



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**BANCO DE PROBLEMAS PARA EL PRIMER RALLY
DE COMPUTACIÓN INTERFACULTADES UNAM**

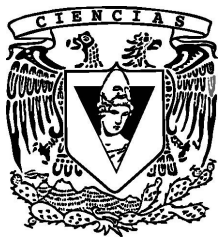
REPORTE DE DIVULGACIÓN

**QUE PARA OBTENER EL TÍTULO DE:
LICENCIADO EN CIENCIAS DE LA
COMPUTACIÓN**

P R E S E N T A:

DIEGO CALZADILLA ESTRADA

TUTORA: DRA. ELISA VISO GUROVICH



2013



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

Hoja de Datos del Jurado

1. Datos del Alumno

Calzadilla
Estrada
Diego
55495529
Universidad Nacional Autónoma de México
Facultad de Ciencias
Ciencias de la Computación
305241416

2. Datos del tutor

Dra
Elisa
Viso
Gurovich

3. Datos del sinodal 1

Dr
José David
Flores
Peñaloza

4. Datos del sinodal 2

Dr
José de Jesús
Galaviz
Casas

5. Datos del sinodal 3

Dra
María de Luz
Gasca
Soto

6. Datos del sinodal 4

M en C
Carlos
Zerón
Martínez

7. Datos del trabajo escrito

Banco de Problemas para el Primer Rally de Computación Interfacultades UNAM
50 p
2013

Índice

I	Introducción	5
II	Banco de problemas	9
III	Bibliografía	48

Parte I

Introducción

Introducción

Un reclamo frecuente entre los estudiantes de licenciatura, en particular los de Ciencias de la Computación, es la falta de relación directa entre los cursos que se imparten en la licenciatura con las últimas tecnologías en uso o problemas prácticos a resolver. No entraré en este trabajo en la discusión de si esto está bien diseñado o no, pero se debe reconocer que falta, entre los estudiantes de nuestra licenciatura, un contacto más claro con la solución de problemas y los conocimientos que adquieren en sus cursos.

El presente trabajo consiste en generar y capturar un banco de problemas relacionados con ciencias de la computación, para ser utilizado en el desarrollo del Primer Rally de Computación Interfacultades UNAM.

Este rally se organizó dentro del proyecto PAPIME PE103313 "Mejoramiento de la relación teórica-práctica en la licenciatura de Ciencias de la Computación", a cargo de la Dra. Elisa Viso Gurovich y el Dr. David Flores Peñaloza, ambos profesores de la Facultad de Ciencias, UNAM. Dicho proyecto tiene como objetivo, como su nombre lo indica, organizar actividades para que los alumnos de la licenciatura en Ciencias de la Computación conozcan, practiquen y aprendan de varias tecnologías y aplicaciones, en las cuales se ponen en uso los conocimientos que se imparten a lo largo de la carrera.

El Primer Rally de Computación Interfacultades UNAM tiene como objetivos el fomentar la socialización, el intercambio de ideas y la difusión de diferentes perspectivas de la computación entre el alumnado de las licenciaturas, enfocadas a este ámbito, de la Facultad de Ciencias y de la Facultad de Ingeniería. El rally se compone de dos fases, la primera consiste en un periodo en línea de calentamiento o preparación y la segunda en el desarrollo del rally en sí, que tendrá lugar en las instalaciones de ambas facultades el mes de Octubre del 2013. Durante la primera fase se buscó familiarizar a los alumnos con el estilo de los problemas que se utilizarán en la segunda, conectarlos con otros alumnos interesados en el rally, con los cuales podrán formar un equipo posteriormente, y mostrar la manera en que dichos problemas se resuelven con sus respectivos argumentos de correctud.

El banco de problemas desarrollado por este proyecto consiste de 150 problemas, con su respectiva solución, inspirados en el estilo usado por concursos de programación, como los organizados por la Association of Computing Machinery, TopCoder, Codeforces, entre otras sociedades e instituciones, así como los utilizados en las entrevistas de trabajo de distintas compañías de renombre mundial tales como Yahoo, Google, IBM, Microsoft, Amazon y otros. Para resolver un problema se le exige al alumno poseer una formación formal en computación, necesaria para generar los razonamientos lógicos que lo llevarán a la solución y que servirán para garantizar su correctud. El manejo práctico y de conceptos de lenguajes de programación también juega un papel importante en la solución de muchos de estos problemas, aunque se ha buscado que no se trate de un lenguaje particular o un paradigma único.

Situado entre paréntesis, al inicio de cada problema, se encuentra un número que lo asocia con las siguientes categorías:

1. Organización de sistemas de cómputo.
2. Algoritmos, estructuras de datos y complejidad.
3. Combinatoria y teoría de números.

4. Bases de datos.
5. Lógica, inteligencia artificial y computabilidad.
6. Autómatas y lenguajes formales.
7. Características de lenguajes de programación.

La razón de esta clasificación es proporcionar a los estudiantes de computación con distintas áreas a las que pudieran estar enfocadas sus licenciaturas, buscando en cada fase de calentamiento, o en el rally mismo, un balance entre el tipo de preguntas que se hacen. Adicionalmente, como lo indica el nombre del proyecto en el que se inscribe esta actividad, se pretende poner al alcance de los participantes conocimientos y, de alguna forma, experiencias, que no siempre es posible que se revisen en un curso curricular.

Parte II

Banco de problemas

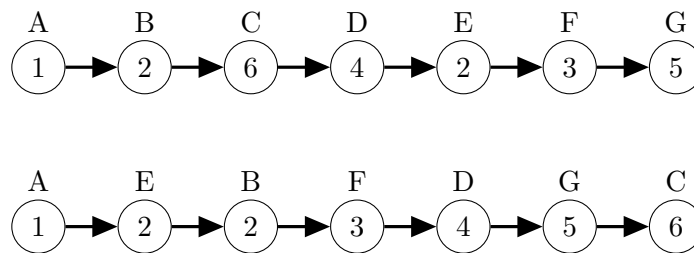
- (2) ¿Cuántos números del 1 al 100,000 (inclusive) cumplen que la suma de sus dígitos es menor a 20?
- (1) Un día el profesor Arquiloco le preguntó a su alumno Moore: Actualmente, en el año 2013, se sabe que en cierto planeta existen circuitos integrados con un único transistor. Según tú, aproximadamente, ¿cuántos transistores habrá en los circuitos integrados de dicho planeta para el año 2028?
(a) 10 (b) 64 (c) 256 (d) 1024
- (7) Si el siguiente pseudocódigo corresponde al de un lenguaje con evaluación en corto circuito, ¿qué regresaría la función f?

```

1 int f() {
2     int a=0;
3     int b=1;
4     try {
5         if (b==1 || 100/a ==10) return 1;
6         else return 0;
7     } catch (ArithmeticException) {
8         return -1;
9     }
10 }

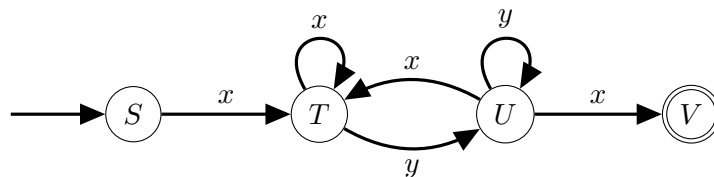
```

- (2) En una clase de algoritmos que toma Pancho "El inestable" sólo se permiten utilizar los algoritmos de ordenación, Heapsort, Insertionsort y Mergesort. Si Pancho utilizó alguno de estos para ordenar la siguiente lista de acuerdo a los enteros que contiene cada nodo, obteniendo el resultado mostrado, ¿qué algoritmo utilizó Pancho?



