.00000E+00	-2.19445E+06	3.31137E-03	4.55556E+06	0.00000E+00
25	4.37500E+05	9.76563E-04	-8.19445E+05	0.00000E+00	9.02778E+05
26	-9.76563E-04	8.75000E+05	-7.81250E-03	-2.19445E+06	1.56250E-02
27	-2.22222E+05	-4.16667E+04	2.63722E-02	1.66667E+05	-1.94445E+05
28	-1.25000E+05	2.77777E+04	1.66667E+05	2.16864E-02	4.16667E+04
29	-1.94445E+05	-4.16667E+04	2.76261E-02	-1.66667E+05	-2.22222E+05
30	-4.16667E+04	-1.94445E+05	-1.66667E+05	4.03817E-02	1.25000E+05
31	1.87500E+05	-1.04167E+04	-2.56945E+05	4.16667E+04	2.25695E+05
32	1.04167E+04	2.50000E+05	4.16667E+04	-5.69445E+05	-7.29167E+04
33	-2.56945E+05	-4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05
34	-4.16667E+04	-5.69445E+05	4.72100E-04	1.05556E+06	4.16667E+04
35	2.25695E+05	7.29167E+04	-2.56945E+05	-4.16667E+04	1.87500E+05
36	7.29167E+04	3.81945E+05	-4.16667E+04	-5.69445E+05	-1.04167E+04
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	26	27	28	29	30
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	7.29167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
12	3.81945E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
13	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	-5.69445E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	-1.04167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	2.50000E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	-1.94445E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	2.77778E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	-9.76563E-04	-2.22222E+05	-1.25000E+05	-1.94445E+05	-4.16667E+04
22	8.75000E+05	-4.16667E+04	2.77777E+04	-4.16667E+04	-1.94445E+05
23	-7.81250E-03	3.73097E-02	1.66667E+05	4.01261E-02	-1.66667E+05
24	-2.19445E+06	1.66667E+05	1.69989E-02	-1.66667E+05	9.13167E-03

25	1.56250E-02	-1.94445E+05	4.16667E+04	-2.22222E+05	1.25000E+05
26	1.52778E+06	4.16667E+04	-1.94445E+05	4.16667E+04	2.77777E+04
27	4.16667E+04	7.77778E+05	1.85859E-02	5.55555E+04	-3.04353E-03
28	-1.94445E+05	1.07734E-02	7.77778E+05	3.63841E-03	-4.44445E+05
29	4.16667E+04	5.55555E+04	5.13406E-04	7.77778E+05	-1.60558E-02
30	2.77777E+04	4.72101E-04	-4.44445E+05	-6.68080E-03	7.77778E+05
31	-7.29167E+04	-2.22222E+05	1.25000E+05	-1.94445E+05	4.16667E+04
32	3.81945E+05	4.16667E+04	2.77778E+04	4.16667E+04	-1.94445E+05
33	4.16667E+04	2.45743E-02	-1.66667E+05	2.53337E-02	1.66667E+05
34	-5.69445E+05	-1.66667E+05	2.71642E-02	1.66667E+05	2.71740E-02
35	1.04167E+04	-1.94445E+05	-4.16667E+04	-2.22222E+05	-1.25000E+05
36	2.50000E+05	-4.16667E+04	-1.94445E+05	-4.16667E+04	2.77778E+04
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	31	32	33	34	35
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
12	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
13	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	1.87500E+05	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05
22	-1.04167E+04	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04
23	-2.56945E+05	4.16667E+04	3.05556E+05	8.62725E-04	-2.56945E+05
24	4.16667E+04	-5.69445E+05	3.63841E-03	1.05556E+06	-4.16667E+04
25	2.25695E+05	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05
26	-7.29167E+04	3.81945E+05	4.16667E+04	-5.69445E+05	1.04167E+04
27	-2.22222E+05	4.16667E+04	8.94927E-03	-1.66667E+05	-1.94445E+05
28	1.25000E+05	2.77778E+04	-1.66667E+05	2.71642E-02	-4.16667E+04
29	-1.94445E+05	4.16667E+04	1.59587E-02	1.66667E+05	-2.22222E+05
30	4.16667E+04	-1.94445E+05	1.66667E+05	2.71740E-02	-1.25000E+05
31	9.02778E+05	0.00000E+00	-8.19445E+05	0.00000E+00	4.37500E+05
32	0.00000E+00	1.52778E+06	0.00000E+00	-2.19445E+06	2.92969E-03
33	-8.19445E+05	0.00000E+00	2.05556E+06	5.76994E-04	-8.19445E+05
34	0.00000E+00	-2.19445E+06	3.31137E-03	4.55556E+06	0.00000E+00
35	4.37500E+05	4.88281E-03	-8.19445E+05	3.90625E-03	9.02778E+05
36	-9.76563E-04	8.75000E+05	0.00000E+00	-2.19445E+06	1.56250E-02
37	-2.22222E+05	-4.16667E+04	2.63722E-02	1.66667E+05	-1.94445E+05

38	-1.25000E+05	2.77777E+04	1.66667E+05	2.16864E-02	4.16667E+04
39	-1.94445E+05	-4.16667E+04	3.23136E-02	-1.66667E+05	-2.22222E+05
40	-4.16667E+04	-1.94445E+05	-1.66667E+05	4.03817E-02	1.25000E+05
	36	37	38	39	40
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	0.00000E+00	0.00000E+00	0.00000E-00	0.00000E+00	0.00000E+00
12	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
13	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	7.29167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
22	3.81945E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
23	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
24	-5.69445E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
25	-1.04167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
26	2.50000E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
27	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	-1.94445E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	2.77778E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	-9.76563E-04	-2.22222E+05	-1.25000E+05	-1.94445E+05	-4.16667E+04
32	8.75000E+05	-4.16666E+04	2.77777E+04	-4.16667E+04	-1.94445E+05
33	0.00000E+00	4.19972E-02	1.66667E+05	4.01261E-02	-1.66667E+05
34	-2.19445E+06	1.66667E+05	2.16864E-02	-1.66667E+05	4.03817E-02
35	1.56250E-02	-1.94445E+05	4.16667E+04	-2.22222E+05	1.25000E+05
36	1.52778E+06	4.16667E+04	-1.94445E+05	4.16667E+04	2.77777E+04
37	4.16667E+04	7.77778E+05	-1.64086E-04	5.55556E+04	3.59710E-03
38	-1.94445E+05	1.39841E-03	7.77778E+05	-1.04910E-03	-4.44445E+05
39	4.16667E+04	5.55556E+04	-4.17410E-03	7.77778E+05	-4.30795E-04
40	2.77777E+04	-1.87165E-03	-4.44445E+05	1.13171E-03	7.77778E+05

VECTOR DE CARGA

1

1	0.000000E+00
2	-1.000000E+01
3	0.000000E+00
4	0.000000E+00
5	0.000000E+00
6	0.000000E+00
7	0.000000E+00

8 0.000000E+00
9 0.000000E+00
10 0.000000E+00
11 0.000000E+00
12 0.000000E+00
13 0.000000E+00
14 0.000000E+00
15 0.000000E+00
16 0.000000E+00
17 0.000000E+00
18 0.000000E+00
19 0.000000E+00
20 0.000000E+00
21 0.000000E+00
22 0.000000E+00
23 0.000000E+00
24 0.000000E+00
25 0.000000E+00
26 0.000000E+00
27 0.000000E+00
28 0.000000E+00
29 0.000000E+00
30 0.000000E+00
31 0.000000E+00
32 0.000000E+00
33 0.000000E+00
34 0.000000E+00
35 0.000000E+00
36 0.000000E+00
37 0.000000E+00
38 0.000000E+00
39 0.000000E+00
40 0.000000E+00

VECTOR DE DESPLAZAMIENTOS

1
1 3.197445E-03
2 -3.423501E-02
3 -2.804843E-06
4 -3.420026E-02
5 -3.176458E-03
6 -3.417191E-02
7 3.137391E-03
8 -2.781814E-02
9 -3.132679E-03
10 -2.783629E-02
11 2.982885E-03
12 -2.166369E-02
13 2.106035E-06
14 -2.164366E-02
15 -2.985894E-03
16 -2.164337E-02
17 2.733818E-03
18 -1.587830E-02

19 -2.735784E-03
 20 -1.588225E-02
 21 2.386987E-03
 22 -1.070867E-02
 23 -2.234183E-07
 24 -1.068794E-02
 25 -2.385185E-03
 26 -1.070198E-02
 27 1.935200E-03
 28 -6.333398E-03
 29 -1.934899E-03
 30 -6.335208E-03
 31 1.378366E-03
 32 -2.971843E-03
 33 1.448563E-07
 34 -2.930157E-03
 35 -1.378618E-03
 36 -2.969896E-03
 37 7.290486E-04
 38 -7.960403E-04
 39 -7.293695E-04
 40 -7.963376E-04

NUMERO DE ELEMENTO 1
 PUNTO GAUSSIANO X 6.422650
 PUNTO GAUSSIANO Y 2.113249E-01
 ESFUERZOS

1
 1 -1.822444E+02
 2 2.370425E+01
 3 -4.476399E+01
 PUNTO GAUSSIANO X 7.577350
 PUNTO GAUSSIANO Y 2.113249E-01
 ESFUERZOS

1
 1 -4.882001E+01
 2 -1.676012E+01
 3 -3.029079E+01
 PUNTO GAUSSIANO X 7.577350
 PUNTO GAUSSIANO Y 7.886751E-01
 ESFUERZOS

1
 1 4.882359E+01
 2 -3.986668E+01
 3 -3.640569E+01
 PUNTO GAUSSIANO X 6.422650
 PUNTO GAUSSIANO Y 7.886751E-01
 ESFUERZOS

1
 1 1.822521E+02
 2 1.847318E+01
 3 -2.194558E+01
 NUMERO DE ELEMENTO 2
 PUNTO GAUSSIANO X 4.422650

PUNTO GAUSSIANO Y 2.113249E-01
ESFUERZOS

1

1 -4.134572E+02

2 -5.589622E+00

3 -3.681737E+01

PUNTO GAUSSIANO X 5.577350

PUNTO GAUSSIANO Y 2.113249E-01

ESFUERZOS

1

1 -2.799564E+02

2 -1.467045E+01

3 -2.186309E+01

PUNTO GAUSSIANO X 5.577350

PUNTO GAUSSIANO Y 7.886751E-01

ESFUERZOS

1

1 2.799623E+02

2 -8.548761E+00

3 -4.488255E+01

PUNTO GAUSSIANO X 4.422650

PUNTO GAUSSIANO Y 7.886751E-01

ESFUERZOS

1

1 4.134646E+02

2 1.385196E+01

3 -2.992510E+01

NUMERO DE ELEMENTO 3

PUNTO GAUSSIANO X 2.422650

PUNTO GAUSSIANO Y 2.113249E-01

ESFUERZOS

1

1 -6.448167E+02

2 3.858911E+01

3 -3.457703E+01

PUNTO GAUSSIANO X 3.577350

PUNTO GAUSSIANO Y 2.113249E-01

ESFUERZOS

1

1 -5.112143E+02

2 -2.370114E+00

3 -3.272036E+01

PUNTO GAUSSIANO X 3.577350

PUNTO GAUSSIANO Y 7.886751E-01

ESFUERZOS

1

1 5.112237E+02

2 -4.030982E+00

3 -3.407222E+01

PUNTO GAUSSIANO X 2.422650

PUNTO GAUSSIANO Y 7.886751E-01

ESFUERZOS

1

1 6.448243E+02

2 -3.404138E+01
3 -3.222144E+01
NUMERO DE ELEMENTO 4
PUNTO GAUSSIANO X 4.226497E-01
PUNTO GAUSSIANO Y 2.113249E-01
ESFUERZOS
1
1 -8.762557E+02
2 -1.346406E+02
3 -3.341526E+01
PUNTO GAUSSIANO X 1.577350
PUNTO GAUSSIANO Y 2.113249E-01
ESFUERZOS
1
1 -7.426282E+02
2 -1.597486E+00
3 -3.240700E+01
PUNTO GAUSSIANO X 1.577350
PUNTO GAUSSIANO Y 7.886751E-01
ESFUERZOS
1
1 7.426365E+02
2 -1.153945E+00
3 -3.440397E+01
PUNTO GAUSSIANO X 4.226497E-01
PUNTO GAUSSIANO Y 7.886751E-01
ESFUERZOS
1
1 8.762640E+02
2 1.363848E+02
3 -3.339817E+01

ARCHIVO B.4. DATOS PARA EL PROCESADOR (INDICADORES DE
ECUACION Y MULTIPLICACION DE MATRICES DE CADA ELEMENTO):

A:\AUXILIA1

GENERADO POR EL PROCESADOR.

15	16	9	10
5	6	3	4
1	2	7	8
11	12	13	14
-1422943.000000	-569177.300000	1897258.000000	-277777.800000
-474314.400000	-50836.860000	694444.500000	203347.500000
-381276.500000	-152510.600000	508368.700000	277777.800000
-127092.200000	-189725.800000	-694444.400000	758903.000000
-284588.600000	-2845886.000000	379451.500000	-1388889.000000
-94862.880000	-254184.300000	138888.900000	1016737.000000
-76255.300000	-762553.000000	101673.700000	1388889.000000
-25418.430000	-948628.800000	-138888.900000	3794515.000000
-1138355.000000	-569177.300000	-555555.600000	758903.000000
-101673.700000	-189725.800000	406694.900000	277777.800000
-305021.200000	-152510.600000	555555.600000	203347.500000
-379451.500000	-50836.870000	1517806.000000	-277777.800000
6.422650	2.113249E-01	474314.400000	-50836.870000
-1897258.000000	-277777.800000	1422943.000000	-569177.300000
694444.500000	758903.000000	127092.200000	-189725.800000
-508368.700000	277777.800000	381276.500000	-152510.600000
-694444.600000	203347.500000	94862.880000	-254184.400000
-379451.500000	-1388889.000000	284588.700000	-2845887.000000
138888.900000	3794515.000000	25418.440000	-948628.800000
-101673.700000	1388889.000000	76255.300000	-762553.100000
-138888.900000	1016737.000000	-101673.700000	189725.800000
-555555.500000	-758903.100000	-1138355.000000	569177.300000
1517806.000000	277777.800000	-379451.500000	50836.870000
555555.600000	-203347.500000	-305021.200000	152510.600000
406694.900000	-277777.800000	7.577350	2.113249E-01
381276.500000	152510.600000	-508368.700000	-277777.800000
127092.200000	189725.800000	694444.500000	-758903.000000
1422943.000000	569177.300000	-1897258.000000	277777.800000
474314.400000	50836.870000	-694444.600000	-203347.500000
76255.300000	762553.100000	-101673.700000	-1388889.000000
25418.440000	948628.800000	138888.900000	-3794515.000000
284588.700000	2845887.000000	-379451.500000	1388889.000000
94862.880000	254184.400000	-138888.900000	-1016737.000000
305021.200000	152510.600000	-555555.600000	-203347.500000
379451.500000	50836.870000	-1517806.000000	277777.800000
1138355.000000	569177.300000	555555.500000	-758903.100000
101673.700000	189725.800000	-406694.900000	-277777.800000
7.577350	7.886751E-01	-127092.200000	189725.800000
508368.700000	-277777.800000	-381276.500000	152510.600000
694444.500000	-203347.500000	-474314.400000	50836.860000
1897258.000000	277777.800000	-1422943.000000	569177.300000
-694444.400000	-758903.000000	-25418.440000	948628.800000
101673.700000	-1388889.000000	-76255.300000	762553.000000
138888.900000	-1016737.000000	-94862.880000	254184.300000
379451.500000	1388889.000000	-284588.600000	2845886.000000
-138888.900000	-3794515.000000	379451.500000	-50836.870000

-555555.600000	203347.500000	305021.200000	-152510.600000
-406694.900000	277777.800000	101673.700000	-189725.800000
555555.600000	758903.000000	1138355.000000	-569177.300000
-1517806.000000	-277777.800000	6.422650	7.886751E-01
25	26	19	20
15	16	13	14
11	12	17	18
21	22	23	24
-1422943.000000	-569177.300000	1897258.000000	-277777.800000
-474314.400000	-50836.860000	694444.500000	203347.500000
-381276.500000	-152510.600000	508368.700000	277777.800000
-127092.200000	-189725.800000	-694444.400000	758903.000000
-284588.600000	-2845886.000000	379451.500000	-1388889.000000
-94862.880000	-254184.300000	138888.900000	1016737.000000
-76255.300000	-762553.000000	101673.700000	1388889.000000
-25418.430000	-948628.800000	-138888.900000	3794515.000000
-1138355.000000	-569177.300000	-555555.600000	758903.000000
-101673.700000	-189725.800000	406694.900000	277777.800000
-305021.200000	-152510.600000	555555.600000	203347.500000
-379451.500000	-50836.870000	1517806.000000	-277777.800000
4.422650	2.113249E-01	474314.400000	-50836.870000
-1897258.000000	-277777.800000	1422943.000000	-569177.300000
694444.500000	758903.000000	127092.200000	-189725.800000
-508368.700000	277777.800000	381276.500000	-152510.600000
-694444.600000	203347.500000	94862.880000	-254184.400000
-379451.500000	-1388889.000000	284588.700000	-2845887.000000
138888.900000	3794515.000000	25418.440000	-948628.800000
-101673.700000	1388889.000000	76255.300000	-762553.100000
-138888.900000	1016737.000000	-101673.700000	189725.800000
-555555.600000	-758903.100000	-1138355.000000	569177.300000
1517806.000000	277777.800000	-379451.500000	50836.870000
555555.600000	-203347.500000	-305021.200000	152510.600000
406694.900000	-277777.800000	5.577350	2.113249E-01
381276.500000	152510.600000	-508368.700000	-277777.800000
127092.200000	189725.800000	694444.500000	-758903.000000
1422943.000000	569177.300000	-1897258.000000	277777.800000
474314.400000	50836.870000	-694444.600000	-203347.500000
76255.300000	762553.100000	-101673.700000	-1388889.000000
25418.440000	948628.800000	138888.900000	-3794515.000000
284588.700000	2845887.000000	-379451.500000	1388889.000000
94862.880000	254184.400000	-138888.900000	-1016737.000000
305021.200000	152510.600000	-555555.600000	-203347.500000
379451.500000	50836.870000	-1517806.000000	277777.800000
1138355.000000	569177.300000	555555.600000	-758903.100000
101673.700000	189725.800000	-406694.900000	-277777.800000
5.577350	7.886751E-01	-127092.200000	189725.800000
508368.700000	-277777.800000	-381276.500000	152510.600000
694444.500000	-203347.500000	-474314.400000	50836.860000
1897258.000000	277777.800000	-1422943.000000	569177.300000
-694444.400000	-758903.000000	-25418.440000	948628.800000
101673.700000	-1388889.000000	-76255.300000	762553.000000
138888.900000	-1016737.000000	-94862.880000	254184.300000
379451.500000	1388889.000000	-284588.600000	2845886.000000
-138888.900000	-3794515.000000	379451.500000	-50836.870000

-555555.600000	203347.500000	305021.200000	-152510.600000
-406694.900000	277777.800000	101673.700000	-189725.800000
555555.600000	758903.000000	1138355.000000	-569177.300000
-1517806.000000	-277777.800000	4.422650	7.886751E-01
35	36	29	30
25	26	23	24
21	22	27	28
31	32	33	34
-1422943.000000	-569177.300000	1897258.000000	-277777.800000
-474314.400000	-50836.870000	694444.500000	203347.500000
-381276.500000	-152510.600000	508368.700000	277777.800000
-127092.200000	-189725.800000	-694444.400000	758903.000000
-284588.600000	-2845886.000000	379451.500000	-1388889.000000
-94862.880000	-254184.300000	138888.900000	1016737.000000
-76255.300000	-762553.000000	101673.700000	1388889.000000
-25418.430000	-948628.800000	-138888.900000	3794515.000000
-1138355.000000	-569177.300000	-555555.600000	758903.000000
-101673.700000	-189725.800000	406694.900000	277777.800000
-305021.200000	-152510.600000	555555.600000	203347.500000
-379451.500000	-50836.870000	1517806.000000	-277777.800000
2.422650	2.113249E-01	474314.400000	-50836.870000
-1897258.000000	-277777.800000	1422943.000000	-569177.300000
694444.400000	758903.000000	127092.200000	-189725.800000
-508368.700000	277777.800000	381276.500000	-152510.600000
-694444.500000	203347.500000	94862.880000	-254184.300000
-379451.500000	-1388889.000000	284588.600000	-2845886.000000
138888.900000	3794515.000000	25418.430000	-948628.800000
-101673.700000	1388889.000000	76255.300000	-762553.000000
-138888.900000	1016737.000000	-101673.700000	189725.800000
-555555.600000	-758903.000000	-1138355.000000	569177.300000
1517806.000000	277777.800000	-379451.500000	50836.870000
555555.600000	-203347.500000	-305021.200000	152510.600000
406694.900000	-277777.800000	3.577350	2.113249E-01
381276.500000	152510.600000	-508368.700000	-277777.800000
127092.200000	189725.800000	694444.400000	-758903.000000
1422943.000000	569177.300000	-1897258.000000	277777.800000
474314.400000	50836.870000	-694444.500000	-203347.500000
76255.300000	762553.000000	-101673.700000	-1388889.000000
25418.440000	948628.800000	138888.900000	-3794515.000000
284588.600000	2845886.000000	-379451.500000	1388889.000000
94862.880000	254184.300000	-138888.900000	-1016737.000000
305021.200000	152510.600000	-555555.600000	-203347.500000
379451.500000	50836.870000	-1517806.000000	277777.800000
1138355.000000	569177.300000	555555.600000	-758903.000000
101673.700000	189725.800000	-406694.900000	-277777.800000
3.577350	7.886751E-01	-127092.200000	189725.800000
508368.700000	-277777.800000	-381276.500000	152510.600000
694444.500000	-203347.500000	-474314.400000	50836.870000
1897258.000000	277777.800000	-1422943.000000	569177.300000
-694444.400000	-758903.000000	-25418.440000	948628.800000
101673.700000	-1388889.000000	-76255.300000	762553.000000
138888.900000	-1016737.000000	-94862.880000	254184.300000
379451.500000	1388889.000000	-284588.600000	2845886.000000
-138888.900000	-3794515.000000	379451.500000	-50836.870000

-555555.600000	203347.500000	305021.200000	-152510.600000
-406694.900000	277777.800000	101673.700000	-189725.800000
555555.600000	758903.000000	1138355.000000	-569177.300000
-1517806.000000	-277777.800000	2.422650	7.886751E-01
0	0	39	40
35	36	33	34
31	32	37	38
0	0	0	0
-1422943.000000	-569177.300000	1897258.000000	-277777.800000
-474314.400000	-50836.870000	694444.500000	203347.500000
-381276.500000	-152510.600000	508368.700000	277777.800000
-127092.200000	-189725.800000	-694444.400000	758903.000000
-284588.600000	-2845886.000000	379451.500000	-1388889.000000
-94862.880000	-254184.300000	138888.900000	1016737.000000
-76255.300000	-762553.000000	101673.700000	1388889.000000
-25418.430000	-948628.800000	-138888.900000	3794515.000000
-1138355.000000	-569177.300000	-555555.600000	758903.000000
-101673.700000	-189725.800000	406694.900000	277777.800000
-305021.200000	-152510.600000	555555.600000	203347.500000
-379451.500000	-50836.870000	1517806.000000	-277777.800000
4.226497E-01	2.113249E-01	474314.400000	-50836.870000
-1897258.000000	-277777.800000	1422943.000000	-569177.300000
694444.400000	758903.000000	127092.200000	-189725.800000
-508368.700000	277777.800000	381276.500000	-152510.600000
-694444.500000	203347.500000	94862.880000	-254184.300000
-379451.500000	-1388889.000000	284588.600000	-2845886.000000
138888.900000	3794515.000000	25418.430000	-948628.800000
-101673.700000	1388889.000000	76255.300000	-762553.000000
-138888.900000	1016737.000000	-101673.700000	189725.800000
-555555.600000	-758903.000000	-1138355.000000	569177.300000
1517806.000000	277777.800000	-379451.500000	50836.870000
555555.600000	-203347.500000	-305021.200000	152510.600000
406694.900000	-277777.800000	1.577350	2.113249E-01
381276.500000	152510.600000	-508368.700000	-277777.800000
127092.200000	189725.800000	694444.400000	-758903.000000
1422943.000000	569177.300000	-1897258.000000	277777.800000
474314.400000	50836.870000	-694444.500000	-203347.500000
76255.300000	762553.000000	-101673.700000	-1388889.000000
25418.440000	948628.800000	138888.900000	-3794515.000000
284588.600000	2845886.000000	-379451.500000	1388889.000000
94862.880000	254184.300000	-138888.900000	-1016737.000000
305021.200000	152510.600000	-555555.600000	-203347.500000
379451.500000	50836.870000	-1517806.000000	277777.800000
1138355.000000	569177.300000	555555.600000	-758903.000000
101673.700000	189725.800000	-406694.900000	-277777.800000
1.577350	7.886751E-01	-127092.200000	189725.800000
508368.700000	-277777.800000	-381276.500000	152510.600000
694444.500000	-203347.500000	-474314.400000	50836.870000
1897258.000000	277777.800000	-1422943.000000	569177.300000
-694444.400000	-758903.000000	-25418.440000	948628.800000
101673.700000	-1388889.000000	-76255.300000	762553.000000
138888.900000	-1016737.000000	-94862.880000	254184.300000
379451.500000	1388889.000000	-284588.600000	2845886.000000
-138888.900000	-3794515.000000	379451.500000	-50836.870000

-555555.600000	203347.500000	305021.200000	-152510.600000
-406694.900000	277777.800000	101673.700000	-189725.800000
555555.600000	758903.000000	1138355.000000	-569177.300000
-1517806.000000	-277777.800000	4.226497E-01	7.886751E-01

ARCHIVO B.5. DATOS PARA CONSULTA EN PANTALLA:
 A:\EJEMPLO2.CNS
 GENERADO POR EL PREPROCESADOR.

No.	PUNTOS NODALES	
	X	Y
1	10.00	2.00
2	10.00	1.50
3	10.00	1.00
4	10.00	0.50
5	10.00	0.00
6	9.00	2.00
7	9.00	1.00
8	9.00	0.00
9	8.00	2.00
10	8.00	1.50
11	8.00	1.00
12	8.00	0.50
13	8.00	0.00
14	6.50	1.00
15	6.50	0.00
16	5.00	1.00
17	5.00	0.50
18	5.00	0.00
19	3.50	1.00
20	3.50	0.00
21	2.00	2.00
22	2.00	1.50
23	2.00	1.00
24	2.00	0.50
25	2.00	0.00
26	1.00	2.00
27	1.00	1.00
28	1.00	0.00
29	0.00	2.00
30	0.00	1.50
31	0.00	1.00
32	0.00	0.50
33	0.00	0.00

ELE.	ELEMENTOS								
	NOD I	NOD J	NOD K	NOD L	NOD M	NOD N	NOD O	NOD P	MATERIAL
1	11	7	3	2	1	6	9	10	1
2	31	27	23	22	21	26	29	30	1
3	13	8	5	4	3	7	11	12	1
4	18	15	13	12	11	14	16	17	1
5	25	20	18	17	16	19	23	24	1
6	33	28	25	24	23	27	31	32	1

CONDICIONES DE FRONTERA

No.	DX	DY
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	1	1
30	1	1
31	1	1
32	1	1
33	1	1

MATERIALES

No.	E	v	Espesor
1	2000000.00	0.20	2.40

NODOS CARGADOS

CARGA	NODO
1	11
2	16
3	23

CARGAS

No.	FX	FY
1	0.00	-15.00
2	0.00	-15.00
3	0.00	-15.00

ARCHIVO B.6. DATOS PARA EL PROCESADOR:
A:\EJEMPLO2.FMT
GENERADO POR EL PREPROCESADOR.

33
6
1
3
3
10.00
10.00
10.00
10.00
10.00
9.00
9.00
9.00
8.00
8.00
8.00
8.00
8.00
6.50
6.50
5.00
5.00
5.00
3.50
3.50
2.00
2.00
2.00
2.00
2.00
1.00
1.00
1.00
0.00
0.00
0.00
0.00
0.00
0.00
2.00
1.50
1.00
0.50
0.00
2.00
1.00
0.00
2.00
1.50
1.00
0.50
0.00

2
22
4
12
17
24
1
21
3
11
16
23
6
26
7
14
19
27
9
29
11
16
23
31
10
30
12
17
24
32
1
1
1
1
1
1
11
16
23
0.00
0.00
0.00
-15.00
-15.00
-15.00

ARCHIVO B.7. DATOS PARA EL USUARIO:

A:\RESULTA2

GENERADO POR EL PROCESADOR.

ELEMENTO NO. - 1

MATRIZ B

	1	2	3	4	5
1	-6.83013E-01	0.00000E+00	9.10684E-01	0.00000E+00	-2.27671E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-6.83013E-01	-6.66667E-01	9.10684E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-1.83013E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-2.27671E-01	4.88034E-01	3.33333E-01	-3.66025E-01	-1.83013E-01
	11	12	13	14	15
1	2.44017E-01	0.00000E+00	-6.10042E-02	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	2.44017E-01	-4.55342E-01	-6.10042E-02	1.82137E+00
	16				
1	0.00000E+00				
2	1.82137E+00				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	2.27671E-01	0.00000E+00	-9.10684E-01	0.00000E+00	6.83013E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	2.27671E-01	-6.66667E-01	-9.10684E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.10042E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	6.83013E-01	1.82137E+00	3.33333E-01	-4.55342E-01	6.10042E-02
	11	12	13	14	15
1	-2.44017E-01	0.00000E+00	1.83013E-01	0.00000E+00	-3.33334E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-2.44017E-01	-3.66025E-01	1.83013E-01	4.88034E-01
	16				
1	0.00000E+00				
2	4.88034E-01				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	1.83013E-01	0.00000E+00	-2.44017E-01	0.00000E+00	6.10042E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.83013E-01	-6.66667E-01	-2.44017E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.83013E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	6.10042E-02	-1.82137E+00	3.33333E-01	1.36603E+00	6.83013E-01
	11	12	13	14	15
1	-9.10684E-01	0.00000E+00	2.27671E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-9.10684E-01	1.22009E-01	2.27671E-01	-4.88034E-01

16
 1 0.00000E+00
 2 -4.88034E-01
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	-6.10042E-02	0.00000E+00	2.44017E-01	0.00000E+00	-1.83013E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-6.10042E-02	-6.66667E-01	2.44017E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-2.27671E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.83013E-01	-4.88034E-01	3.33333E-01	1.22009E-01	-2.27671E-01
	11	12	13	14	15
1	9.10684E-01	0.00000E+00	-6.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	9.10684E-01	1.36603E+00	-6.83013E-01	-1.82137E+00