- (a) Heapsort (b) Insertionsort (c) Mergesort
- (2) Un día Daniel programó una versión iterativa del algoritmo de búsqueda en profundidad, así como su respectiva versión recursiva. Al día siguiente le contó a todos sus amigos que dada una gráfica suficientemente grande como entrada, al utilizar la versión recursiva su programa terminaba con un error de ejecución mientras que con la iterativa todo corría con normalidad. ¿Es posible que Daniel haya dicho la verdad o necesariamente cometió un error en alguna de sus implementaciones? Fundamenta tu respuesta.

6. (2) En cierta implementación un entero positivo n se representa en binario como una lista ligada asociando la cabeza al bit menos significativo. Por ejemplo, la lista ligada asociada a $n = 12$ es: $0 \rightarrow 0 \rightarrow 1 \rightarrow 1$. Pedro diseñó una función eficiente que recibe un apuntador a la cabeza de una lista ligada que representa un entero n y modifica dicha lista para representar ahora a $\lfloor \frac{n}{4} \rfloor$. Utilizando notación de complejidad, representa la de dicha función.
7. (4) En una mini base de datos que sólo almacena nombres se tiene que los únicos atributos de una entidad Persona, son: $\langle \text{Nombre} \rangle$, $\langle \text{Apellido Paterno} \rangle$ y $\langle \text{Apellido Materno} \rangle$. Si se sabe que los siguientes nombres son los únicos que tienen que almacenarse en la base de datos: Juan Lozano Vieyra, María Villa Cruz, Juan Barajas Torres, Pedro Manjarrez Torres, Daniel Calvo Mena y Lucía Barajas Esquivel, ¿cuál es el mínimo número de atributos que se tienen que utilizar para formar la llave de dicha entidad?
8. (2) Implementa un método o función en el lenguaje que gustes, que determine si una lista ligada es cíclica o no utilizando menos de 15 líneas de código.
9. (6) Describe con una expresión regular el lenguaje aceptado por el siguiente autómata finito:



10. (2) Una función de dispersión mapea entradas de 8 bits a valores de 4 bits. Si se almacenan al menos 40 entradas podemos garantizar que al menos hay n entradas que la función mapea al mismo valor ¿Cuál es el valor de n ?
11. (1) El siguiente pseudocódigo sirve para verificar si un entero positivo j es _____.
if **((int)(j & (j-1)) == 0)** **return true** ;
12. (3) Ian y Marco juegan a lanzar volados. Si al final salieron más águilas que soles Ian gana; Marco gana en caso contrario. Si en total lanzan 4 volados ¿Cuál es la probabilidad de que Marco gane?
13. (1) Un árbol binario lleno es un árbol binario donde todo nodo interno tiene 2 hijos. Si un árbol binario lleno tiene h hojas, ¿es posible saber cuántos nodos internos tiene? Justifica tu respuesta.
14. (3) Si se tienen dos enteros positivos m y n tales que su producto es 200 y su máximo común divisor es 4, encuentra su mínimo común múltiplo.
15. (2) Un arreglo es "realista" si se cumple que:

1. Contiene exactamente un elemento "forever alone".
2. Todos sus elementos son enteros.
3. Todo elemento que no es "forever alone" se encuentra exactamente dos veces dentro del arreglo.
4. El elemento "forever alone" se encuentra exactamente una vez dentro del arreglo.

Elabora una función que se ejecute en tiempo lineal, utilice memoria constante y que reciba un arreglo "realista" y devuelva su elemento *forever alone*.

16. (3) Elabora un método o función que reciba tres enteros a , b y c e imprima si el triángulo formado por dichas longitudes es escaleno ('e'), isósceles ('i'), equilátero ('q') o degenerado ('d').
17. (2) La maestra Linealina le pidió a su alumno Recursio que elaborara una función que recibiera el apuntador al nodo cabeza de una lista ligada de enteros y devolviera un apuntador al nodo cabeza de la lista resultante al invertir el orden por parejas de nodos. Es decir, si la lista original era: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$, la función debe volver el apuntador a la cabeza de la lista: $B \rightarrow A \rightarrow D \rightarrow C \rightarrow E$. Elabora una función eficiente y con menos de 15 líneas de código para ayudar a Recursio.
18. (5) En el país de Logimex, el senado está compuesto por 101 personas y cada senador es o bien del clan de la mentira, es decir, siempre dice mentiras; o bien del clan de la verdad, es decir, siempre dice la verdad. Por recortes de presupuesto se tiene que sacar a un senador, pero cada uno de ellos argumentó que si lo sacaban a él, la mayoría de los senadores restantes serían del pueblo de la mentira. ¿Cuántos senadores del clan de la verdad hay en el senado, si todavía no se ha decidido sacar a nadie?
19. (2) Konán Count tiene que elaborar un método o función que reciba dos cadenas de longitud menor a 100 formadas por caracteres ASCII sin espacios y determine si una es un anagrama de la otra. Ayuda a Konán a elaborar dicha función en el lenguaje que gustes. Esta función debe ejecutarse en tiempo lineal y utilizar memoria constante.
20. (2) Elabora una función que reciba dos cadenas de letras minúsculas y verifique si una es rotación de la otra. Tu función no debe utilizar más de 15 líneas de código. Ejemplo: perrote es rotación de roteper.
21. (2) El siguiente código elimina el nodo cabeza de una lista ligada de enteros en C++. Encuentra al menos dos errores en su implementación.

```
typedef struct nodoT {
    int data;
    struct nodoT *next;
} nodo;
```

```
void remueveCabeza(nodo *head) {
    head=head->next;
    free(head);
}
```

O si te sientes más a gusto, encuentra los errores en el código de Java, que son exactamente los mismos, suponiendo que la clase `Nodo` está definida acorde.

```
public static void remueveCabeza(Nodo cabeza) {
    cabeza = cabeza.sig;
    try {
        cabeza.finalize(); // Elimina el espacio del heap
    } catch(Throwable e) {
        System.err.println("Marcó error");
    }
}
```

22. (1) Una computadora que distribuye su banda ancha equitativamente está descargando dos archivos. Si en un instante el tiempo esperado de descarga del primero es de 1 hora, mientras que el del segundo es de 30 minutos. ¿Cuánto tiempo tardará en descargar ambos archivos?
23. (1) Un switch de paquetes recibe un paquete X y determina el enlace externo por el cuál el paquete debe ser reenviado. Cuando dicho paquete llega, existe otro que se encuentra en la mitad del proceso de ser transmitido por dicho enlace y otros tres se encuentran en la lista de espera. Si cada paquete está compuesto por 2,000 bytes y la tasa del enlace es de 2Mbps aproximadamente, ¿cuál es el valor del retraso de encolamiento del paquete X?
24. (1) ¿Cuántos bytes hay en un Megabit?
25. (7) ¿Qué entero regresa la siguiente función?

```
1 int f() {
2     unsigned int a;
3     int b;
4     b=0;
5     for(a=100; a>=0; a--) {
6         if(b==200) return 0;
7         b++;
8     }
9     return 1;
10 }
```

26. (3) A una fiesta asisten 20 personas. Si todos saludan a cada uno de los otros invitados una única vez, ¿cuántos saludos se dieron en total?
27. (1) Elabora un método o función para saber si la arquitectura de almacenamiento en memoria de datos de tu computadora es Little Endian o Big Endian.
28. (3) Si tenemos 32 personas, donde no hay dos que tengan la misma estatura, ¿es posible con sólo 35 preguntas saber quiénes son las dos más altas? Nota: Las preguntas son del estilo “¿Es más alto A que B?”.

29. (1) Menciona 2 aplicaciones para las que es natural pensar en una arquitectura de red P2P.
30. (2) Mone Dita tiene una obsesión compulsiva con el cambio, por lo que continuamente se pregunta de cuántas maneras puede sumar una cantidad de centavos x , utilizando únicamente monedas de 1, 5, 10, 25 y 50 centavos. Elabora un programa que ayude a Mone Dita a lidiar con su obsesión. Tu programa debe poder procesar cantidades de hasta 3000 centavos y terminar cuando la cantidad que se le de sea 0 o un EOF.
31. (2) Don Adhoc tiene una fila de macetas pero vive en un departamento, por lo que necesita de lámparas especiales para que sus plantas puedan crecer. Una lámpara se puede situar arriba de una maceta que tenga o no una planta y ésta iluminará a dicha maceta y las dos adyacentes a ella. Ayuda a Don Adhoc con un programa para obtener el mínimo número de lámparas necesarias para iluminar sus plantas. Dicho programa debe cumplir las siguientes especificaciones:
1. El primer renglón de la entrada representa el número de casos t a procesar.
 2. Los siguientes $2t$ renglones de la entrada representan los casos de entrada. El primer renglón de cada par de líneas contiene un entero n que representa el número de macetas en la fila; el segundo renglón es una cadena de n caracteres formados por "." y "X", donde un "." representa una maceta con planta y una "X" una maceta sin planta.
 3. El programa debe imprimir un renglón por cada caso con el siguiente formato: "Caso <#Caso>: <Respuesta>".
32. (2) El Maestro Cruz le pidió a su alumno Productavio que elaborara un método o función que reciba las coordenadas enteras $x_a, y_a, x_b, y_b, x_c, y_c, x_d$ y y_d de cuatro puntos a, b, c y d en posición general e imprima: "Mismo lado" si c y d se encuentran del mismo lado de la recta formada por a y b o "Lado opuesto" en caso contrario.
33. (2) El sabio Bin–Bin siempre dice la verdad. Cuentan que alguna vez al encontrarse con un comerciante le dijo lo siguiente: "Comerciante, piensa en un número entero mayor a 0 y menor a 32. Si me respondes con la verdad, en n preguntas de respuesta afirmativa o negativa podré adivinar tu número. ¿Cuál es el valor de n ?"
34. (3) Un cierto pueblo de organismos celulares duplica su población cada segundo. Si a las 7 horas con 30 minutos y 12 segundos del 14 de febrero del 2013 había 5600 de dichos organismos celulares, ¿en qué momento hubo en el pueblo 1400 de los mismos organismos?

35. (2) ¿Cuál es la complejidad de la siguiente función?

```
1 void f(int n) {
2     int den=1;
3     int cont;
4     while( n / den ) {
5         for(cont = 0; cont < (n/den); cont++) {
6             printf("VIVO\n");
7         }
8         den*=2;
9     }
10 }
```

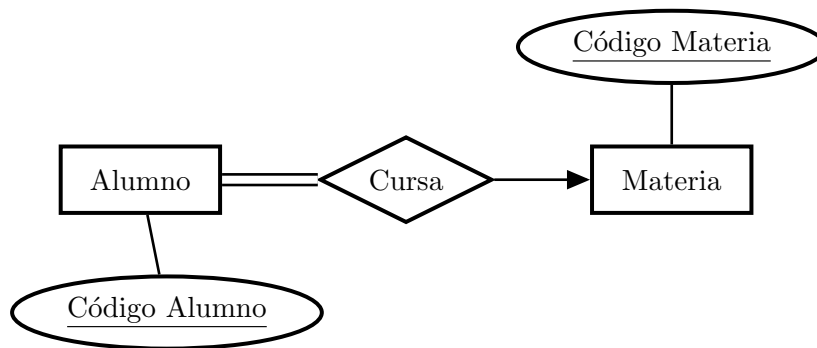
36. (2) ¿Cuál es la complejidad de la siguiente función?

```
1 void f(int n) {
2     for(int i=0; i<n; i++) {
3         for(int j=i; j<n; j++) {
4             printf("VIVO\n");
5         }
6     }
7 }
```

37. (7) Si sabemos que en alguna implementación en Java se encuentran las siguientes líneas, ¿qué valor tiene la variable c al final de las líneas mostradas? Nota: La primera presencia de objetos de la clase B se muestra en estas líneas y se encuentran dentro del método main.

```
1 public class B {
2     ...
3     private static int a=0;
4     ...
5     public B() {
6     }
7     public int getA() {
8         return a;
9     }
10    ...
11    public void f() {
12        a++;
13    }
14    ...
15 }
16 ...
17 B x;
18 B y;
19 y = new B();
20 x = new B();
21 int c;
22 x.f();
23 x.f();
24 y.f();
25 c = y.getA();
26 ...
```

38. (1) Sea $x = 1001001001001001001001$. Si x está escrito en base 2013, ¿Cómo escribimos a x en base 2013³?
39. (3) ¿Cuántos ceros tiene 500! en su expresión decimal?
40. (3) Si tengo 424,242,424,242 paletas, 228,228,228,228 chicles y 116,116,116,116 bombones y no planeo desperdiciar ninguno ¿Cuál es el máximo número de paquetes de regalo que puedo formar, de tal manera que todo par de paquetes tengan el mismo número de paletas, chicles y bombones?
41. (4) Contesta verdadero o falso teniendo el esquema de Entidad/Relación de la siguiente página.
- (a) Todo alumno tiene que estar cursando alguna materia. (a) falso (b) verdadero
- (b) Alumno es una entidad débil. (a) falso (b) verdadero
- (c) Una materia puede tener varios alumnos cursándola. (a) falso (b) verdadero
- (d) Un alumno puede cursar más de una materia a la vez. (a) falso (b) verdadero



42. (4) En SQL, ¿cuál es la diferencia entre usar varchar(n) y char(n) al declarar atributos?
43. (4) Encuentra el error en el siguiente código en SQL:

```

1 CREATE DATABASE tiendadeautos;
2 CREATE TABLE empleado(
3     numempleado varchar(20) PRIMARY KEY,
4     nombrecompleto char(30),
5 );
6 CREATE TABLE autovendido(
7     idcompra integer PRIMARY KEY,
8     numvendedor char(30) REFERENCES empleado(numempleado)
9 );
  