16
 1 0.00000E+00
 2 -1.82137E+00
 3 -3.33333E-01

MATRIZ DE RIGIDECES DEL ELEMENTO 1

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-2.22222E+05	-1.25000E+05	1.87500E+05
2	1.77083E+05	7.63889E+05	-4.16667E+04	2.77778E+04	-1.04167E+04
3	-2.22222E+05	-4.16667E+04	7.77778E+05	-2.19152E-02	-2.22222E+05
4	-1.25000E+05	2.77778E+04	-2.42589E-02	7.77778E+05	1.25000E+05
5	1.87500E+05	-1.04167E+04	-2.22222E+05	1.25000E+05	4.51389E+05
6	1.04167E+04	2.50000E+05	4.16667E+04	2.77777E+04	-1.77083E+05
7	-2.56945E+05	-4.16667E+04	2.45011E-02	-1.66667E+05	-4.09722E+05
8	-4.16667E+04	-5.69445E+05	-1.66667E+05	5.60067E-02	4.16667E+04
9	2.25695E+05	7.29167E+04	-1.94445E+05	-4.16667E+04	2.18750E+05
10	7.29167E+04	3.81945E+05	-4.16667E+04	-1.94445E+05	1.04167E+04
11	-1.94445E+05	-4.16667E+04	5.55556E+04	-3.04353E-03	-1.94445E+05
12	-4.16667E+04	-1.94445E+05	-4.17410E-03	-4.44444E+05	4.16667E+04
13	2.18750E+05	1.04167E+04	-1.94445E+05	4.16667E+04	2.25695E+05
14	-1.04167E+04	4.37500E+05	4.16667E+04	-1.94445E+05	-7.29167E+04
15	-4.09722E+05	-1.25000E+05	9.70868E-03	1.66667E+05	-2.56945E+05
16	-4.16667E+04	-1.09722E+06	1.66667E+05	-4.07597E-03	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05	7.29167E+04
2	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04	3.81945E+05
3	4.16667E+04	2.76261E-02	-1.66667E+05	-1.94445E+05	-4.16667E+04
4	2.77777E+04	-1.66667E+05	8.72567E-02	-4.16667E+04	-1.94445E+05
5	-1.77083E+05	-4.09722E+05	4.16667E+04	2.18750E+05	1.04167E+04
6	7.63889E+05	1.25000E+05	-1.09722E+06	-1.04167E+04	4.37500E+05
7	1.25000E+05	1.02778E+06	-4.30795E-04	-4.09722E+05	-1.25000E+05
8	-1.09722E+06	-4.30795E+06	2.27778E+06	-4.16667E+04	-1.09722E+06
9	-1.04167E+04	-4.09722E+05	-4.16667E+04	4.51389E+05	1.77083E+05
10	4.37500E+05	-1.25000E+05	-1.09722E+06	1.77083E+05	7.63889E+05
11	4.16667E+04	4.19972E-02	1.66667E+05	-2.22222E+05	-4.16667E+04
12	-1.94445E+05	1.66667E+05	5.29364E-02	-1.25000E+05	2.77777E+04

13	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05	-1.04167E+04
14	3.81945E+05	4.16667E+04	-5.69444E+05	1.04167E+04	2.50000E+05
15	4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05	-4.16667E+04
16	-5.69445E+05	-1.48103E-03	1.05556E+06	-4.16667E+04	-5.69445E+05
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05
2	-4.16667E+04	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05
3	5.55556E+04	5.13406E-04	-1.94445E+05	4.16667E+04	3.15837E-02
4	4.37835E-03	-4.44444E+05	4.16667E+04	-1.94445E+05	1.66667E+05
5	-1.94445E+05	4.16667E+04	2.25695E+05	-7.29167E+04	-2.56945E+05
6	4.16667E+04	-1.94445E+05	-7.29167E+04	3.81945E+05	4.16667E+04
7	3.73097E-02	1.66667E+05	-2.56945E+05	4.16667E+04	3.05556E+05
8	1.66667E+05	7.94989E-02	4.16667E+04	-5.69445E+05	-4.17410E-03
9	-2.22222E+05	-1.25000E+05	1.87500E+05	1.04167E+04	-2.56945E+05
10	-4.16667E+04	2.77777E+04	-1.04167E+04	2.50000E+05	-4.16667E+04
11	7.77778E+05	2.63984E-02	-2.22222E+05	4.16667E+04	2.45743E-02
12	2.63984E-02	7.77778E+05	1.25000E+05	2.77778E+04	-1.66667E+05
13	-2.22222E+05	1.25000E+05	4.51389E+05	-1.77083E+05	-4.09722E+05
14	4.16667E+04	2.77778E+04	-1.77083E+05	7.63889E+05	1.25000E+05
15	4.01993E-02	-1.66667E+05	-4.09722E+05	1.25000E+05	1.02778E+06
16	-1.66667E+05	-4.08585E-03	4.16667E+04	-1.09722E+06	-7.97659E-03
	16				
1	-4.16667E+04				
2	-1.09722E+06				
3	1.66667E+05				
4	-4.07597E-03				
5	4.16667E+04				
6	-5.69445E+05				
7	-4.99665E-03				
8	1.05556E+06				
9	-4.16667E+04				
10	-5.69445E+05				
11	-1.66667E+05				
12	-4.08585E-03				
13	4.16667E+04				
14	-1.09722E+06				
15	1.39841E-03				
16	2.27778E+06				

ELEMENTO NO. - 2

MATRIZ B

	1	2	3	4	5
1	-6.83013E-01	0.00000E+00	9.10684E-01	0.00000E+00	-2.27671E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-6.83013E-01	-6.66667E-01	9.10684E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-1.83013E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-2.27671E-01	4.88034E-01	3.33333E-01	-3.66025E-01	-1.83013E-01
	11	12	13	14	15
1	2.44017E-01	0.00000E+00	-6.10042E-02	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	2.44017E-01	-4.55342E-01	-6.10042E-02	1.82137E+00

16
 1 0.00000E+00
 2 1.82137E+00
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	2.27671E-01	0.00000E+00	-9.10684E-01	0.00000E+00	6.83013E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	2.27671E-01	-6.66667E-01	-9.10684E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.10042E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	6.83013E-01	1.82137E+00	3.33333E-01	-4.55342E-01	6.10042E-02
	11	12	13	14	15
1	-2.44017E-01	0.00000E+00	1.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-2.44017E-01	-3.66025E-01	1.83013E-01	4.88034E-01
	16				
1	0.00000E+00				
2	4.88034E-01				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	1.83013E-01	0.00000E+00	-2.44017E-01	0.00000E+00	6.10042E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.83013E-01	-6.66667E-01	-2.44017E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.83013E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	6.10042E-02	-1.82137E+00	3.33333E-01	1.36603E+00	6.83013E-01
	11	12	13	14	15
1	-9.10684E-01	0.00000E+00	2.27671E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-9.10684E-01	1.22009E-01	2.27671E-01	-4.88034E-01
	16				
1	0.00000E+00				
2	-4.88034E-01				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	-6.10042E-02	0.00000E+00	2.44017E-01	0.00000E+00	-1.83013E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-6.10042E-02	-6.66667E-01	2.44017E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-2.27671E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.83013E-01	-4.88034E-01	3.33333E-01	1.22009E-01	-2.27671E-01
	11	12	13	14	15
1	9.10684E-01	0.00000E+00	-6.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	9.10684E-01	1.36603E+00	-6.83013E-01	-1.82137E+00

16

1 0.00000E+00
 2 -1.82137E+00
 3 -3.33333E-01

MATRIZ DE RIGIDECES DEL ELEMENTO 2

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-2.22222E+05	-1.25000E+05	1.87500E+05
2	1.77083E+05	7.63889E+05	-4.16667E+04	2.77778E+04	-1.04167E+04
3	-2.22222E+05	-4.16667E+04	7.77778E+05	-4.30795E-04	-2.22222E+05
4	-1.25000E+05	2.77778E+04	1.13171E-03	7.77778E+05	1.25000E+05
5	1.87500E+05	-1.04167E+04	-2.22222E+05	1.25000E+05	4.51389E+05
6	1.04167E+04	2.50000E+05	4.16667E+04	2.77777E+04	-1.77083E+05
7	-2.56945E+05	-4.16667E+04	2.45011E-02	-1.66667E+05	-4.09722E+05
8	-4.16667E+04	-5.69445E+05	-1.66667E+05	4.03817E-02	4.16667E+04
9	2.25695E+05	7.29167E+04	-1.94445E+05	-4.16667E+04	2.18750E+05
10	7.29167E+04	3.81945E+05	-4.16667E+04	-1.94445E+05	1.04167E+04
11	-1.94445E+05	-4.16667E+04	5.55555E+04	3.59710E-03	-1.94445E+05
12	-4.16667E+04	-1.94445E+05	-1.04910E-03	-4.44445E+05	4.16667E+04
13	2.18750E+05	1.04167E+04	-1.94445E+05	4.16667E+04	2.25695E+05
14	-1.04167E+04	4.37500E+05	4.16667E+04	-1.94445E+05	-7.29167E+04
15	-4.09722E+05	-1.25000E+05	2.53337E-02	1.66667E+05	-2.56945E+05
16	-4.16667E+04	-1.09722E+06	1.66667E+05	5.84240E-02	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05	7.29167E+04
2	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04	3.81945E+05
3	4.16667E+04	3.23136E-02	-1.66667E+05	-1.94445E+05	-4.16667E+04
4	2.77777E+04	-1.66667E+05	4.03817E-02	-4.16667E+04	-1.94445E+05
5	-1.77083E+05	-4.09722E+05	4.16667E+04	2.18750E+05	1.04167E+04
6	7.63889E+05	1.25000E+05	-1.09722E+06	-1.04167E+04	4.37500E+05
7	1.25000E+05	1.02778E+06	1.48036E-02	-4.09722E+05	-1.25000E+05
8	-1.09722E+06	1.12880E-02	2.27778E+06	-4.16667E+04	-1.09722E+06
9	-1.04167E+04	-4.09722E+05	-4.16667E+04	4.51389E+05	1.77083E+05
10	4.37500E+05	-1.25000E+05	-1.09722E+06	1.77083E+05	7.63889E+05
11	4.16667E+04	2.63722E-02	1.66667E+05	-2.22222E+05	-4.16667E+04
12	-1.94445E+05	1.66667E+05	2.16864E-02	-1.25000E+05	2.77777E+04
13	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05	-1.04167E+04
14	3.81945E+05	4.16667E+04	-5.69445E+05	1.04167E+04	2.50000E+05
15	4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05	-4.16667E+04
16	-5.69445E+05	5.55023E-03	1.05556E+06	-4.16667E+04	-5.69445E+05
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05
2	-4.16667E+04	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05
3	5.55556E+04	3.63841E-03	-1.94445E+05	4.16667E+04	3.15837E-02
4	-1.87165E-03	-4.44445E+05	4.16667E+04	-1.94445E+05	1.66667E+05
5	-1.94445E+05	4.16667E+04	2.25695E+05	-7.29167E+04	-2.56945E+05
6	4.16667E+04	-1.94445E+05	-7.29167E+04	3.81945E+05	4.16667E+04
7	4.19972E-02	1.66667E+05	-2.56945E+05	4.16667E+04	3.05556E+05
8	1.66667E+05	2.16864E-02	4.16667E+04	-5.69445E+05	-4.17410E-03
9	-2.22222E+05	-1.25000E+05	1.87500E+05	1.04167E+04	-2.56945E+05
10	-4.16667E+04	2.77777E+04	-1.04167E+04	2.50000E+05	-4.16667E+04
11	7.77778E+05	-1.64086E-04	-2.22222E+05	4.16667E+04	2.76993E-02
12	1.39841E-03	7.77778E+05	1.25000E+05	2.77778E+04	-1.66667E+05
13	-2.22222E+05	1.25000E+05	4.51389E+05	-1.77083E+05	-4.09722E+05

14	4.16667E+04	2.77778E+04	-1.77083E+05	7.63889E+05	1.25000E+05
15	4.33243E-02	-1.66667E+05	-4.09722E+05	1.25000E+05	1.02778E+06
16	-1.66667E+05	6.46642E-02	4.16667E+04	-1.09722E+06	-7.97659E-03
16					
1	-4.16667E+04				
2	-1.09722E+06				
3	1.66667E+05				
4	5.84240E-02				
5	4.16667E+04				
6	-5.69445E+05				
7	2.03460E-03				
8	1.05556E+06				
9	-4.16667E+04				
10	-5.69445E+05				
11	-1.66667E+05				
12	6.46642E-02				
13	4.16667E+04				
14	-1.09722E+06				
15	1.39841E-03				
16	2.27778E+06				

ELEMENTO NO. - 3

MATRIZ B

	1	2	3	4	5
1	-6.83013E-01	0.00000E+00	9.10684E-01	0.00000E+00	-2.27671E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-6.83013E-01	-6.66667E-01	9.10684E-01	-1.22009E-01
6					
1	0.00000E+00	3.33333E-01	0.00000E+00	-1.83013E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-2.27671E-01	4.88034E-01	3.33333E-01	-3.66025E-01	-1.83013E-01
11					
1	2.44017E-01	0.00000E+00	-6.10042E-02	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	2.44017E-01	-4.55342E-01	-6.10042E-02	1.82137E+00
16					
1	0.00000E+00				
2	1.82137E+00				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	2.27671E-01	0.00000E+00	-9.10684E-01	0.00000E+00	6.83013E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	2.27671E-01	-6.66667E-01	-9.10684E-01	-1.36603E+00
6					
1	0.00000E+00	3.33333E-01	0.00000E+00	6.10042E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	6.83013E-01	1.82137E+00	3.33333E-01	-4.55342E-01	6.10042E-02
11					
1	-2.44017E-01	0.00000E+00	1.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-2.44017E-01	-3.66025E-01	1.83013E-01	4.88034E-01

16
 1 0.0000E+00
 2 4.8803E-01
 3 -3.3333E-01

MATRIZ B

	1	2	3	4	5
1	1.83013E-01	0.00000E+00	-2.44017E-01	0.00000E+00	6.10042E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.83013E-01	-6.66667E-01	-2.44017E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.83013E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	6.10042E-02	-1.82137E+00	3.33333E-01	1.36603E+00	6.83013E-01
	11	12	13	14	15
1	-9.10684E-01	0.00000E+00	2.27671E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-9.10684E-01	1.22009E-01	2.27671E-01	-4.88034E-01
	16				
1	0.00000E+00				
2	-4.88034E-01				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	-6.10042E-02	0.00000E+00	2.44017E-01	0.00000E+00	-1.83013E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-6.10042E-02	-6.66667E-01	2.44017E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-2.27671E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.83013E-01	-4.88034E-01	3.33333E-01	1.22009E-01	-2.27671E-01
	11	12	13	14	15
1	9.10684E-01	0.00000E+00	-6.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	9.10684E-01	1.36603E+00	-6.83013E-01	-1.82137E+00
	16				
1	0.00000E+00				
2	-1.82137E+00				
3	-3.33333E-01				

MATRIZ DE RIGIDECES DEL ELEMENTO 3

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-2.22222E+05	-1.25000E+05	1.87500E+05
2	1.77083E+05	7.63889E+05	-4.16667E+04	2.77778E+04	-1.04167E+04
3	-2.22222E+05	-4.16667E+04	7.77778E+05	-2.19152E-02	-2.22222E+05
4	-1.25000E+05	2.77778E+04	-2.42589E-02	7.77778E+05	1.25000E+05
5	1.87500E+05	-1.04167E+04	-2.22222E+05	1.25000E+05	4.51389E+05
6	1.04167E+04	2.50000E+05	4.16667E+04	2.77778E+04	-1.77083E+05
7	-2.56945E+05	-4.16667E+04	2.45011E-02	-1.66667E+05	-4.09722E+05
8	-4.16667E+04	-5.69445E+05	-1.66667E+05	5.60067E-02	4.16667E+04
9	2.25695E+05	7.29167E+04	-1.94445E+05	-4.16667E+04	2.18750E+05
10	7.29167E+04	3.81945E+05	-4.16667E+04	-1.94445E+05	1.04167E+04
11	-1.94445E+05	-4.16667E+04	5.55556E+04	-3.04353E-03	-1.94445E+05

12	-4.16667E+04	-1.94445E+05	-4.17410E-03	-4.44444E+05	4.16667E+04
13	2.18750E+05	1.04167E+04	-1.94445E+05	4.16667E+04	2.25695E+05
14	-1.04167E+04	4.37500E+05	4.16667E+04	-1.94445E+05	-7.29167E+04
15	-4.09722E+05	-1.25000E+05	4.09587E-02	1.66667E+05	-2.56945E+05
16	-4.16667E+04	-1.09722E+06	1.66667E+05	-4.07597E-03	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05	7.29167E+04
2	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04	3.81945E+05
3	4.16667E+04	2.76261E-02	-1.66667E+05	-1.94445E+05	-4.16667E+04
4	2.77777E+04	-1.66667E+05	8.72567E-02	-4.16667E+04	-1.94445E+05
5	-1.77083E+05	-4.09722E+05	4.16667E+04	2.18750E+05	1.04167E+04
6	7.63889E+05	1.25000E+05	-1.09722E+06	-1.04167E+04	4.37500E+05
7	1.25000E+05	1.02778E+06	-4.30795E-04	-4.09722E+05	-1.25000E+05
8	-1.09722E+06	-4.30795E-04	2.27778E+06	-4.16667E+04	-1.09722E+06
9	-1.04167E+04	-4.09722E+05	-4.16667E+04	4.51389E+05	1.77083E+05
10	4.37500E+05	-1.25000E+05	-1.09722E+06	1.77083E+05	7.63889E+05
11	4.16667E+04	4.19972E-02	1.66667E+05	-2.22222E+05	-4.16667E+04
12	-1.94445E+05	1.66667E+05	8.41864E-02	-1.25000E+05	2.77777E+04
13	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05	-1.04167E+04
14	3.81945E+05	4.16667E+04	-5.69444E+05	1.04167E+04	2.50000E+05
15	4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05	-4.16667E+04
16	-5.69445E+05	2.42523E-03	1.05556E+06	-4.16667E+04	-5.69445E+05
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05
2	-4.16667E+04	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05
3	5.55556E+04	5.13406E-04	-1.94445E+05	4.16667E+04	3.15837E-02
4	4.37835E-03	-4.44444E+05	4.16667E+04	-1.94445E+05	1.66667E+05
5	-1.94445E+05	4.16667E+04	2.25695E+05	-7.29167E+04	-2.56945E+05
6	4.16667E+04	-1.94445E+05	-7.29167E+04	3.81945E+05	4.16667E+04
7	3.73097E-02	1.66667E+05	-2.56945E+05	4.16667E+04	3.05556E+05
8	1.66667E+05	7.94989E-02	4.16667E+04	-5.69445E+05	3.63841E-03
9	-2.22222E+05	-1.25000E+05	1.87500E+05	1.04167E+04	-2.56945E+05
10	-4.16667E+04	2.77777E+04	-1.04167E+04	2.50000E+05	-4.16667E+04
11	7.77778E+05	2.63984E-02	-2.22222E+05	4.16667E+04	2.45743E-02
12	2.63984E-02	7.77778E+05	1.25000E+05	2.77778E+04	-1.66667E+05
13	-2.22222E+05	1.25000E+05	4.51389E+05	-1.77083E+05	-4.09722E+05
14	4.16667E+04	2.77778E+04	-1.77083E+05	7.63889E+05	1.25000E+05
15	4.01993E-02	-1.66667E+05	-4.09722E+05	1.25000E+05	1.02778E+06
16	-1.66667E+05	-4.08585E-03	4.16667E+04	-1.09722E+06	7.64841E-03
	16				
1	-4.16667E+04				
2	-1.09722E+06				
3	1.66667E+05				
4	-4.07597E-03				
5	4.16667E+04				
6	-5.69445E+05				
7	-1.09040E-03				
8	1.05556E+06				
9	-4.16667E+04				
10	-5.69445E+05				
11	-1.66667E+05				
12	-4.08585E-03				
13	4.16667E+04				
14	-1.09722E+06				

15 1.398414E-03
 16 2.277778E+06

ELEMENTO NO. - 4
 MATRIZ B

	1	2	3	4	5
1	-4.55342E-01	0.00000E+00	6.07122E-01	0.00000E+00	-1.51781E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-4.55342E-01	-6.66667E-01	6.07122E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	2.22222E-01	0.00000E+00	-1.22009E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-1.51781E-01	4.88034E-01	2.22222E-01	-3.66025E-01	-1.22009E-01
	11	12	13	14	15
1	1.62678E-01	0.00000E+00	-4.06695E-02	0.00000E+00	-2.22222E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	1.62678E-01	-4.55342E-01	-4.06695E-02	1.82137E+00
	16				
1	0.00000E+00				
2	1.82137E+00				
3	-2.22222E-01				

MATRIZ B

	1	2	3	4	5
1	1.51781E-01	0.00000E+00	-6.07122E-01	0.00000E+00	4.55342E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	1.51781E-01	-6.66667E-01	-6.07122E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	2.22222E-01	0.00000E+00	4.06695E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	4.55342E-01	1.82137E+00	2.22222E-01	-4.55342E-01	4.06695E-02
	11	12	13	14	15
1	-1.62678E-01	0.00000E+00	1.22009E-01	0.00000E+00	-2.22222E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-1.62678E-01	-3.66025E-01	1.22009E-01	4.88034E-01
	16				
1	0.00000E+00				
2	4.88034E-01				
3	-2.22222E-01				

MATRIZ B

	1	2	3	4	5
1	1.22009E-01	0.00000E+00	-1.62678E-01	0.00000E+00	4.06695E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.22009E-01	-6.66667E-01	-1.62678E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	2.22222E-01	0.00000E+00	4.55342E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	4.06695E-02	-1.82137E+00	2.22222E-01	1.36603E+00	4.55342E-01
	11	12	13	14	15
1	-6.07122E-01	0.00000E+00	1.51781E-01	0.00000E+00	-2.22222E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-6.07122E-01	1.22009E-01	1.51781E-01	-4.88034E-01

16
 1 0.00000E+00
 2 -4.88034E-01
 3 -2.22222E-01

MATRIZ B

	1	2	3	4	5
1	-4.06695E-02	0.00000E+00	1.62678E-01	0.00000E+00	-1.22009E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-4.06695E-02	-6.66667E-01	1.62678E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	2.22222E-01	0.00000E+00	-1.51781E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.22009E-01	-4.88034E-01	2.22222E-01	1.22009E-01	-1.51781E-01
	11	12	13	14	15
1	6.07122E-01	0.00000E+00	-4.55342E-01	0.00000E+00	-2.22222E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	6.07122E-01	1.36603E+00	-4.55342E-01	-1.82137E+00
	16				
1	0.00000E+00				
2	-1.82137E+00				
3	-2.22222E-01				

MATRIZ DE RIGIDECES DEL ELEMENTO 4

	1	2	3	4	5
1	5.32407E+05	1.77083E+05	-1.01852E+05	-1.25000E+05	1.94445E+05
2	1.77083E+05	1.08796E+06	-4.16667E+04	1.34259E+05	-1.04167E+04
3	-1.01852E+05	-4.16667E+04	7.03704E+05	-9.80580E-03	-1.01852E+05
4	-1.25000E+05	1.34259E+05	-3.55580E-03	9.81482E+05	1.25000E+05
5	1.94445E+05	-1.04167E+04	-1.01852E+05	1.25000E+05	5.32407E+05
6	1.04167E+04	3.40278E+05	4.16667E+04	1.34259E+05	-1.77083E+05
7	-3.56482E+05	-4.16667E+04	4.76287E-02	-1.66667E+05	-6.43519E+05
8	-4.16667E+04	-8.42593E+05	-1.66667E+05	2.66421E-02	4.16667E+04
9	2.66204E+05	7.29167E+04	-1.75926E+05	-4.16667E+04	2.84722E+05
10	7.29167E+04	5.43982E+05	-4.16667E+04	-2.45370E+05	1.04167E+04
11	-1.75926E+05	-4.16667E+04	-1.48148E+05	2.76788E-04	-1.75926E+05
12	-4.16667E+04	-2.45370E+05	5.98216E-03	-7.59259E+05	4.16667E+04
13	2.84722E+05	1.04167E+04	-1.75926E+05	4.16667E+04	2.66204E+05
14	-1.04167E+04	6.38889E+05	4.16667E+04	-2.45370E+05	-7.29167E+04
15	-6.43519E+05	-1.25000E+05	2.39985E-02	1.66667E+05	-3.56482E+05
16	-4.16667E+04	-1.65741E+06	1.66667E+05	5.35977E-02	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-3.56482E+05	-4.16667E+04	2.66204E+05	7.29167E+04
2	3.40278E+05	-4.16667E+04	-8.42593E+05	7.29167E+04	5.43982E+05
3	4.16667E+04	2.49725E-02	-1.66667E+05	-1.75926E+05	-4.16667E+04
4	1.34259E+05	-1.66667E+05	7.50796E-02	-4.16667E+04	-2.45370E+05
5	-1.77083E+05	-6.43519E+05	4.16667E+04	2.84722E+05	1.04167E+04
6	1.08796E+06	1.25000E+05	-1.65741E+06	-1.04167E+04	6.38889E+05
7	1.25000E+05	1.42593E+06	-5.89955E-03	-6.43519E+05	-1.25000E+05
8	-1.65741E+06	6.20983E-03	3.37037E+06	-4.16667E+04	-1.65741E+06
9	-1.04167E+04	-6.43519E+05	-4.16667E+04	5.32407E+05	1.77083E+05
10	6.38889E+05	-1.25000E+05	-1.65741E+06	1.77083E+05	1.08796E+06
11	4.16667E+04	3.08506E-02	1.66667E+05	-1.01852E+05	-4.16667E+04
12	-2.45370E+05	1.66667E+05	6.68200E-02	-1.25000E+05	1.34259E+05

13	-7.29167E+04	-3.56482E+05	4.16667E+04	1.94445E+05	-1.04167E+04
14	5.43982E+05	4.16667E+04	-8.42593E+05	1.04167E+04	3.40278E+05
15	4.16667E+04	5.74074E+05	-1.04910E-03	-3.56482E+05	-4.16667E+04
16	-8.42593E+05	2.42523E-03	1.62963E+06	-4.16667E+04	-8.42593E+05
	11	12	13	14	15
1	-1.75926E+05	-4.16667E+04	2.84722E+05	-1.04167E+04	-6.43519E+05
2	-4.16667E+04	-2.45370E+05	1.04167E+04	6.38889E+05	-1.25000E+05
3	-1.48148E+05	-1.83035E-03	-1.75926E+05	4.16667E+04	9.93597E-03
4	-6.16853E-03	-7.59259E+05	4.16667E+04	-2.45370E+05	1.66667E+05
5	-1.75926E+05	4.16667E+04	2.66204E+05	-7.29167E+04	-3.56482E+05
6	4.16667E+04	-2.45370E+05	-7.29167E+04	5.43982E+05	4.16667E+04
7	5.92442E-02	1.66667E+05	-3.56482E+05	4.16667E+04	5.74074E+05
8	1.66667E+05	6.05700E-02	4.16667E+04	-8.42593E+05	-1.04910E-03
9	-1.01852E+05	-1.25000E+05	1.94445E+05	1.04167E+04	-3.56482E+05
10	-4.16667E+04	1.34259E+05	-1.04167E+04	3.40278E+05	-4.16667E+04
11	7.03704E+05	1.15547E-02	-1.01852E+05	4.16667E+04	1.03152E-02
12	3.74216E-03	9.81482E+05	1.25000E+05	1.34259E+05	-1.66667E+05
13	-1.01852E+05	1.25000E+05	5.32407E+05	-1.77083E+05	-6.43519E+05
14	4.16667E+04	1.34259E+05	-1.77083E+05	1.08796E+06	1.25000E+05
15	4.15652E-02	-1.66667E+05	-6.43519E+05	1.25000E+05	1.42593E+06
16	-1.66667E+05	8.26277E-02	4.16667E+04	-1.65741E+06	-3.28909E-03
	16				
1	-4.16667E+04				
2	-1.65741E+06				
3	1.66667E+05				
4	5.35977E-02				
5	4.16667E+04				
6	-8.42593E+05				
7	2.42523E-03				
8	1.62963E+06				
9	-4.16667E+04				
10	-8.42593E+05				
11	-1.66667E+05				
12	8.26277E-02				
13	4.16667E+04				
14	-1.65741E+06				
15	-3.28909E-03				
16	3.37037E+06				

ELEMENTO NO. - 5
MATRIZ B

	1	2	3	4	5
1	-4.55342E-01	0.00000E+00	6.07122E-01	0.00000E+00	-1.51781E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-4.55342E-01	-6.66667E-01	6.07122E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	2.22222E-01	0.00000E+00	-1.22009E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-1.51781E-01	4.88034E-01	2.22222E-01	-3.66025E-01	-1.22009E-01
	11	12	13	14	15
1	1.62678E-01	0.00000E+00	-4.06695E-02	0.00000E+00	-2.22222E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	1.62678E-01	-4.55342E-01	-4.06695E-02	1.82137E+00

16

1 0.00000E+00
 2 1.82137E+00
 3 -2.22222E-01

MATRIZ B

	1	2	3	4	5
1	1.51781E-01	0.00000E+00	-6.07122E-01	0.00000E+00	4.55342E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	1.51781E-01	-6.66667E-01	-6.07122E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	2.22222E-01	0.00000E+00	4.06695E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	4.55342E-01	1.82137E+00	2.22222E-01	-4.55342E-01	4.06695E-02
	11	12	13	14	15
1	-1.62678E-01	0.00000E+00	1.22009E-01	0.00000E+00	-2.22222E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-1.62678E-01	-3.66025E-01	1.22009E-01	4.88034E-01

16

1 0.00000E+00
 2 4.88034E-01
 3 -2.22222E-01

MATRIZ B

	1	2	3	4	5
1	1.22009E-01	0.00000E+00	-1.62678E-01	0.00000E+00	4.06695E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.22009E-01	-6.66667E-01	-1.62678E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	2.22222E-01	0.00000E+00	4.55342E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	4.06695E-02	-1.82137E+00	2.22222E-01	1.36603E+00	4.55342E-01
	11	12	13	14	15
1	-6.07122E-01	0.00000E+00	1.51781E-01	0.00000E+00	-2.22222E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-6.07122E-01	1.22009E-01	1.51781E-01	-4.88034E-01

16

1 0.00000E+00
 2 -4.88034E-01
 3 -2.22222E-01

MATRIZ B

	1	2	3	4	5
1	-4.06695E-02	0.00000E+00	1.62678E-01	0.00000E+00	-1.22009E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-4.06695E-02	-6.66667E-01	1.62678E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	2.22222E-01	0.00000E+00	-1.51781E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.22009E-01	-4.88034E-01	2.22222E-01	1.22009E-01	-1.51781E-01
	11	12	13	14	15
1	6.07122E-01	0.00000E+00	-4.55342E-01	0.00000E+00	-2.22222E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	6.07122E-01	1.36603E+00	-4.55342E-01	-1.82137E+00