```

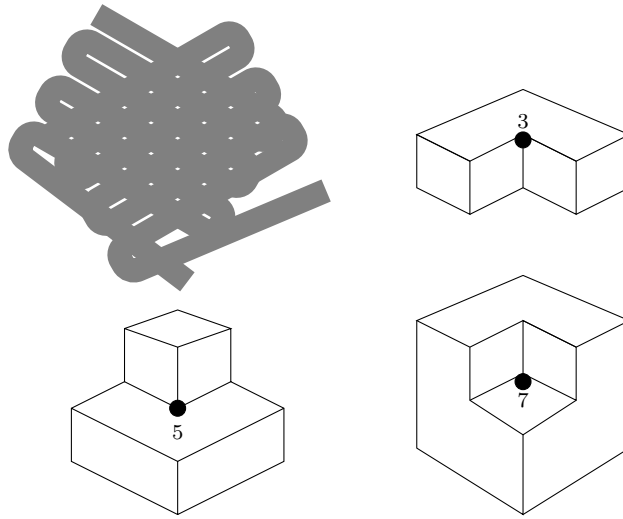
44. (4) Supongamos que a es un ejemplar (objeto) de una clase A . Tenemos que el ejemplar a es a la clase A lo que una fila es a una _____ en una base de datos.
45. (4) ¿Cuál es la diferencia entre una base de datos, un sistema manejador de datos y un sistema de base de datos?
46. (4) En un sistema de bases de datos se espera que una transferencia sea una unidad lógica con ciertas propiedades. Menciona dos.
47. (5) Un agente que tenga de función ser el piloto automático de un taxi se maneja en un entorno
 (a) Plano (b) Continuo (c) Discreto (d) Episódico
48. (5) El agente representado por un software que sea capaz de jugar ajedrez se maneja en un entorno
 (a) Episódico y discreto (b) Secuencial y discreto (c) Episódico y continuo
 (d) Secuencial y continuo
49. (5) En el año 2023 IAn diseñó un computador al cual se le aplicó la prueba de Turing. Durante la prueba el computador respondió erróneamente la pregunta ¿cuál es la raíz cuadrada de 4096? ¿Es correcto afirmar que dicho computador no pasó la prueba? Argumenta tu respuesta.
50. (5) Tenemos un tablero de tamaño $n \times n$, donde cada casilla contiene un entero que puede ser positivo o negativo. Un agente se encuentra en la esquina superior izquierda y busca llegar a la esquina inferior derecha sin pasar dos veces por la misma casilla y minimizando la suma total de las casillas por las que pasó. Si el algoritmo de búsqueda de dicho agente se basa en el algoritmo A^* utilizando como heurística la función que regresa el entero de la respectiva casilla, ¿es correcto asegurar que dicho agente encontrará la solución óptima?.
51. (2) Explica por qué en una gráfica donde todo arco tiene el mismo peso (o costo), la búsqueda primero en profundidad no es óptima mientras que la de primero en amplitud sí lo es. Nota: Óptima hace referencia a la solución devuelta por el algoritmo.
52. (5) Encuentra el error en la siguiente demostración por inducción donde se pretende demostrar que cualesquiera n puntos son colineales.

B.I.: Todo par de puntos son colineales ($n = 2$).

H.I.: Si se tienen cualesquiera k puntos, éstos son colineales ($n = k$).

P.I: Para $n = k + 1$, tenemos que si numeramos todos los puntos del 1 al $k + 1$, por H.I. tenemos que los puntos del 1 al k son colineales y que aparte los puntos del 2 al $k + 1$ también son colineales; pero ambas rectas comparten los puntos del 2 al k , por lo que son la misma recta y los $k + 1$ puntos son colineales como queríamos demostrar.

53. (5) El alumno Clowes trazó 4 esquemas en un pizarrón. Sin embargo, su malévolo compañero Huffman borró uno de los cuatro esquemas, resultando así la siguiente imagen.



Proporciona un esquema equivalente al faltante.

54. (5) Desde el punto de vista de la inteligencia artificial, ¿cuál es la diferencia entre actuar como humano y pensar racionalmente?
55. (4) ¿Qué tabla genera la siguiente expresión de álgebra relacional?

$$\Pi_{\text{nombre, appat, apmat}}(\sigma_{\text{Colonia}=\text{''Doctores''}}(\text{Empleado}))$$

56. (5) Tino Comparán no puede comprender por qué se maneja que *counting sort* "tiene complejidad lineal", si su maestra le había dicho que todo algoritmo de ordenación basado en comparaciones se encuentra acotado inferiormente por $n \log n$.
57. (5) La frase "La estrella en la pared" presenta una ambigüedad de tipo
(a) sintáctica (b) semántica
58. (3) El dominó aldeano es como un dominó normal con la excepción de que en vez de tener fichas cuyos valores interiores varían del 0 al 6, en el aldeano se cuenta con valores del 0 al 2013. ¿Cuántas fichas existen en el dominó aldeano?
59. (3) Si la probabilidad de que llueva en un día de cielo nublado es de 0.8, mientras que la probabilidad de que llueva y se trate de un día aleatorio es de 0.2, ¿cuál es la probabilidad de que un día aleatorio se encuentre nublado?
60. (2) Construye un árbol rojinegro tal que la trayectoria más larga de la raíz a una hoja tenga 10 vértices rojos y 10 vértices negros.

61. (6) Romeo y Julieta se quieren enviar mensajes de amor que sólo ellos entiendan. Codifican cada letra del alfabeto y definen dos máquinas, una que dado el mensaje lo codifique, y otra que dado el mensaje codificado lo descifre. En la jerarquía de lenguajes formales, ¿cuál es el tipo más sencillo de máquinas que nos pueden servir para cada una de las tareas (codificar y descifrar)?

62. (3) Diana Bunny ama los números y especialmente le encanta contar cosas. El otro día se preguntó: Dados dos enteros m y n con $m \leq n$, ¿cuántos cuadrados perfectos x cumplen que $m \leq x \leq n$? Elabora un programa para ayudar a Diana Bunny a resolver su duda. Tu programa debe cumplir las siguientes especificaciones:

- (a) Cada línea de la entrada contiene dos enteros m y n . La línea donde $m = 0$ y $n = 0$ determina el fin de la entrada y no debe ser procesada.
- (b) Tu programa debe resolver cada caso en tiempo constante.
- (c) La salida de cada caso de prueba debe mostrarse en una línea independiente, con el formato mostrado a continuación.
- (d) Ejemplo de entrada:

```
1 2013
1 100
9 9
0 0
```

- (e) Ejemplo de salida:

```
44
10
1
```

63. (2) Hace mucho tiempo el mundo de Enteropia se encontraba dividido en Enteropia del Norte y Enteropia del Sur. Cada uno tenía su bandera y cada bandera estaba formada por varios lienzos del mismo tamaño y donde cada lienzo tenía escrito un número entero. Después de la revolución que unió Enteropia, la emperadora Isabel te encomendó elaborar un programa que dadas dos banderas, encuentre la longitud de la bandera más grande cuyos colores se puedan encontrar en el mismo orden que en las dos banderas proporcionadas, aunque no estén contiguos. Tu programa debe cumplir con las siguientes características:

- (a) La entrada consiste de varios casos de prueba. Cada caso de prueba describe un par de banderas. El caso 0 0 debe terminar el programa.
- (b) La primera línea de cada caso de prueba contiene dos enteros n y s menores a 101, indicando la longitud de la bandera del Norte y del Sur, respectivamente. Las siguientes dos líneas contienen n y s enteros, respectivamente, representando los colores en orden de cada una de las banderas.
- (c) La salida de cada caso de prueba debe mostrarse en una línea independiente que contenga el entero que representa la longitud de la bandera pedida.

(d) Ejemplo de entrada:

```
4 5
10 12 20 20
20 20 20 20 20
6 3
10 12 12 11 11 11
10 11 12
0 0
```

(e) Ejemplo de salida:

```
2
2
```

64. (2) Se dice que una cadena es k -periódica si puede formarse con la concatenación de una o más cadenas iguales de longitud k . Escribe una función que reciba una cadena y devuelva la mínima k para la cuál dicha cadena es k -periódica.

65. (3) Sea $f(n)$ la función que recibe un natural menor a 200,000,000 y devuelve la suma de sus dígitos. Tu misión es elaborar un programa que calcule $f^{2013}(n)$, donde el 2013 indica 2013 composiciones sucesivas al resultado obtenido. Debe cumplir las siguientes condiciones:

(a) La entrada consiste de varias líneas, cada una con un entero, del cual hay que calcular $f^{2013}(n)$. Una línea con un 0 determina el fin de los datos y no se debe procesar.

(b) La salida de cada caso de prueba debe mostrarse en una línea independiente.