16
 1 0.00000E+00
 2 -1.82137E+00
 3 -2.22222E-01

MATRIZ DE RIGIDECES DEL ELEMENTO 5

	1	2	3	4	5
1	5.32407E+05	1.77083E+05	-1.01852E+05	-1.25000E+05	1.94445E+05
2	1.77083E+05	1.08796E+06	-4.16667E+04	1.34259E+05	-1.04167E+04
3	-1.01852E+05	-4.16667E+04	7.03704E+05	-1.99330E-03	-1.01852E+05
4	-1.25000E+05	1.34259E+05	-3.55580E-03	9.81482E+05	1.25000E+05
5	1.94445E+05	-1.04167E+04	-1.01852E+05	1.25000E+05	5.32407E+05
6	1.04167E+04	3.40278E+05	4.16667E+04	1.34259E+05	-1.77083E+05
7	-3.56482E+05	-4.16667E+04	4.76287E-02	-1.66667E+05	-6.43519E+05
8	-4.16667E+04	-8.42593E+05	-1.66667E+05	2.66421E-02	4.16667E+04
9	2.66204E+05	7.29167E+04	-1.75926E+05	-4.16667E+04	2.84722E+05
10	7.29167E+04	5.43982E+05	-4.16667E+04	-2.45370E+05	1.04167E+04
11	-1.75926E+05	-4.16667E+04	-1.48148E+05	-1.48103E-03	-1.75926E+05
12	-4.16667E+04	-2.45370E+05	-1.04910E-03	-7.59259E+05	4.16667E+04
13	2.84722E+05	1.04167E+04	-1.75926E+05	4.16667E+04	2.66204E+05
14	-1.04167E+04	6.38889E+05	4.16667E+04	-2.45370E+05	-7.29167E+04
15	-6.43519E+05	-1.25000E+05	5.52485E-02	1.66667E+05	-3.56482E+05
16	-4.16667E+04	-1.65741E+06	1.66667E+05	5.35977E-02	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-3.56482E+05	-4.16667E+04	2.66204E+05	7.29167E+04
2	3.40278E+05	-4.16667E+04	-8.42593E+05	7.29167E+04	5.43982E+05
3	4.16667E+04	2.49725E-02	-1.66667E+05	-1.75926E+05	-4.16667E+04
4	1.34259E+05	-1.66667E+05	7.50796E-02	-4.16667E+04	-2.45370E+05
5	-1.77083E+05	-6.43519E+05	4.16667E+04	2.84722E+05	1.04167E+04
6	1.08796E+06	1.25000E+05	-1.65741E+06	-1.04167E+04	6.38889E+05
7	1.25000E+05	1.42593E+06	-5.89955E-03	-6.43519E+05	-1.25000E+05
8	-1.65741E+06	6.20983E-03	3.37037E+06	-4.16667E+04	-1.65741E+06
9	-1.04167E+04	-6.43519E+05	-4.16667E+04	5.32407E+05	1.77083E+05
10	6.38889E+05	-1.25000E+05	-1.65741E+06	1.77083E+05	1.08796E+06
11	4.16667E+04	3.63438E-02	1.66667E+05	-1.01852E+05	-4.16667E+04
12	-2.45370E+05	1.66667E+05	8.08825E-02	-1.25000E+05	1.34259E+05
13	-7.29167E+04	-3.56482E+05	4.16667E+04	1.94445E+05	-1.04167E+04
14	5.43982E+05	4.16667E+04	-8.42593E+05	1.04167E+04	3.40278E+05
15	4.16667E+04	5.74074E+05	-1.04910E-03	-3.56482E+05	-4.16667E+04
16	-8.42593E+05	2.42523E-03	1.62963E+06	-4.16667E+04	-8.42593E+05
	11	12	13	14	15
1	-1.75926E+05	-4.16667E+04	2.84722E+05	-1.04167E+04	-6.43519E+05
2	-4.16667E+04	-2.45370E+05	1.04167E+04	6.38889E+05	-1.25000E+05
3	-1.48148E+05	-1.83035E-03	-1.75926E+05	4.16667E+04	4.11860E-02
4	-1.13837E-04	-7.59259E+05	4.16667E+04	-2.45370E+05	1.66667E+05
5	-1.75926E+05	4.16667E+04	2.66204E+05	-7.29167E+04	-3.56482E+05
6	4.16667E+04	-2.45370E+05	-7.29167E+04	5.43982E+05	4.16667E+04
7	6.60557E-02	1.66667E+05	-3.56482E+05	4.16667E+04	5.74074E+05
8	1.66667E+05	7.46325E-02	4.16667E+04	-8.42593E+05	-1.04910E-03
9	-1.01852E+05	-1.25000E+05	1.94445E+05	1.04167E+04	3.56482E+05
10	-4.16667E+04	1.34259E+05	-1.04167E+04	3.40278E+05	-4.16667E+04
11	7.03704E+05	-2.50784E-03	-1.01852E+05	4.16667E+04	3.84402E-02
12	-3.28909E-03	9.81482E+05	1.25000E+05	1.34259E+05	-1.66667E+05
13	-1.01852E+05	1.25000E+05	5.32407E+05	-1.77083E+05	-6.43519E+05

14	4.16667E+04	1.34259E+05	-1.77083E+05	1.08796E+06	1.25000E+05
15	6.96902E-02	-1.66667E+05	-6.43519E+05	1.25000E+05	1.42593E+06
16	-1.66667E+05	8.26277E-02	4.16667E+04	-1.65741E+06	-3.28909E-03
	16				
1	-4.16667E+04				
2	-1.65741E+06				
3	1.66667E+05				
4	1.16098E-01				
5	4.16667E+04				
6	-8.42593E+05				
7	2.42523E-03				
8	1.62963E+06				
9	-4.16667E+04				
10	-8.42593E+05				
11	-1.66667E+05				
12	1.38878E-01				
13	4.16667E+04				
14	-1.65741E+06				
15	-3.28909E-03				
16	3.37037E+06				

ELEMENTO NO.- 6

MATRIZ B

	1	2	3	4	5
1	-6.83013E-01	0.00000E+00	9.10684E-01	0.00000E+00	-2.27671E-01
2	0.00000E+00	-1.36603E+00	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.36603E+00	-6.83013E-01	-6.66667E-01	9.10684E-01	-1.22009E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-1.83013E-01	0.00000E+00
2	-1.22009E-01	0.00000E+00	4.88034E-01	0.00000E+00	-3.66025E-01
3	-2.27671E-01	4.88034E-01	3.33333E-01	-3.66025E-01	-1.83013E-01
	11	12	13	14	15
1	2.44017E-01	0.00000E+00	-6.10042E-02	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-4.55342E-01	0.00000E+00
3	6.66667E-01	2.44017E-01	-4.55342E-01	-6.10042E-02	1.82137E+00
	16				
1	0.00000E+00				
2	1.82137E+00				
3	-3.33333E-01				

MATRIZ B

	1	2	3	4	5
1	2.27671E-01	0.00000E+00	-9.10684E-01	0.00000E+00	6.83013E-01
2	0.00000E+00	-1.22009E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	-1.22009E-01	2.27671E-01	-6.66667E-01	-9.10684E-01	-1.36603E+00
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.10042E-02	0.00000E+00
2	-1.36603E+00	0.00000E+00	1.82137E+00	0.00000E+00	-4.55342E-01
3	6.83013E-01	1.82137E+00	3.33333E-01	-4.55342E-01	6.10042E-02
	11	12	13	14	15
1	-2.44017E-01	0.00000E+00	1.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	-3.66025E-01	0.00000E+00
3	6.66667E-01	-2.44017E-01	-3.66025E-01	1.83013E-01	4.88034E-01

16
 1 0.00000E+00
 2 4.88034E-01
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	1.83013E-01	0.00000E+00	-2.44017E-01	0.00000E+00	6.10042E-02
2	0.00000E+00	3.66025E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	3.66025E-01	1.83013E-01	-6.66667E-01	-2.44017E-01	4.55342E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	6.83013E-01	0.00000E+00
2	4.55342E-01	0.00000E+00	-1.82137E+00	0.00000E+00	1.36603E+00
3	6.10042E-02	-1.82137E+00	3.33333E-01	1.36603E+00	6.83013E-01
	11	12	13	14	15
1	-9.10684E-01	0.00000E+00	2.27671E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.22009E-01	0.00000E+00
3	6.66667E-01	-9.10684E-01	1.22009E-01	2.27671E-01	-4.88034E-01

16
 1 0.00000E+00
 2 -4.88034E-01
 3 -3.33333E-01

MATRIZ B

	1	2	3	4	5
1	-6.10042E-02	0.00000E+00	2.44017E-01	0.00000E+00	-1.83013E-01
2	0.00000E+00	4.55342E-01	0.00000E+00	-6.66667E-01	0.00000E+00
3	4.55342E-01	-6.10042E-02	-6.66667E-01	2.44017E-01	3.66025E-01
	6	7	8	9	10
1	0.00000E+00	3.33333E-01	0.00000E+00	-2.27671E-01	0.00000E+00
2	3.66025E-01	0.00000E+00	-4.88034E-01	0.00000E+00	1.22009E-01
3	-1.83013E-01	-4.88034E-01	3.33333E-01	1.22009E-01	-2.27671E-01
	11	12	13	14	15
1	9.10684E-01	0.00000E+00	-6.83013E-01	0.00000E+00	-3.33333E-01
2	0.00000E+00	6.66667E-01	0.00000E+00	1.36603E+00	0.00000E+00
3	6.66667E-01	9.10684E-01	1.36603E+00	-6.83013E-01	-1.82137E+00

16
 1 0.00000E+00
 2 -1.82137E+00
 3 -3.33333E-01

MATRIZ DE RIGIDECES DEL ELEMENTO 6

	1	2	3	4	5
1	4.51389E+05	1.77083E+05	-2.22222E+05	-1.25000E+05	1.87500E+05
2	1.77083E+05	7.63889E+05	-4.16667E+04	2.77778E+04	-1.04167E+04
3	-2.22222E+05	-4.16667E+04	7.77778E+05	-4.30795E-04	-2.22222E+05
4	-1.25000E+05	2.77778E+04	1.13171E-03	7.77778E+05	1.25000E+05
5	1.87500E+05	-1.04167E+04	-2.22222E+05	1.25000E+05	4.51389E+05
6	1.04167E+04	2.50000E+05	4.16667E+04	2.77777E+04	-1.77083E+05
7	-2.56945E+05	-4.16667E+04	4.01261E-02	-1.66667E+05	-4.09722E+05
8	-4.16667E+04	-5.69445E+05	-1.66667E+05	4.03817E-02	4.16667E+04
9	2.25695E+05	7.29167E+04	-1.94445E+05	-4.16667E+04	2.18750E+05
10	7.29167E+04	3.81945E+05	-4.16667E+04	-1.94445E+05	1.04167E+04
11	-1.94445E+05	-4.16667E+04	5.55556E+04	3.59710E-03	-1.94445E+05

12	-4.16667E+04	-1.94445E+05	-1.04910E-03	-4.44445E+05	4.16667E+04
13	2.18750E+05	1.04167E+04	-1.94445E+05	4.16667E+04	2.25695E+05
14	-1.04167E+04	4.37500E+05	4.16667E+04	-1.94445E+05	-7.29167E+04
15	-4.09722E+05	-1.25000E+05	4.09587E-02	1.66667E+05	-2.56945E+05
16	-4.16667E+04	-1.09722E+06	1.66667E+05	5.84240E-02	4.16667E+04
	6	7	8	9	10
1	1.04167E+04	-2.56945E+05	-4.16667E+04	2.25695E+05	7.29167E+04
2	2.50000E+05	-4.16667E+04	-5.69445E+05	7.29167E+04	3.81945E+05
3	4.16667E+04	3.23136E-02	-1.66667E+05	-1.94445E+05	-4.16667E+04
4	2.77777E+04	-1.66667E+05	4.03817E-02	-4.16667E+04	-1.94445E+05
5	-1.77083E+05	-4.09722E+05	4.16667E+04	2.18750E+05	1.04167E+04
6	7.63889E+05	1.25000E+05	-1.09722E+06	-1.04167E+04	4.37500E+05
7	1.25000E+05	1.02778E+06	-8.21421E-04	-4.09722E+05	-1.25000E+05
8	-1.09722E+06	-4.33705E-03	2.27778E+06	-4.16667E+04	-1.09722E+06
9	-1.04167E+04	-4.09722E+05	-4.16667E+04	4.51389E+05	1.77083E+05
10	4.37500E+05	-1.25000E+05	-1.09722E+06	1.77083E+05	7.63889E+05
11	4.16667E+04	2.63722E-02	1.66667E+05	-2.22222E+05	-4.16667E+04
12	-1.94445E+05	1.66667E+05	2.16864E-02	-1.25000E+05	2.77777E+04
13	-7.29167E+04	-2.56945E+05	4.16667E+04	1.87500E+05	-1.04167E+04
14	3.81945E+05	4.16667E+04	-5.69445E+05	1.04167E+04	2.50000E+05
15	4.16667E+04	3.05556E+05	-1.04910E-03	-2.56945E+05	-4.16667E+04
16	-5.69445E+05	-2.26228E-03	1.05556E+06	-4.16667E+04	-5.69445E+05
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05
2	-4.16667E+04	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05
3	5.55556E+04	-4.17410E-03	-1.94445E+05	4.16667E+04	3.15837E-02
4	-1.87165E-03	-4.44445E+05	4.16667E+04	-1.94445E+05	1.66667E+05
5	-1.94445E+05	4.16667E+04	2.25695E+05	-7.29167E+04	-2.56945E+05
6	4.16667E+04	-1.94445E+05	-7.29167E+04	3.81945E+05	4.16667E+04
7	4.19972E-02	1.66667E+05	-2.56945E+05	4.16667E+04	3.05556E+05
8	1.66667E+05	2.16864E-02	4.16667E+04	-5.69445E+05	3.63841E-03
9	-2.22222E+05	-1.25000E+05	1.87500E+05	1.04167E+04	-2.56945E+05
10	-4.16667E+04	2.77777E+04	-1.04167E+04	2.50000E+05	-4.16667E+04
11	7.77778E+05	-1.64086E-04	-2.22222E+05	4.16667E+04	2.76993E-02
12	1.39841E-03	7.77778E+05	1.25000E+05	2.77778E+04	-1.66667E+05
13	-2.22222E+05	1.25000E+05	4.51389E+05	-1.77083E+05	-4.09722E+05
14	4.16667E+04	2.77778E+04	-1.77083E+05	7.63889E+05	1.25000E+05
15	4.33243E-02	-1.66667E+05	-4.09722E+05	1.25000E+05	1.02778E+06
16	-1.66667E+05	6.46642E-02	4.16667E+04	-1.09722E+06	7.64841E-03
	16				
1	-4.16667E+04				
2	-1.09722E+06				
3	1.66667E+05				
4	5.84240E-02				
5	4.16667E+04				
6	-5.69445E+05				
7	2.03460E-03				
8	1.05556E+06				
9	-4.16667E+04				
10	-5.69445E+05				
11	-1.66667E+05				
12	6.46642E-02				
13	4.16667E+04				
14	-1.09722E+06				

15 1.39841E-03
16 2.27778E+06

MATRIZ DE RIGIDECES DE LA ESTRUCTURA

	1	2	3	4	5
1	7.77778E+05	2.63984E-02	5.55556E+04	-3.04353E-03	0.00000E+00
2	2.63984E-02	7.77778E+05	-4.17410E-03	-4.44444E+05	0.00000E+00
3	5.55556E+04	5.13406E-04	1.55556E+06	4.48324E-03	5.55556E+04
4	4.37835E-03	-4.44444E+05	2.13949E-03	1.55556E+06	-4.17410E-03
5	0.00000E+00	0.00000E+00	5.55556E+04	5.13406E-04	7.77778E+05
6	0.00000E+00	0.00000E+00	4.37835E-03	-4.44444E+05	-2.42589E-02
7	-2.22222E+05	1.25000E+05	-1.94445E+05	4.16667E+04	0.00000E+00
8	4.16667E+04	2.77778E+04	4.16667E+04	-1.94445E+05	0.00000E+00
9	4.01993E-02	-1.66667E+05	9.70868E-03	1.66667E+05	0.00000E+00
10	-1.66667E+05	-4.08585E-03	1.66667E+05	-4.07597E-03	0.00000E+00
11	-1.94445E+05	-4.16667E+04	-4.44445E+05	0.00000E+00	-1.94445E+05
12	-4.16667E+04	-1.94445E+05	0.00000E+00	5.55556E+04	4.16667E+04
13	0.00000E+00	0.00000E+00	4.01993E-02	-1.66667E+05	4.09587E-02
14	0.00000E+00	0.00000E+00	-1.66667E+05	-4.08585E-03	1.66667E+05
15	0.00000E+00	0.00000E+00	-1.94445E+05	-4.16667E+04	-2.22222E+05
16	0.00000E+00	0.00000E+00	-4.16667E+04	-1.94445E+05	-4.16667E+04
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
22	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
23	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
24	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
25	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
26	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
41	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
42	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
43	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
44	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
45	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
46	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	6	7	8	9	10
1	0.00000E+00	-2.22222E+05	4.16667E+04	2.45743E-02	-1.66667E+05
2	0.00000E+00	1.25000E+05	2.77778E+04	-1.66667E+05	-4.08585E-03

3	-3.04353E-03	-1.94445E+05	4.16667E+04	3.15837E-02	1.66667E+05
4	-4.44444E+05	4.16667E+04	-1.94445E+05	1.66667E+05	-4.07597E-03
5	-2.19152E-02	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	7.77778E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	4.51389E+05	-1.77083E+05	-4.09722E+05	4.16667E+04
8	0.00000E+00	-1.77083E+05	7.63889E+05	1.25000E+05	-1.09722E+06
9	0.00000E+00	-4.09722E+05	1.25000E+05	1.02778E+06	1.39841E-03
10	0.00000E+00	4.16667E+04	-1.09722E+06	-7.97659E-03	2.27778E+06
11	4.16667E+04	2.18750E+05	-1.04167E+04	-4.09722E+05	-4.16667E+04
12	-1.94445E+05	1.04167E+04	4.37500E+05	-1.25000E+05	-1.09722E+06
13	1.66667E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	-4.07597E-03	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	-1.25000E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	2.77778E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
22	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
23	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
24	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
25	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
26	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
41	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
42	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
43	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
44	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
45	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
46	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	11	12	13	14	15
1	-1.94445E+05	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00
2	-4.16667E+04	-1.94445E+05	0.00000E+00	0.00000E+00	0.00000E+00
3	-4.44445E+05	0.00000E+00	2.45743E-02	-1.66667E+05	-1.94445E+05
4	0.00000E+00	5.55555E+04	-1.66667E+05	-4.08585E-03	-4.16667E+04
5	-1.94445E+05	4.16667E+04	3.15837E-02	1.66667E+05	-2.22222E+05
6	4.16667E+04	-1.94445E+05	1.66667E+05	-4.07597E-03	-1.25000E+05
7	2.18750E+05	1.04167E+04	0.00000E+00	0.00000E+00	0.00000E+00
8	-1.04167E+04	4.37500E+05	0.00000E+00	0.00000E+00	0.00000E+00
9	-4.09722E+05	-1.25000E+05	0.00000E+00	0.00000E+00	0.00000E+00

10	-4.16667E+04	-1.09722E+06	0.00000E+00	0.00000E+00	0.00000E+00
11	1.43519E+06	1.77083E+05	-1.05324E+06	0.00000E+00	5.03472E+05
12	1.77083E+05	2.61574E+06	0.00000E+00	-2.75463E+06	4.88281E-03
13	-1.05324E+06	0.00000E+00	2.45370E+06	-4.50113E-03	-1.05324E+06
14	0.00000E+00	-2.75463E+06	1.38582E-02	5.64815E+06	0.00000E+00
15	5.03472E+05	9.76563E-04	-1.05324E+06	0.00000E+00	9.83796E+05
16	0.00000E+00	1.07639E+06	-7.81250E-03	-2.75463E+06	0.00000E+00
17	-1.01852E+05	-4.16667E+04	3.08506E-02	1.66667E+05	-1.75926E+05
18	-1.25000E+05	1.34259E+05	1.66667E+05	6.68200E-02	4.16667E+04
19	-1.75926E+05	-4.16667E+04	2.49725E-02	-1.66667E+05	-1.01852E+05
20	-4.16667E+04	-2.45370E+05	-1.66667E+05	7.50796E-02	1.25000E+05
21	1.94445E+05	-1.04167E+04	-3.56482E+05	4.16667E+04	2.66204E+05
22	1.04167E+04	3.40278E+05	4.16667E+04	-8.42593E+05	-7.29167E+04
23	-3.56482E+05	-4.16667E+04	5.74074E+05	-1.04910E-03	-3.56482E+05
24	-4.16667E+04	-8.42593E+05	2.42523E-03	1.62963E+06	4.16667E+04
25	2.66204E+05	7.29167E+04	-3.56482E+05	-4.16667E+04	1.94445E+05
26	7.29167E+04	5.43982E+05	-4.16667E+04	-8.42593E+05	-1.04167E+04
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
41	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
42	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
43	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
44	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
45	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
46	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	16	17	18	19	20
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	-1.94445E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	2.77778E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	0.00000E+00	-1.01852E+05	-1.25000E+05	-1.75926E+05	-4.16667E+04
12	1.07639E+06	-4.16667E+04	1.34259E+05	-4.16667E+04	-2.45370E+05
13	-7.81250E-03	5.92442E-02	1.66667E+05	4.76287E-02	-1.66667E+05
14	-2.75463E+06	1.66667E+05	6.05700E-02	-1.66667E+05	2.66421E-02
15	0.00000E+00	-1.75926E+05	4.16667E+04	-1.01852E+05	1.25000E+05
16	1.85185E+06	4.16667E+04	-2.45370E+05	4.16667E+04	1.34259E+05

17	4.16667E+04	7.03704E+05	1.15547E-02	-1.48148E+05	2.76788E-04
18	-2.45370E+05	3.74216E-03	9.81482E+05	5.98216E-03	-7.59259E+05
19	4.16667E+04	-1.48148E+05	-1.83035E-03	7.03704E+05	-9.80580E-03
20	1.34259E+05	-6.16853E-03	-7.59259E+05	-3.55580E-03	9.81482E+05
21	-7.29167E+04	-1.01852E+05	1.25000E+05	-1.75926E+05	4.16667E+04
22	5.43982E+05	4.16667E+04	1.34259E+05	4.16667E+04	-2.45370E+05
23	4.16667E+04	4.15652E-02	-1.66667E+05	2.39985E-02	1.66667E+05
24	-8.42593E+05	-1.66667E+05	8.26277E-02	1.66667E+05	5.35977E-02
25	1.04167E+04	-1.75926E+05	-4.16667E+04	-1.01852E+05	-1.25000E+05
26	3.40278E+05	-4.16667E+04	-2.45370E+05	-4.16667E+04	1.34259E+05
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
36	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
41	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
42	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
43	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
44	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
45	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
46	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	21	22	23	24	25
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	1.94445E+05	1.04167E+04	-3.56482E+05	-4.16667E+04	2.66204E+05
12	-1.04167E+04	3.40278E+05	-4.16667E+04	-8.42593E+05	7.29167E+04
13	-3.56482E+05	4.16667E+04	5.74074E+05	2.42523E-03	-3.56482E+05
14	4.16667E+04	-8.42593E+05	-1.04910E-03	1.62963E+06	-4.16667E+04
15	2.66204E+05	-7.29167E+04	-3.56482E+05	4.16667E+04	1.94445E+05
16	-7.29167E+04	5.43982E+05	4.16667E+04	-8.42593E+05	1.04167E+04
17	-1.01852E+05	4.16667E+04	1.03152E-02	-1.66667E+05	-1.75926E+05
18	1.25000E+05	1.34259E+05	-1.66667E+05	8.26277E-02	-4.16667E+04
19	-1.75926E+05	4.16667E+04	9.93597E-03	1.66667E+05	-1.01852E+05
20	4.16667E+04	-2.45370E+05	1.66667E+05	5.35977E-02	-1.25000E+05
21	1.06482E+06	0.00000E+00	-1.28704E+06	3.90625E-03	5.69445E+05
22	0.00000E+00	2.17593E+06	7.81250E-03	-3.31482E+06	-1.95313E-03
23	-1.28704E+06	7.81250E-03	2.85185E+06	-9.18863E-03	-1.28704E+06

24	3.90625E-03	-3.31482E+06	2.92074E-03	6.74074E+06	0.00000E+00
25	5.69445E+05	-1.95313E-03	-1.28704E+06	0.00000E+00	1.06482E+06
26	-1.95313E-03	1.27778E+06	-7.81250E-03	-3.31482E+06	1.56250E-02
27	-1.01852E+05	-4.16667E+04	3.63438E-02	1.66667E+05	-1.75926E+05
28	-1.25000E+05	1.34259E+05	1.66667E+05	8.08825E-02	4.16667E+04
29	-1.75926E+05	-4.16667E+04	2.49725E-02	-1.66667E+05	-1.01852E+05
30	-4.16667E+04	-2.45370E+05	-1.66667E+05	7.50796E-02	1.25000E+05
31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	1.94445E+05	-1.04167E+04	-3.56482E+05	4.16667E+04	2.66204E+05
36	1.04167E+04	3.40278E+05	4.16667E+04	-8.42593E+05	-7.29167E+04
37	-3.56482E+05	-4.16667E+04	5.74074E+05	-1.04910E-03	-3.56482E+05
38	-4.16667E+04	-8.42593E+05	2.42523E-03	1.62963E+06	4.16667E+04
39	2.66204E+05	7.29167E+04	-3.56482E+05	-4.16667E+04	1.94445E+05
40	7.29167E+04	5.43982E+05	-4.16667E+04	-8.42593E+05	-1.04167E+04
41	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
42	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
43	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
44	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
45	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
46	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	26	27	28	29	30
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	7.29167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
12	5.43982E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
13	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	-8.42593E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	-1.04167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	3.40278E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	-2.45370E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	-4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	1.34259E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	-1.95313E-03	-1.01852E+05	-1.25000E+05	-1.75926E+05	-4.16667E+04
22	1.27778E+06	-4.16667E+04	1.34259E+05	-4.16667E+04	-2.45370E+05
23	-7.81250E-03	6.60557E-02	1.66667E+05	4.76287E-02	-1.66667E+05
24	-3.31482E+06	1.66667E+05	7.46325E-02	-1.66667E+05	2.66421E-02
25	1.56250E-02	-1.75926E+05	4.16667E+04	-1.01852E+05	1.25000E+05
26	2.17593E+06	4.16667E+04	-2.45370E+05	4.16667E+04	1.34259E+05
27	4.16667E+04	7.03704E+05	-2.50784E-03	-1.48148E+05	-1.48103E-03
28	-2.45370E+05	-3.28909E+03	9.81482E+05	-1.04910E-03	-7.59259E+05
29	4.16667E+04	-1.48148E+05	-1.83035E-03	7.03704E+05	-1.99330E-03
30	1.34259E+05	-1.13837E-04	-7.59259E+05	-3.55580E-03	9.81482E+05

31	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	-7.29167E+04	-1.01852E+05	1.25000E+05	-1.75926E+05	4.16667E+04
36	5.43982E+05	4.16667E+04	1.34259E+05	4.16667E+04	-2.45370E+05
37	4.16667E+04	6.96902E-02	-1.66667E+05	5.52485E-02	1.66667E+05
38	-8.42593E+05	-1.66667E+05	8.26277E-02	1.66667E+05	5.35977E-02
39	1.04167E+04	-1.75926E+05	-4.16667E+04	-1.01852E+05	-1.25000E+05
40	3.40278E+05	-4.16667E+04	-2.45370E+05	-4.16667E+04	1.34259E+05
41	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
42	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
43	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
44	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
45	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
46	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
	31	32	33	34	35
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
12	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
13	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1.94445E+05
22	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	-1.04167E+04
23	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	-3.56482E+05
24	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	4.16667E+04
25	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	2.66204E+05
26	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	-7.29167E+04
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	-1.01852E+05
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	1.25000E+05
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	-1.75926E+05
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	4.16667E+04
31	4.51389E+05	1.77083E+05	-4.09722E+05	-4.16667E+04	2.18750E+05
32	1.77083E+05	7.63889E+05	-1.25000E+05	-1.09722E+06	1.04167E+04
33	-4.09722E+05	-1.25000E+05	1.02778E+06	1.48036E-02	-4.09722E+05
34	-4.16667E+04	-1.09722E+06	1.12880E-02	2.27778E+06	4.16667E+04
35	2.18750E+05	1.04167E+04	-4.09722E+05	4.16667E+04	1.43519E+06
36	-1.04167E+04	4.37500E+05	1.25000E+05	-1.09722E+06	-1.77083E+05
37	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	-1.05324E+06

38	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
39	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	5.03472E+05
40	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	9.76563E-04
41	-2.22222E+05	-4.16667E+04	2.63722E-02	1.66667E+05	-1.94445E+05
42	-1.25000E+05	2.77777E+04	1.66667E+05	2.16864E-02	4.16667E+04
43	-1.94445E+05	-4.16667E+04	3.23136E-02	-1.66667E+05	-4.44445E+05
44	-4.16667E+04	-1.94445E+05	-1.66667E+05	4.03817E-02	-7.81250E-03
45	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	-1.94445E+05
46	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	-4.16667E+04
	36	37	38	39	40
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
12	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
13	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	1.04167E+04	-3.56482E+05	-4.16667E+04	2.66204E+05	7.29167E+04
22	3.40278E+05	-4.16667E+04	-8.42593E+05	7.29167E+04	5.43982E+05
23	4.16667E+04	5.74074E+05	2.42523E-03	-3.56482E+05	-4.16667E+04
24	-8.42593E+05	-1.04910E-03	1.62963E+06	-4.16667E+04	-8.42593E+05
25	-7.29167E+04	-3.56482E+05	4.16667E+04	1.94445E+05	-1.04167E+04
26	5.43982E+05	4.16667E+04	-8.42593E+05	1.04167E+04	3.40278E+05
27	4.16667E+04	3.84402E-02	-1.66667E+05	-1.75926E+05	-4.16667E+04
28	1.34259E+05	-1.66667E+05	1.38878E-01	-4.16667E+04	-2.45370E+05
29	4.16667E+04	4.11860E-02	1.66667E+05	-1.01852E+05	-4.16667E+04
30	-2.45370E+05	1.66667E+05	1.16098E-01	-1.25000E+05	1.34259E+05
31	-1.04167E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
32	4.37500E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
33	1.25000E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
34	-1.09722E+06	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
35	-1.77083E+05	-1.05324E+06	7.81250E-03	5.03472E+05	9.76563E-04
36	2.61574E+06	7.81250E-03	-2.75463E+06	-3.90625E-03	1.07639E+06
37	7.81250E-03	2.45370E+06	-4.11051E-03	-1.05324E+06	-7.81250E-03
38	-2.75463E+06	-7.62613E-03	5.64815E+06	0.00000E+00	-2.75463E+06
39	-1.95313E-03	-1.05324E+06	-3.90625E-03	9.83796E+05	3.12500E-02
40	1.07639E+06	0.00000E+00	-2.75463E+06	3.12500E-02	1.85185E+06
41	4.16667E+04	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
42	-1.94445E+05	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
43	0.00000E+00	2.63722E-02	1.66667E+05	-1.94445E+05	4.16667E+04
44	5.55555E+04	1.66667E+05	2.16864E-02	4.16667E+04	-1.94445E+05