(c) Ejemplo de entrada:

```
29111421
49323
3333
0
```

(d) Ejemplo de salida:

```
3
3
3
```

66. (2) En el changarro de Don Rata actualmente trabajan tres despachadores. Sin embargo, dada la crisis económica, Don Rata busca despedir al que gana más y al que gana menos. Elabora un programa que devuelva el sueldo en centavos de la persona que conservará su trabajo.

(a) La primera línea de la entrada contiene un entero n .

(b) El resto de la entrada consiste de n líneas, cada una con 3 números distintos, donde cada uno representa un salario en centavos de algún despachador.

(c) La salida de cada caso de prueba debe mostrarse en una línea independiente con el formato mostrado a continuación.

(d) Ejemplo de entrada:

```
3
100 200 300
101 102 103
10000 20000 10300
```

(e) Ejemplo de salida:

```
Caso 1: 200
Caso 2: 102
Caso 3: 10300
```

67. (2) En la oficina de la compañía DLB (Desde La Base) tu jefe (sí, así es, no lo sabías pero eres un empleado de DLB) te ha puesto a implementar tu propio programa para identificar si una cadena es subsucesión de otra, donde una cadena es subsucesión de otra cadena si podemos remover, de la segunda, un número finito de caracteres y obtener la primera. Sin embargo, por cuestiones legales, no se te ha permitido utilizar ningún tipo string, por lo que tu función debe manejarse únicamente con arreglos de caracteres. Además, tu función debe cumplir las siguientes especificaciones.

(a) El encabezado (o prototipo) debe ser `bool subsucesion(char *p1, int m, char *p2, int n)`

(b) `p1` es el primero arreglo de caracteres de longitud `m`.

(c) `p2` es el segundo arreglo de caracteres de longitud `n`.

(d) La función debe de devolver `true` si `p1` es subsucesión de `p2`.

(e) Ejemplos:

```
subsucesion(roto , 4, alboroto , 8)=true
subsucesion(sucio , 5, suplicio , 8)=false
subsucesion(TODO, 4, TodoSeFueDeNuestroOrgullo , 25)=false
subsucesion(lalala , 6, LAAlaLAlaLAlaLA , 14)=true
```

68. (2) El otro día Manolito recordaba cómo en la primaria, al sumar dos números, siempre que era necesario decía en voz alta "y llevo 1" cuando la suma de dos dígitos sobrepasaba al 9. Para recordar viejos tiempos, Manolito te pidió que elaboraras un programa que, utilizando memoria constante, reciba una serie de parejas de enteros menores a 100,000,000,000 e imprima cuántas operaciones de "llevo 1" se realizarían al sumarlos al estilo Manolito. Tu programa debe cumplir las siguientes condiciones.

(a) Cada pareja de enteros de entrada irá en una línea separada. La pareja 0 0 determinará el fin de los casos de entrada.

(b) La salida de cada caso de prueba debe mostrarse en una línea independiente con el formato mostrado a continuación.

(c) Ejemplo de entrada:

```
999 999
100 200
988 111
0 0
```

(d) Ejemplo de salida:

```
3 ops de acarreo .
Ninguna op de acarreo .
1 op de acarreo .
```

69. (2) En la empresa RSS (Restricciones Sin Sentido) trabajan 10 programadores, todos ellos bajo la autoridad de un jefe un poco "especial". Tan "especial" que a la hora de descanso los programadores tienen que formarse en una fila de acuerdo al largo de su cabello, ya sea de manera ascendente o descendente. Elabora un programa que ayude al jefe a ver si sus empleados están formados de la manera correcta. Tu programa debe cumplir las siguientes condiciones:

- (a) En tu programa sólo puedes utilizar dos instrucciones de control de tipo if y ninguna de tipo switch.
- (b) La primera línea de la entrada contiene un entero n que corresponde al número de renglones con los diez programadores formados van a venir.
- (c) Las siguientes n líneas contienen 10 enteros cada una representando la fila de empleados y el largo del cabello de cada uno.

Ejemplo de entrada:

```
3
100 200 300 301 302 303 304 305 306 307
1000 900 800 700 600 500 400 300 200 100
20 98 13 25 65 29 38 999 30 20
```

(d) La salida de cada caso de prueba debe mostrarse en una línea independiente con el formato mostrado a continuación.

Ejemplo de salida:

```
Ordenados
Ordenados
Desordenados
```

70. (3) Diana Bunny ama las matemáticas y muchas veces sus amigos programadores la molestan por ello. Para vengarse, Diana Bunny suele ponerles problemas de programación donde el ser ignorante en diversos temas de matemáticas suele tener graves consecuencias. Y es así como este problema ha llegado a tus manos. Si en una mesa circular se encuentran n pares de personas con brazos muy muy largos, queremos saber cuántas formas distintas hay de que se saluden de mano al mismo tiempo por parejas, de tal manera que no se crucen dichos saludos. Elabora una función para resolver dicho problema y que cumpla con las siguientes especificaciones:

- (a) El encabezado o prototipo de la función debe ser `long long cuentaFormasSaludar(int n)`.
- (b) n es el número de pares de personas sentadas en la mesa con $n < 20$.

(c) Ejemplos:

```
cuentaFormasSaludar(1)=1
cuentaFormasSaludar(2)=2
cuentaFormasSaludar(3)=5
cuentaFormasSaludar(4)=14
```

71. (2) Este Domingo la UNAM y el Poli jugaron un partido de americano. Sin embargo, al tener tanta tarea de cálculo no pudiste asistir a ver el partido y lo grabaste. Al pasar por un metro escuchaste cuántos puntos m se anotaron en total, quién gano y cuál fue la diferencia n de puntajes entre ambos equipos. Elabora dos funciones que reciban dos enteros m y n , una devuelva el puntaje del equipo ganador y la otra el del perdedor. Ambas funciones deberán de regresar -1 si es imposible dichos resultados. Tus funciones deben cumplir las siguientes especificaciones.

- (a) El encabezado o prototipo de la primera función debe ser `int puntajeGanador(int m, int n)`.
- (b) El encabezado o prototipo de la segunda función debe ser `int puntajePerdedor(int m, int n)`.
- (c) m es el total de puntos anotados.
- (d) n es la diferencia de puntos.
- (e) Puedes utilizar las funciones, estructuras y variables auxiliares que gustes.
- (f) Ejemplos de uso de las funciones:

```
puntajeGanador(30, 10)=20
puntajePerdedor(30, 10)=10
puntajeGanador(40, 60)=-1
puntajePerdedor(40, 60)=-1
```

72. (2) Al profesor Gabriel Rafael Almanza Farías le gusta resolver laberintos en el periódico. Sin embargo, últimamente anda muy ocupado por lo que te ha pedido que elabores un programa que reciba un laberinto y determine si dicho laberinto tiene solución o no. Tu programa debe cumplir las siguientes condiciones:

- (a) La primera línea de la entrada contiene un entero n con $n \leq 50$. El final de la entrada estará determinada por el entero $n = 0$.
- (b) El resto de la entrada consiste de n líneas, representando cada una las n casillas del tablero en esa línea. Una "X" representa que en dicha casilla hay muro, mientras una "O" representa que dicha casilla se encuentra libre. La casilla de inicio siempre será la esquina superior izquierda y la meta será la esquina inferior derecha; en ninguna de las dos habrá muro.

Ejemplo de entrada:

```
3
OXO
OXO
OXO
```

```
7
OOXXOO
XOOOOOX
XOXXOX
XOXXOX
XOXXOX
XOXXOX
XXXXOO
0
```

(c) La salida de cada caso de prueba debe mostrarse en una línea independiente con el formato mostrado a continuación.