45	-4.16667E+04	3.23136E-02	-1.66667E+05	-2.22222E+05	4.16667E+04
46	-1.94445E+05	-1.66667E+05	4.03817E-02	1.25000E+05	2.77777E+04
	41	42	43	44	45
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
3	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
6	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
7	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
8	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
9	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
10	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
11	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
12	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
13	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
14	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
15	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
16	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
17	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
18	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
19	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
20	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
21	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
22	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
23	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
24	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
25	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
26	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
27	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
28	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
29	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
30	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
31	-2.22222E+05	-1.25000E+05	-1.94445E+05	-4.16667E+04	0.00000E+00
32	-4.16667E+04	2.77777E+04	-4.16667E+04	-1.94445E+05	0.00000E+00
33	4.19972E-02	1.66667E+05	2.45011E-02	-1.66667E+05	0.00000E+00
34	1.66667E+05	2.16864E-02	-1.66667E+05	4.03817E-02	0.00000E+00
35	-1.94445E+05	4.16667E+04	-4.44445E+05	7.81250E-03	-1.94445E+05
36	4.16667E+04	-1.94445E+05	0.00000E+00	5.55555E+04	-4.16667E+04
37	0.00000E+00	0.00000E+00	4.19972E-02	1.66667E+05	4.01261E-02
38	0.00000E+00	0.00000E+00	1.66667E+05	2.16864E-02	-1.66667E+05
39	0.00000E+00	0.00000E+00	-1.94445E+05	4.16667E+04	-2.22222E+05
40	0.00000E+00	0.00000E+00	4.16667E+04	-1.94445E+05	4.16667E+04
41	7.77778E+05	-1.64086E-04	5.55555E+04	3.59710E-03	0.00000E+00
42	1.39841E-03	7.77778E+05	-1.04910E-03	-4.44445E+05	0.00000E+00
43	5.55556E+04	3.63841E-03	1.55556E+06	-5.94881E-04	5.55556E+04
44	-1.87165E-03	-4.44445E+05	2.53012E-03	1.55556E+06	-1.04910E-03
45	0.00000E+00	0.00000E+00	5.55556E+04	-4.17410E-03	7.77778E+05
46	0.00000E+00	0.00000E+00	-1.87165E-03	-4.44445E+05	1.13171E-03
	46				
1	0.00000E+00				
2	0.00000E+00				
3	0.00000E+00				
4	0.00000E+00				

5 0.00000E+00
6 0.00000E+00
7 0.00000E+00
8 0.00000E+00
9 0.00000E+00
10 0.00000E+00
11 0.00000E+00
12 0.00000E+00
13 0.00000E+00
14 0.00000E+00
15 0.00000E+00
16 0.00000E+00
17 0.00000E+00
18 0.00000E+00
19 0.00000E+00
20 0.00000E+00
21 0.00000E+00
22 0.00000E+00
23 0.00000E+00
24 0.00000E+00
25 0.00000E+00
26 0.00000E+00
27 0.00000E+00
28 0.00000E+00
29 0.00000E+00
30 0.00000E+00
31 0.00000E+00
32 0.00000E+00
33 0.00000E+00
34 0.00000E+00
35 -4.16667E+04
36 -1.94445E+05
37 -1.66667E+05
38 4.03817E-02
39 1.25000E+05
40 2.77777E+04
41 0.00000E+00
42 0.00000E+00
43 3.59710E-03
44 -4.44445E+05
45 -4.30795E-04
46 7.77778E+05

VECTOR DE CARGA

1
1 0.000000E+00
2 0.000000E+00
3 0.000000E+00
4 0.000000E+00
5 0.000000E+00
6 0.000000E+00
7 0.000000E+00
8 0.000000E+00

9 0.000000E+00
10 0.000000E+00
11 0.000000E+00
12 -1.500000E+01
13 0.000000E+00
14 0.000000E+00
15 0.000000E+00
16 0.000000E+00
17 0.000000E+00
18 0.000000E+00
19 0.000000E+00
20 0.000000E+00
21 0.000000E+00
22 -1.500000E+01
23 0.000000E+00
24 0.000000E+00
25 0.000000E+00
26 0.000000E+00
27 0.000000E+00
28 0.000000E+00
29 0.000000E+00
30 0.000000E+00
31 0.000000E+00
32 0.000000E+00
33 0.000000E+00
34 0.000000E+00
35 0.000000E+00
36 -1.500000E+01
37 0.000000E+00
38 0.000000E+00
39 0.000000E+00
40 0.000000E+00
41 0.000000E+00
42 0.000000E+00
43 0.000000E+00
44 0.000000E+00
45 0.000000E+00
46 0.000000E+00

DESPLAZAMIENTOS

1
1 -1.026306E-04
2 -1.139396E-04
3 -1.182269E-05
4 -9.275171E-05
5 9.092480E-05
6 -9.875158E-05
7 -1.262546E-04
8 -2.327569E-04
9 -7.630730E-05
10 -2.540072E-04
11 -4.090899E-05
12 -2.839967E-04
13 4.471329E-05

14 -2.715796E-04
 15 1.439969E-04
 16 -2.658903E-04
 17 -1.027260E-04
 18 -6.882166E-04
 19 1.569230E-04
 20 -6.963440E-04
 21 2.396702E-10
 22 -9.888333E-04
 23 2.361354E-10
 24 -9.762172E-04
 25 2.266010E-10
 26 -9.530414E-04
 27 1.027265E-04
 28 -6.882164E-04
 29 -1.569228E-04
 30 -6.963437E-04
 31 1.262543E-04
 32 -2.327562E-04
 33 7.630728E-05
 34 -2.540066E-04
 35 4.090924E-05
 36 -2.839960E-04
 37 -4.471300E-05
 38 -2.715790E-04
 39 -1.439965E-04
 40 -2.658898E-04
 41 1.026305E-04
 42 -1.139394E-04
 43 1.182277E-05
 44 -9.275149E-05
 45 -9.092452E-05
 46 -9.875128E-05

NUMERO DE ELEMENTO 1

PUNTO GAUSSIANO X 8.422649

PUNTO GAUSSIANO Y 1.211325

ESFUERZOS

1

1 5.979550E+01

2 4.628879E+01

3 7.868370E+01

PUNTO GAUSSIANO X 9.577350

PUNTO GAUSSIANO Y 1.211325

ESFUERZOS

1

1 5.245256E+01

2 -2.599832E+01

3 3.592415E+01

PUNTO GAUSSIANO X 9.577350

PUNTO GAUSSIANO Y 1.788675

ESFUERZOS

1

1 1.688573E+02

2 -1.124755E+01
 3 4.368509E+01
 PUNTO GAUSSIANO X 8.422649
 PUNTO GAUSSIANO Y 1.788675
 ESFUERZOS
 1
 1 3.763638E+01
 2 1.002304E+01
 3 1.827028E+01
 NUMERO DE ELEMENTO 2
 PUNTO GAUSSIANO X 4.226497E-01
 PUNTO GAUSSIANO Y 1.211325
 ESFUERZOS
 1
 1 5.245259E+01
 2 -2.599843E+01
 3 -3.592405E+01
 PUNTO GAUSSIANO X 1.577350
 PUNTO GAUSSIANO Y 1.211325
 ESFUERZOS
 1
 1 5.979570E+01
 2 4.628868E+01
 3 -7.868365E+01
 PUNTO GAUSSIANO X 1.577350
 PUNTO GAUSSIANO Y 1.788675
 ESFUERZOS
 1
 1 3.763624E+01
 2 1.002313E+01
 3 -1.827026E+01
 PUNTO GAUSSIANO X 4.226497E-01
 PUNTO GAUSSIANO Y 1.788675
 ESFUERZOS
 1
 1 1.688571E+02
 2 -1.124753E+01
 3 -4.368501E+01
 NUMERO DE ELEMENTO 3
 PUNTO GAUSSIANO X 8.422649
 PUNTO GAUSSIANO Y 2.113249E-01
 ESFUERZOS
 1
 1 -6.346922E+01
 2 -8.929427E+00
 3 9.798850E+00
 PUNTO GAUSSIANO X 9.577350
 PUNTO GAUSSIANO Y 2.113249E-01
 ESFUERZOS
 1
 1 -1.415751E+02
 2 -1.261338E+01
 3 3.442387E+01
 PUNTO GAUSSIANO X 9.577350

PUNTO GAUSSIANO Y 7.886751E-01
ESFUERZOS

1

1 -1.744156E+01
2 5.646231E+00
3 3.597277E+01

PUNTO GAUSSIANO X 8.422649

PUNTO GAUSSIANO Y 7.886751E-01
ESFUERZOS

1

1 2.833050E+01
2 -1.507701E+01
3 4.325267E+01

NUMERO DE ELEMENTO 4

PUNTO GAUSSIANO X 5.633975

PUNTO GAUSSIANO Y 2.113249E-01
ESFUERZOS

1

1 1.422972E+02
2 -4.830717E+00
3 1.430962E+01

PUNTO GAUSSIANO X 7.366025

PUNTO GAUSSIANO Y 2.113249E-01
ESFUERZOS

1

1 -7.711845E+00
2 8.639527E+00
3 1.971146E+01

PUNTO GAUSSIANO X 7.366025

PUNTO GAUSSIANO Y 7.886751E-01
ESFUERZOS

1

1 7.000503E+01
2 9.981368E+00
3 3.029179E+01

PUNTO GAUSSIANO X 5.633975

PUNTO GAUSSIANO Y 7.886751E-01
ESFUERZOS

1

1 -8.000408E+01
2 -1.739123E+01
3 3.569318E+01

NUMERO DE ELEMENTO 5

PUNTO GAUSSIANO X 2.633975

PUNTO GAUSSIANO Y 2.113249E-01
ESFUERZOS

1

1 -7.711892E+00
2 8.639504E+00
3 -1.971152E+01

PUNTO GAUSSIANO X 4.366025

PUNTO GAUSSIANO Y 2.113249E-01
ESFUERZOS

1
 1 1.422975E+02
 2 -4.830732E+00
 3 -1.430960E+01
 PUNTO GAUSSIANO X 4.366025
 PUNTO GAUSSIANO Y 7.886751E-01
 ESFUERZOS

1
 1 -8.000406E+01
 2 -1.739138E+01
 3 -3.569316E+01
 PUNTO GAUSSIANO X 2.633975
 PUNTO GAUSSIANO Y 7.886751E-01
 ESFUERZOS

1
 1 7.000526E+01
 2 9.981536E+00
 3 -3.029188E+01
 NUMERO DE ELEMENTO 6
 PUNTO GAUSSIANO X 4.226497E-01
 PUNTO GAUSSIANO Y 2.113249E-01
 ESFUERZOS

1
 1 -1.415746E+02
 2 -1.261344E+01
 3 -3.442372E+01
 PUNTO GAUSSIANO X 1.577350
 PUNTO GAUSSIANO Y 2.113249E-01
 ESFUERZOS

1
 1 -6.346903E+01
 2 -8.929205E+00
 3 -9.798805E+00
 PUNTO GAUSSIANO X 1.577350
 PUNTO GAUSSIANO Y 7.886751E-01
 ESFUERZOS

1
 1 2.833077E+01
 2 -1.507709E+01
 3 -4.325247E+01
 PUNTO GAUSSIANO X 4.226497E-01
 PUNTO GAUSSIANO Y 7.886751E-01
 ESFUERZOS

1
 1 -1.744135E+01
 2 5.646152E+00
 3 -3.597266E+01

ARCHIVO B.8. DATOS PARA EL PROCESADOR (INDICADORES DE ECUACION Y MULTIPLICACION DE MATRICES DE CADA ELEMENTO):

A:\AUXILIA2

GENERADO POR EL PROCESADOR.

	11	12	3	4
	0	0	0	0
	0	0	1	2
	7	8	9	10
-142943.000000	-569177.300000	1897258.000000	-277777.800000	
-474314.400000	-50836.860000	694444.500000	203347.500000	
-381276.500000	-152510.600000	508368.600000	277777.800000	
-127092.200000	-189725.800000	-694444.500000	758903.000000	
-284588.600000	-2845886.000000	379451.500000	-1388889.000000	
-94862.880000	-254184.300000	138888.900000	1016737.000000	
-76255.300000	-762553.000000	101673.700000	1388889.000000	
-25418.430000	-948628.800000	-138888.900000	3794515.000000	
-1138355.000000	-569177.300000	-555555.600000	758903.000000	
-101673.700000	-189725.800000	406694.900000	277777.800000	
-305021.200000	-152510.600000	555555.600000	203347.500000	
-379451.500000	-50836.870000	1517806.000000	-277777.800000	
8.422649	1.211325	474314.400000	-50836.880000	
-1897258.000000	-277777.800000	1422943.000000	-569177.300000	
694444.500000	758903.000000	127092.200000	-189725.800000	
-508368.700000	277777.800000	381276.500000	-152510.600000	
-694444.600000	203347.500000	94862.880000	-254184.400000	
-379451.500000	-1388889.000000	284588.700000	-2845887.000000	
138888.900000	3794515.000000	25418.440000	-948628.800000	
-101673.700000	1388889.000000	76255.300000	-762553.100000	
-138888.900000	1016737.000000	-101673.800000	189725.800000	
-555555.500000	-758903.100000	-1138355.000000	569177.300000	
1517806.000000	-277777.800000	-379451.500000	50836.870000	
555555.600000	-203347.500000	-305021.200000	152510.600000	
406694.900000	-277777.800000	9.577350	1.211325	
381276.500000	152510.600000	-508368.700000	-277777.800000	
127092.200000	189725.800000	694444.500000	-758903.000000	
1422943.000000	569177.300000	-1897258.000000	277777.800000	
474314.400000	50836.880000	-694444.600000	-203347.500000	
76255.300000	762553.100000	-101673.700000	-1388889.000000	
25418.440000	948628.800000	138888.900000	-3794515.000000	
284588.700000	2845887.000000	-379451.500000	1388889.000000	
94862.880000	254184.400000	-138888.900000	-1016737.000000	
305021.200000	152510.600000	-555555.600000	-203347.500000	
379451.500000	50836.870000	-1517806.000000	277777.800000	
1138355.000000	569177.300000	555555.500000	-758903.100000	
101673.800000	189725.800000	-406694.900000	-277777.800000	
9.577350	1.788675	-127092.200000	189725.800000	
508368.700000	-277777.800000	-381276.500000	152510.600000	
694444.500000	-203347.500000	-474314.400000	50836.860000	
1897258.000000	277777.800000	-1422943.000000	569177.300000	
-694444.400000	-758903.000000	-25418.440000	948628.800000	
101673.700000	-1388889.000000	-76255.300000	762553.000000	
138888.900000	-1016737.000000	-94862.880000	254184.300000	
379451.500000	1388889.000000	-284588.600000	2845886.000000	
-138888.900000	-3794515.000000	379451.500000	-50836.870000	