Ejemplo de salida:

```
Imposible
Posible
```

73. (3) Diana Bunny se puso a escribir en el pizarrón números pares en grupos con un número impar de pares, de la siguiente manera:

```
0
2 4 6
8 10 12 14 16
```

Como a Diana le gustan mucho los números, suele preguntarse cosas sobre ellos. Un día se preguntó cuál sería la suma de los últimos tres pares de la línea en que escribió n pares. Sin embargo, cuando regresó al salón donde había escrito sus preciados números, los encontró borrados. Desde entonces la pregunta no la deja dormir por las noches. Elabora una función que ayude a resolver dicha duda a Diana para que pueda volver a dormir en paz por la noche. Tu función debe cumplir las siguientes especificaciones.

- (a) El prototipo o encabezado de la función debe ser `unsigned long long encuentraSuma(int n)`.
- (b) n es el impar asociado a la línea donde fueron escritos n pares con $n > 2$.
- (c) Diana te ha garantizado que el resultado de la suma buscada no sobrepasa de $2^{63} - 1$.
- (d) Puedes utilizar las funciones, estructuras y variables auxiliares que gustes.
- (e) Ejemplos:

```
encuentraSuma(3)=12
encuentraSuma(5)=42
encuentraSuma(2013)=6084282
```

74. (2) Adriana y Alex juegan un juego. Primero ponen n piedras en una mesa y elaboran una lista de $m > 1$ enteros distintos en un papel. Dicha lista siempre contiene al 1. Después, por turnos, van quitando piedras de la mesa de tal manera que siempre quitan exactamente un número de piedras igual a alguno escrito en la lista que elaboraron. Gana el que quita la última piedra (pierde al que le toca jugar cuando ya no hay piedras en la mesa). Elabora una función que,

dados la n y los m enteros de la lista en un arreglo, devuelva *true* si Adriana tiene estrategia ganadora y *false* si la tiene Alex. Tu función debe cumplir las siguientes especificaciones.

- (a) El prototipo (o encabezado) de la función debe ser `bool adrianaGana(int n, int m, int *p)`.
- (b) n es el número de piedras con $n < 10000$.
- (c) m es la cantidad de números en la lista con $m < 15$.
- (d) p es el arreglo que representa la lista de números.
- (e) El 1 siempre debe ser parte de la lista de enteros.
- (f) Puedes utilizar las funciones, estructuras y variables auxiliares que gustes.

Ejemplos:

```
p[1] = {1};
adrianaGana(3, 1, p) = true
adrianaGana(4, 1, p) = false
p[2] = {1, 2};
adrianaGana(5, 2, p) = true
p[3] = {1, 2, 4};
adrianaGana(6, 3, p) = false
```

75. (2) Gerardo tiene un tablero de $2n \times 2n$ unidades dividido en cuadritos de una unidad por lado. Carlos ha decidido trazar un círculo de diámetro $2n - 1$ con la punta del compás situada en el centro del tablero de Gerardo. Elabora dos funciones que determinen cuántas casillas se encuentran totalmente dentro del círculo y cuántas intersectan su circunferencia. Tu función debe cumplir las siguientes especificaciones.

- (a) El prototipo (encabezado) de la primera función debe ser `int dentroCirculoTotalmente(int n)`.
- (b) El prototipo (encabezado) de la segunda función debe ser `int intersectanCirculo(int n)`.
- (c) n es el entero asociado al tablero de $2n \times 2n$.
- (d) Puedes utilizar las funciones, estructuras y variables auxiliares que gustes.

Ejemplos:

```
dentroCirculoTotalmente(1) = 0
intersectanCirculo(1) = 4
dentroCirculoTotalmente(5) = 52
intersectanCirculo(5) = 36
```

76. (2) Diana Bunny viene a molestarte de nuevo. Esta vez va a jugar contigo un juego de geometría. Ella te va a dar 8 enteros $x_{inicio}, y_{inicio}, x_{fin}, y_{fin}, x_{supizq}, y_{supizq}, x_{infder}$ e y_{infder} y tú tienes que determinar si el segmento de recta formado por los puntos (x_{inicio}, y_{inicio}) y (x_{fin}, y_{fin}) tiene al menos un punto en común con el rectángulo cuya esquina superior izquierda es el punto (x_{supizq}, y_{supizq}) y cuya esquina inferior derecha es el punto (x_{infder}, y_{infder}) . Dicho rectángulo se considera que abarca todos los puntos tanto de su perímetro como de la región contenida dentro de él. Sin embargo, al ser tu pasión la programación, has decidido elaborar una función para vencer a Diana Bunny en su propio juego. Tu función debe cumplir las siguientes especificaciones.

- (a) El encabezado o prototipo de la función debe ser
`bool puntoComun(int xi, int yi, int xf, int yf, int xsi, int ysi, int xid, int yid).`
- (b) x_i e y_i son las coordenadas del inicio del segmento.
- (c) x_f e y_f son las coordenadas del final del segmento.
- (d) x_{si} e y_{si} son las coordenadas de la esquina superior izquierda del rectángulo.
- (e) x_{id} e y_{id} son las coordenadas de la esquina inferior derecha del rectángulo.
- (f) Tu función debe devolver `true` si comparten un punto y `false` en caso contrario.
- (g) Puedes utilizar las funciones, estructuras y variables auxiliares que gustes.
- (h) Ejemplos:

```
puntoComun(4, 9, 11, 2, 1, 5, 7, 1)=false
puntoComun(0, 0, 5, 5, 2, 2, -4, -4)=true
```

77. (3) Una pareja de primos gemelos es una pareja de la forma $(p, p + 2)$ donde ambas entradas son primos. Tu trabajo es calcular la n -ésima pareja de primos gemelos con $n \leq 100,000$. Te podemos asegurar que dicha pareja no ocupa números mayores a 20,000,000. Tu función debe cumplir las siguientes especificaciones:

- (a) Debe de ser implementada en C/C++.
- (b) Debe de utilizar la siguiente estructura donde $x < y$:

```
1 typedef struct parejaT {
2     int x;
3     int y;
4 } pareja;
```

- (c) El prototipo de la función debe ser `pareja primosGemelos(int n)` donde n corresponde a la n -ésima pareja que tu función tiene que regresar.
- (d) Ejemplos:

```
primosGemelos(1)=[3, 5]
primosGemelos(100)=[3821, 3823]
primosGemelos(100000)=[18409199, 18409201]
```

78. (2) Cuando era joven a tu maestro de estructuras de datos le gustaba organizar filas de dominós y ver cómo todos caían en efecto cadena. Por ello, te ha puesto a elaborar una función que, dada una configuración de piezas de dominó, determine cuántas piezas se caen al tirar un subconjunto de ellas con la mano. Tu función debe cumplir las siguientes condiciones.

- (a) El encabezado o prototipo de la función debe ser
`int cuentaDominos(int n, int *a, int *b, int m, int *p, int l).`
- (b) n es el número de fichas de dominó que pueden estar formadas, numeradas del 1 al n con $n \leq 10,000$.
- (c) a y b son dos arreglos de longitud m , que representan la relación de como se acomodaron las fichas, representando que si se cae a_i ésta tira a la ficha b_i .

- (c) Ninguna de las cadenas producidas tiene un número impar de dígitos 1 consecutivos.
(a) Falso (b) Verdadero
- (d) Toda cadena producida tiene al menos tantos dígitos 1 como dígitos 0.
(a) Falso (b) Verdadero
82. (1) Supongamos que tenemos un procedimiento recursivo y que podemos garantizar que termina. ¿Cuáles de las siguientes afirmaciones son necesariamente ciertas?
- (a) El procedimiento tiene una variable local. (a) Falso (b) Verdadero
- (b) El procedimiento tiene al menos un parámetro. (a) Falso (b) Verdadero
- (c) El procedimiento tiene al menos un camino de ejecución donde no se llama a si mismo.
(a) Falso (b) Verdadero
83. (3) Cierta software de graficación requiere que para cada pareja de pixeles adyacentes el nivel de gris entre ellos no difiera en más de dos unidades. Si se tienen n niveles de gris (del 0 al $n - 1$), ¿cuántas de las n^2 posibles configuraciones de grises que hay de una pareja de pixeles son válidas?
84. (2) Si tenemos tres matrices A, B y C de tamaño $m \times n$, $n \times p$ y $p \times q$, respectivamente, ¿cuál de las siguientes condiciones es necesaria para que al calcular el producto ABC como (AB)(C) se realicen menos operaciones que al calcularlo como A(BC)?
- (a) $n < p$ (b) $p < n$ (c) $\frac{1}{n} + \frac{1}{q} < \frac{1}{m} + \frac{1}{p}$ (d) El número de operaciones no varía
85. (3) Maricarmen y Ana Cristina tienen un conjunto A de 2013 elementos. Primero Maricarmen cuenta todos los subconjuntos de A tales que su cardinalidad es impar, mientras que Ana Cristina aquéllos cuya cardinalidad es par. Al final ganó la que contó más subconjuntos. ¿Quién ganó?
86. (3) Encuentra una fórmula cerrada para calcular el número de cuadrados que se pueden formar utilizando los vértices y las aristas de una cuadrícula de $n \times n$. Argumenta tu respuesta.
87. (3) El Gogobol es un juego muy divertido que involucra cuatro pelotas, dos guitarras y un avestruz. Sin embargo, lo más interesante del Gogobol es que en cada partido el entrenador puede elegir cualquier número de sus jugadores para participar en el juego, siempre y cuando haya un capitán de equipo. La liga mexicana de Gogobol te ha encomendado elaborar una función que dado el número de miembros disponibles en un equipo de Gogobol, encuentre el número de distintas formas en que su entrenador puede elegir a los que van a participar en un juego. Como este número puede ser muy grande, tu función debe devolver dicho número módulo 1,000,000,007. Tu función debe cumplir con las siguientes especificaciones:
- (a) El prototipo o encabezado de la función debe ser `long long formasGogobol(int n)`.
- (b) n es el número de miembros disponibles para formar el equipo.
- (c) Tu función debe tener complejidad logarítmica con respecto al valor del entero n .

(d) El valor de retorno debe estar en su congruencia módulo 1,000,000,007.

(e) Ejemplos:

```
formasGogobol(1)=1
formasGogobol(2)=4
formasGogobol(3)=12
formasGogobol(2013)=427369524
```

88. (2) El Conde Primo, que es primo del Conde Contador de Plaza Sésamo, se puso a practicar el siguiente juego. La idea es que dados dos números primos de cuatro dígitos se busca llegar de uno al otro utilizando movimientos válidos. Un movimiento válido consiste en cambiar un único dígito del primo actual, de manera que el resultado siga siendo un número primo de cuatro cifras. El Conde Primo busca contratarte para que elabores una función que dados dos primos calcule el mínimo número de movimientos válidos para transformar uno en el otro. De esta manera él tendrá una forma de medir su desempeño en su loco juego. Tu función debe cumplir las siguientes especificaciones:

(a) El prototipo o encabezado debe ser `int numMovimientos(int p, int q)`.

(b) `p` y `q` son los primos para los cuales hay que calcular el mínimo número de movimientos para transformar uno en otro.

(c) Tienes garantizado que ambos números sí serán primos y que existe una secuencia de movimientos para llegar de primero al segundo.

(d) Ejemplos:

```
numMovimientos(1033, 8179)=6;
numMovimientos(1373, 8017)=7;
```

(e) Puedes utilizar las variables y funciones auxiliares que gustes.

89. (2) Leonardo McGato es un famoso inversionista que es conocido por su avaricia y tacañería. Lamentablemente te ha contratado para elaborar una función que debe encontrar el mínimo costo que puede obtener el sumar de dos en dos una lista de enteros, donde cada suma cuesta el resultado de la misma. Veamos cómo calcular el costo en un ejemplo, si suponemos que los números a sumar son 1, 2, 3 y 4:

Números por sumar	Sumandos	Lista resultante	Costo acumulado
1, 2, 3, 4	1+4=5	2, 3, 5	5
2, 3, 5	3+5=8	2, 8	5+8
2, 8	2+8=10	10	5+8+10

por lo que el costo de estas sumas es la suma de los costos de cada suma binaria: $5+8+10=23$. Este costo podría mejorarse, ya que no es óptimo. Podríamos haber elegido otras combinaciones:

Números por sumar	Sumandos	Lista resultante	Costo acumulado
1, 2, 3, 4	1+4=5	2, 3, 5	5
2, 3, 5	2+3=5	5, 5	5+5
5, 5	5+5=10	5+5+10	

por lo que este costo es 20, que es menor al anterior, pero todavía no es el mejor que podemos hacer. Lo podemos mejorar de la siguiente manera:

Números por sumar	Sumandos	Lista resultante	Costo acumulado
1, 2, 3, 4	1+2=3	3, 3, 4	3
3, 3, 4	3+3=6	4, 6	3+6
4, 6	4+6=10	3+6+10	

lo que nos da un costo de 19, que es ya el óptimo.

Tu función, que tiene que encontrar el óptimo para esta suma, debe cumplir las siguientes condiciones:

- El prototipo o encabezado debe ser `long long int encuentraCosteMin(int *a, int n)`.
- `a` es el arreglo que representa la lista de enteros.
- `n` es el tamaño de `a`.
- Ejemplos:

```
int x[4]={1, 2, 3, 4};
int y[3]={4,5,10};
int q=4;
encuentraCosteMin(x, q)=19
int q=3;
encuentraCosteMin(y, q)=9
```

90. (2) Tu nuevo jefe te ha encargado que revises el funcionamiento de un antiguo programa de la empresa. Tu tarea es, finalmente, determinar las estructuras de datos implementadas por dicho programa. Pero empezaremos con algo sencillo: tienes que diseñar un programa que sea capaz de analizar una serie de operaciones de tipo 1 o de tipo 2, donde el tipo 1 inserta un entero a una estructura de datos y el tipo 2 saca exitosamente un entero de ella. Tu programa debe determinar si se trata de una pila, una cola, una cola de prioridades (que regresa el entero mayor bajo la operación de tipo 2). No se puede saber cuál de los tres es o que no se trata de ninguno de los tres. Tu programa debe cumplir las siguientes condiciones:

- La entrada consiste de varios casos de prueba. Cada caso de prueba contiene una línea con un entero n , el número de operaciones a analizar, seguido de n líneas cada una representando una operación. El fin de la entrada está determinado por un EOF.
- La operación de insertar un entero, se representa por el entero 1 seguido del entero a insertar. La operación de sacar un entero, se representa por el entero 2 seguido del entero que regresa la estructura de datos misteriosa.
- Tu programa debe imprimir en la salida "cola", "cola de prioridades", "pila", "no se puede saber cual" o "ninguna de las tres" en una línea independiente de acuerdo al resultado del análisis.
- La prioridad de a es mayor que la de b (a va antes que b en la cola de prioridades) si $a > b$. En caso de igualdad, se decide arbitrariamente por cualquiera de los dos.
- Ejemplo de entrada:


```

6
1 1
1 2
1 3
2 3
2 2
2 1
6
1 1
1 2
1 3
2 1
2 2
2 3
2
1 1
2 2
7
1 2
1 5
1 1
1 3
2 5
1 4
2 4
4
1 2
1 1
2 1
2 2