-555555.600000	203347.500000	305021.200000	-152510.600000
-406694.900000	277777.800000	101673.700000	-189725.800000
555555.600000	758903.000000	1138355.000000	-569177.300000
-1517806.000000	-277777.800000	8.422649	1.788675
0	0	43	44
35	36	33	34
31	32	41	42
0	0	0	0
-1422943.000000	-569177.300000	1897258.000000	-277777.800000
-474314.400000	-50836.870000	694444.500000	203347.500000
-381276.500000	-152510.600000	508368.600000	277777.800000
-127092.200000	-189725.800000	-694444.500000	758903.000000
-284588.600000	-2845886.000000	379451.500000	-1388889.000000
-94862.880000	-254184.300000	138888.900000	1016737.000000
-76255.300000	-762553.000000	101673.700000	1388889.000000
-25418.430000	-948628.800000	-138888.900000	3794515.000000
-1138355.000000	-569177.300000	-555555.600000	758903.000000
-101673.700000	-189725.800000	406694.900000	277777.800000
-305021.200000	-152510.600000	555555.600000	203347.500000
-379451.500000	-50836.870000	1517806.000000	-277777.800000
4.226497E-01	1.211325	474314.400000	-50836.870000
-1897258.000000	-277777.800000	1422943.000000	-569177.300000
694444.500000	758903.000000	127092.200000	-189725.800000
-508368.600000	277777.800000	381276.500000	-152510.600000
-694444.500000	203347.500000	94862.880000	-254184.300000
-379451.500000	-1388889.000000	284588.600000	-2845886.000000
138888.900000	3794515.000000	25418.430000	-948628.800000
-101673.700000	1388889.000000	76255.300000	-762553.000000
-138888.900000	1016737.000000	-101673.700000	189725.800000
-555555.600000	-758903.000000	-1138355.000000	569177.300000
1517806.000000	277777.800000	-379451.500000	50836.870000
555555.600000	-203347.500000	-305021.200000	152510.600000
406694.900000	-277777.800000	1.577350	1.211325
381276.500000	152510.600000	-508368.700000	-277777.800000
127092.200000	189725.800000	694444.400000	-758903.000000
1422943.000000	569177.300000	-1897258.000000	277777.800000
474314.400000	50836.870000	-694444.500000	-203347.500000
76255.300000	762553.000000	-101673.700000	-1388889.000000
25418.440000	948628.800000	138888.800000	-3794515.000000
284588.600000	2845886.000000	-379451.500000	1388889.000000
94862.880000	254184.300000	-138888.900000	-1016737.000000
305021.200000	152510.600000	-555555.600000	-203347.500000
379451.500000	50836.870000	-1517806.000000	277777.800000
1138355.000000	569177.300000	555555.600000	-758903.000000
101673.700000	189725.800000	-406694.900000	-277777.800000
1.577350	1.788675	-127092.200000	189725.800000
508368.700000	-277777.800000	-381276.500000	152510.600000
694444.500000	-203347.500000	-474314.400000	50836.870000
1897258.000000	277777.800000	-1422943.000000	569177.300000
-694444.400000	-758903.000000	-25418.440000	948628.800000
101673.700000	-1388889.000000	-76255.300000	762553.000000
138888.900000	-1016737.000000	-94862.880000	254184.300000
379451.500000	1388889.000000	-284588.600000	2845886.000000
-138888.900000	-3794515.000000	379451.500000	-50836.870000

-555555.600000	203347.500000	305021.200000	-152510.600000
-406694.900000	277777.800000	101673.700000	-189725.800000
555555.600000	758903.000000	1138355.000000	-569177.300000
-1517806.000000	-277777.800000	4.226497E-01	1.788675
15	16	5	6
0	0	0	0
0	0	3	4
11	12	13	14
-1422943.000000	-569177.300000	1897258.000000	-277777.800000
-474314.400000	-50836.860000	694444.500000	203347.500000
-381276.500000	-152510.600000	508368.700000	277777.800000
-127092.200000	-189725.800000	-694444.400000	758903.000000
-284588.600000	-2845886.000000	379451.500000	-1388889.000000
-94862.880000	-254184.300000	138888.900000	1016737.000000
-76255.300000	-762553.000000	101673.700000	1388889.000000
-25418.430000	-948628.800000	-138888.900000	3794515.000000
-1138355.000000	-569177.300000	-555555.600000	758903.000000
-101673.700000	-189725.800000	406694.900000	277777.800000
-305021.200000	-152510.600000	555555.600000	203347.500000
-379451.500000	-50836.870000	1517806.000000	-277777.800000
8.422649	2.113249E-01	474314.400000	-50836.880000
-1897258.000000	-277777.800000	1422943.000000	-569177.300000
694444.500000	758903.000000	127092.200000	-189725.800000
-508368.700000	277777.800000	381276.500000	-152510.600000
-694444.600000	203347.500000	94862.880000	-254184.400000
-379451.500000	-1388889.000000	284588.700000	-2845887.000000
138888.900000	3794515.000000	25418.440000	-948628.800000
-101673.700000	1388889.000000	76255.300000	-762553.100000
-138888.900000	1016737.000000	-101673.800000	189725.800000
-555555.500000	-758903.100000	-1138355.000000	569177.300000
1517806.000000	277777.800000	-379451.500000	50836.870000
555555.600000	-203347.500000	-305021.200000	152510.600000
406694.900000	-277777.800000	9.577350	2.113249E-01
381276.500000	152510.600000	-508368.700000	-277777.800000
127092.200000	189725.800000	694444.500000	-758903.000000
1422943.000000	569177.300000	-1897258.000000	277777.800000
474314.400000	50836.880000	-694444.600000	-203347.500000
76255.300000	762553.100000	-101673.700000	-1388889.000000
25418.440000	948628.800000	138888.900000	-3794515.000000
284588.700000	2845887.000000	-379451.500000	1388889.000000
94862.880000	254184.400000	-138888.900000	-1016737.000000
305021.200000	152510.600000	-555555.600000	-203347.500000
379451.500000	50836.870000	-1517806.000000	277777.800000
1138355.000000	569177.300000	555555.500000	-758903.100000
101673.800000	189725.800000	-406694.900000	-277777.800000
9.577350	7.886751E-01	-127092.200000	189725.800000
508368.700000	-277777.800000	-381276.500000	152510.600000
694444.500000	-203347.500000	-474314.400000	50836.860000
1897258.000000	277777.800000	-1422943.000000	569177.300000
-694444.400000	-758903.000000	-25418.440000	948628.800000
101673.700000	-1388889.000000	-76255.300000	762553.000000
138888.900000	-1016737.000000	-94862.880000	254184.300000
379451.500000	1388889.000000	-284588.600000	2845886.000000
-138888.900000	-3794515.000000	379451.500000	-50836.870000

-555555.600000	203347.500000	305021.200000	-152510.600000
-406694.900000	277777.800000	101673.700000	-189725.800000
555555.600000	758903.000000	1138355.000000	-569177.300000
-1517806.000000	-277777.800000	8.422649	7.886751E-01
25	26	19	20
15	16	13	14
11	12	17	18
21	22	23	24
-948628.800000	-569177.300000	1264838.000000	-277777.800000
-316209.600000	-50836.870000	462963.000000	203347.500000
-254184.300000	-152510.600000	338912.400000	277777.800000
-84728.110000	-189725.800000	-462962.900000	758903.000000
-189725.800000	-2845886.000000	252967.700000	-1388889.000000
-63241.910000	-254184.300000	92592.600000	1016737.000000
-50836.870000	-762553.000000	67782.480000	1388889.000000
-16945.620000	-948628.800000	-92592.590000	3794515.000000
-1138355.000000	-379451.500000	-555555.600000	505935.300000
-101673.700000	-126483.800000	406694.900000	185185.200000
-305021.200000	-101673.700000	555555.600000	135565.000000
-379451.500000	-33891.250000	1517806.000000	-185185.200000
5.633975	2.113249E-01	316209.600000	-50836.870000
-1264838.000000	-277777.800000	948628.800000	-569177.300000
462962.900000	758903.000000	84728.110000	-189725.800000
-338912.400000	277777.800000	254184.300000	-152510.600000
-462963.000000	203347.500000	63241.910000	-254184.400000
-252967.700000	-1388889.000000	189725.800000	-2845886.000000
92592.590000	3794515.000000	16945.620000	-948628.800000
-67782.480000	1388889.000000	50836.870000	-762553.100000
-92592.600000	1016737.000000	-101673.700000	126483.800000
-555555.600000	-505935.300000	-1138355.000000	379451.500000
1517806.000000	185185.200000	-379451.500000	33891.250000
555555.600000	-135565.000000	-305021.200000	101673.700000
406694.900000	-185185.200000	7.366025	2.113249E-01
254184.300000	152510.600000	-338912.400000	-277777.800000
84728.110000	189725.800000	462962.900000	-758903.000000
948628.800000	569177.300000	-1264838.000000	277777.800000
316209.600000	50836.870000	-462963.000000	-203347.500000
50836.870000	762553.100000	-67782.480000	-1388889.000000
16945.620000	948628.800000	92592.590000	-3794515.000000
189725.800000	2845886.000000	-252967.700000	1388889.000000
63241.910000	254184.400000	-92592.600000	-1016737.000000
305021.200000	101673.700000	-555555.600000	-135565.000000
379451.500000	33891.250000	-1517806.000000	185185.200000
1138355.000000	379451.500000	555555.600000	-505935.300000
101673.700000	126483.800000	-406694.900000	-185185.200000
7.366025	7.886751E-01	-84728.110000	189725.800000
338912.400000	-277777.800000	-254184.300000	152510.600000
462963.000000	-203347.500000	-316209.600000	50836.870000
1264838.000000	277777.800000	-948628.800000	569177.300000
-462962.900000	-758903.000000	-16945.620000	948628.800000
67782.480000	-1388889.000000	-50836.870000	762553.000000
92592.600000	-1016737.000000	-63241.910000	254184.300000
252967.700000	1388889.000000	-189725.800000	2845886.000000
-92592.590000	-3794515.000000	379451.500000	-33891.250000

-555555.600000	135565.000000	305021.200000	-101673.700000
-406694.900000	185185.200000	101673.700000	-126483.800000
555555.600000	505935.300000	1138355.000000	-379451.500000
-1517806.000000	-185185.200000	5.633975	7.886751E-01
39	40	29	30
25	26	23	24
21	22	27	28
35	36	37	38
-948628.800000	-569177.300000	1264838.000000	-277777.800000
-316209.600000	-50836.870000	462963.000000	203347.500000
-254184.300000	-152510.600000	338912.400000	277777.800000
-84728.110000	-189725.800000	-462962.900000	758903.000000
-189725.800000	-2845886.000000	252967.700000	-1388889.000000
-63241.910000	-254184.300000	92592.600000	1016737.000000
-50836.870000	-762553.000000	67782.480000	1388889.000000
-16945.620000	-948628.800000	-92592.590000	3794515.000000
-1138355.000000	-379451.500000	-555555.600000	505935.300000
-101673.700000	-126483.800000	406694.900000	185185.200000
-305021.200000	-101673.700000	555555.600000	135565.000000
-379451.500000	-33891.250000	1517806.000000	-185185.200000
2.633975	2.113249E-01	316209.600000	-50836.870000
-1264838.000000	-277777.800000	948628.800000	-569177.300000
462962.900000	758903.000000	84728.110000	-189725.800000
-338912.400000	277777.800000	254184.300000	-152510.600000
-462963.000000	203347.500000	63241.910000	-254184.300000
-252967.700000	-1388889.000000	189725.800000	-2845886.000000
92592.590000	3794515.000000	16945.620000	-948628.800000
-67782.480000	1388889.000000	50836.870000	-762553.000000
-92592.600000	1016737.000000	-101673.700000	126483.800000
-555555.600000	-505935.300000	-1138355.000000	379451.500000
1517806.000000	185185.200000	-379451.500000	33891.250000
555555.600000	-135565.000000	-305021.200000	101673.700000
406694.900000	-185185.200000	4.366025	2.113249E-01
254184.300000	152510.600000	-338912.400000	-277777.800000
84728.110000	189725.800000	462962.900000	-758903.000000
948628.800000	569177.300000	-1264838.000000	277777.800000
316209.600000	50836.870000	-462963.000000	-203347.500000
50836.870000	762553.000000	-67782.480000	-1388889.000000
16945.620000	948628.800000	92592.590000	-3794515.000000
189725.800000	2845886.000000	-252967.700000	1388889.000000
63241.910000	254184.300000	-92592.600000	-1016737.000000
305021.200000	101673.700000	-555555.600000	-135565.000000
379451.500000	33891.250000	-1517806.000000	185185.200000
1138355.000000	379451.500000	555555.600000	-505935.300000
101673.700000	126483.800000	-406694.900000	-185185.200000
4.366025	7.886751E-01	-84728.110000	189725.800000
338912.400000	-277777.800000	-254184.300000	152510.600000
462963.000000	-203347.500000	-316209.600000	50836.870000
1264838.000000	277777.800000	-948628.800000	569177.300000
-462962.900000	-758903.000000	-16945.620000	948628.800000
67782.480000	-1388889.000000	-50836.870000	762553.000000
92592.600000	-1016737.000000	-63241.910000	254184.300000
252967.700000	1388889.000000	-189725.800000	2845886.000000
-92592.590000	-3794515.000000	379451.500000	-33891.250000

-555555.600000	135565.000000	305021.200000	-101673.700000
-406694.900000	185185.200000	101673.700000	-126483.800000
555555.600000	505935.300000	1138355.000000	-379451.500000
-1517806.000000	-185185.200000	2.633975	7.886751E-01
0	0	45	46
39	40	37	38
35	36	43	44
0	0	0	0
-1422943.000000	-569177.300000	1897258.000000	-277777.800000
-474314.400000	-50836.870000	694444.500000	203347.500000
-381276.500000	-152510.600000	508368.700000	277777.800000
-127092.200000	-189725.800000	-694444.400000	758903.000000
-284588.600000	-2845886.000000	379451.500000	-1388889.000000
-94862.880000	-254184.300000	138888.900000	1016737.000000
-76255.300000	-762553.000000	101673.700000	1388889.000000
-25418.430000	-948628.800000	-138888.900000	3794515.000000
-1138355.000000	-569177.300000	-555555.600000	758903.000000
-101673.700000	-189725.800000	406694.900000	277777.800000
-305021.200000	-152510.600000	555555.600000	203347.500000
-379451.500000	-50836.870000	1517806.000000	-277777.800000
4.226497E-01	2.113249E-01	474314.400000	-50836.870000
-1897258.000000	-277777.800000	1422943.000000	-569177.300000
694444.400000	758903.000000	127092.200000	-189725.800000
-508368.700000	277777.800000	381276.500000	-152510.600000
-694444.500000	203347.500000	94862.880000	-254184.300000
-379451.500000	-1388889.000000	284588.600000	-2845886.000000
138888.900000	3794515.000000	25418.430000	-48628.800000
-101673.700000	1388889.000000	76255.300000	-762553.000000
-138888.900000	1016737.000000	-101673.700000	189725.800000
-555555.600000	-758903.000000	-1138355.000000	569177.300000
1517806.000000	277777.800000	-379451.500000	50836.870000
555555.600000	-203347.500000	-305021.200000	152510.600000
406694.900000	-277777.800000	1.577350	2.113249E-01
381276.500000	152510.600000	-508368.700000	-277777.800000
127092.200000	189725.800000	694444.400000	-758903.000000
1422943.000000	569177.300000	-1897258.000000	277777.800000
474314.400000	50836.870000	-694444.500000	-203347.500000
76255.300000	762553.000000	-101673.700000	-1388889.000000
25418.440000	948628.800000	138888.900000	-3794515.000000
284588.600000	2845886.000000	-379451.500000	1388889.000000
94862.880000	254184.300000	-138888.900000	-1016737.000000
305021.200000	152510.600000	-555555.600000	-203347.500000
379451.500000	50836.870000	-1517806.000000	277777.800000
1138355.000000	569177.300000	555555.600000	-758903.000000
101673.700000	189725.800000	-406694.900000	-277777.800000
1.577350	7.886751E-01	-127092.200000	189725.800000
508368.700000	-277777.800000	-381276.500000	152510.600000
694444.500000	-203347.500000	-474314.400000	50836.870000
1897258.000000	277777.800000	-1422943.000000	569177.300000
-694444.400000	-758903.000000	-25418.440000	948628.800000
101673.700000	-1388889.000000	-76255.300000	762553.000000
138888.900000	-1016737.000000	-94862.880000	254184.300000
379451.500000	1388889.000000	-284588.600000	2845886.000000
-138888.900000	-3794515.000000	379451.500000	-50836.870000

-55555.60000	203347.50000	305021.20000	-152510.60000
-406694.90000	277777.80000	101673.70000	-189725.80000
55555.60000	758903.00000	1138355.00000	-569177.30000
-1517806.00000	-277777.80000	4.226497E-01	7.886751E-01