```

(f) Ejemplo de salida:

```

no se puede saber cual
cola
ninguna de las tres
cola de prioridades
pila

```

91. (2) La Comisión de Control de Gatos de la Colonia Educación, CCGCE, ha utilizado tecnología satelital para clasificar y contar, por su raza, a todos los gatos que existen en dicha colonia. Tu tarea como empleado de la CCGCE es elaborar un programa que analice los datos recopilados y determine el porcentaje que cada raza representa en la población felina. Tu programa debe cumplir con las siguientes condiciones:
- La primera línea de la entrada es un entero n representando el número de casos de prueba, seguido de una línea en blanco. El fin de la entrada se determina por un EOF.
 - Cada caso de prueba consiste de una lista de las razas de los gatos observados por el satélite, la especie de cada gato en una línea. Ningún nombre de raza excede los 33 caracteres ni

contiene espacios; no hay más de 10,000 especies ni más de 1,000,000 de gatos. Hay una línea en blanco separando cada caso de prueba consecutivo.

- (c) Para cada caso de prueba imprime, en orden alfabético, el nombre de cada raza representada en la población de gatos seguido del porcentaje de la población que representa, con precisión a cuatro decimales. Imprime una línea en blanco entre dos casos de prueba consecutivos.
- (d) Ejemplo de entrada:

```
2

Bombay
Egipcio
Mixto
Mixto
Bombay
MaineCoon
GatoDeLosBosquesNegros

Bombay
Mixto
Mixto
MaineCoon
MaineCoon
```

- (e) Ejemplo de salida:

```
Bombay 28.5714
Egipcio 14.2857
GatoDeLosBosquesNegros 14.2857
MaineCoon 14.2857
Mixto 28.5714

Bombay 20.0000
MaineCoon 40.0000
Mixto 40.0000
```

92. (7) Regresemos a nuestros orígenes. Imaginemos que tenemos el siguiente pseudocódigo estilo Pascal:

```
1 procedure S(c, d: integer);
2 begin
3   c:=c-d;
4   d:=c+d;
5   c:=d-c;
6 end
```

Si ambos parámetros de S son pasados por referencia ¿Qué función tiene dicho procedimiento? Si $a = 13$ y $b = 20$ son variables enteras. ¿Cuál es su valor después de ser llamado $S(a,b)$? Y si las variables son pasadas por valor, ¿cuál es su valor después de ser llamado $S(a,b)$?

93. (7) Regresemos a nuestros orígenes, imaginando que tenemos el siguiente pseudocódigo estilo Pascal:

```
1 procedure S(c, d: integer);  
2 begin  
3   c:=c ^ d;  
4   d:=c ^ d;  
5   c:=c ^ d;  
6 end
```

Si ambos parámetros de S son pasados por valor, $a = 20$ y $b = 13$ son variables enteras y \wedge es un operador XOR. ¿Cuál es su valor después de ser llamado $S(a,b)$? Si los parámetros de S fueran pasados por referencia, ¿qué función tendría S?

94. (7) Poner comentarios en tu código sirve para algo. Si no nos crees, observa el siguiente problema. Supongamos que tenemos el siguiente código en C/C++, ¿qué papel realiza la función s? y ¿qué complejidad tiene con respecto a los valores de a y b ?

```
1 void s(int a, int &b) {  
2   int pot=1;  
3   while(a/pot || b/pot) {  
4     a+=((((a/pot)%2 + (b/pot)%2)%2 - (a/pot)%2)%2)*pot;  
5     pot*=2;  
6   }  
7   pot=1;  
8   while(a/pot || b/pot) {  
9     b+=((((a/pot)%2 + (b/pot)%2)%2 - (b/pot)%2)%2)*pot;  
10    pot*=2;  
11  }  
12  pot=1;  
13  while(a/pot || b/pot) {  
14    a+=((((a/pot)%2 + (b/pot)%2)%2 - (a/pot)%2)%2)*pot;  
15    pot*=2;  
16  }  
17 }
```

95. (1) En la cafetería de una prisión administrada por un capataz loco, los prisioneros se forman en una fila que es administrada con una política Q: los prisioneros se pueden formar las veces que quieran en la fila y en el instante que se les antoje. ¿Cuál de las siguientes políticas garantiza que ningún prisionero se morirá de "hambre"?

(a) *Round Robin* (b) *Shortest Job First* (c) *Youngest Job First* (d) Ninguna de las anteriores

96. (1) Siempre es bueno saber cómo se guardan ciertos tipos de datos en la computadora. Si sabemos que en C los tipo `float` son análogos al formato IEEE de precisión sencilla, ¿cuántas líneas imprime el siguiente programa?

```
1 #include <stdio.h>  
2 float sum=0.0;  
3 float a=2.0;  
4 float b=1.0;
```

```

5 int main() {
6     while(a/b>0) {
7         b*=2;
8         sum+= a/b;
9         printf("%f\n" , sum);
10    }
11    return 0;
12 }

```

97. (1) Siempre es bueno saber cómo se guardan ciertos tipos de datos en la computadora. Si sabemos que en C los tipo float son análogos al formato IEEE de precisión sencilla, ¿cuál es el entero más cercano al último valor de sum que se calcule en el siguiente programa?

```

1 #include <stdio.h>
2 float sum=0.0;
3 float a=2.0;
4 float b=1.0;
5 int main() {
6     while(a/b>0) {
7         b*=2;
8         sum+= a/b;
9         printf("%f\n" , sum);
10    }
11    return 0;
12 }

```

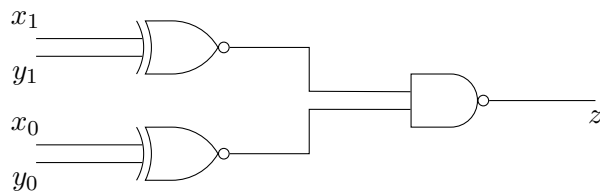
98. (3) Si tenemos los siguientes conjuntos de números:

- (a) $C_1 = \{-8, -4, -2, -1, 0, 1, 2, 4, 8\}$
- (b) $C_2 = \{-8, -4, -2, -1, 1, 2, 4, 8\}$
- (c) $C_3 = \{1\}$
- (d) $C_4 = \{-6, -4, -3, -2, -1, 1, 2, 4, 6\}$
- (e) $C_5 = \{-1, 1\}$
- (f) $C_6 = \{1, 2, 3, 6\}$

¿Cuáles de los C_i 's pueden ser conjuntos que describan a los divisores comunes enteros de dos enteros positivos?

99. (6) El nuevo juego de "El Automata Determinista Finito Misterioso" ya se encuentra a la venta en varias librerías. En la tapa del empaque viene una pregunta ejemplo que dice así: "Este autómata reconoce al conjunto $S \in \{a, b, c\}^*$ donde $x \in S$ si y sólo si el número de letras a en x es par, el número de letras b es impar y el número de letras c es divisible por k . En términos de k ¿Cuál es el mínimo número de estados que tiene nuestro autómata? Justifica tu respuesta.

100. (1) En cierto sistema se cuenta con una memoria virtual de m páginas residentes y una cadena de referencia de n peticiones de página $C = q_1, q_2, q_3, \dots, q_n$. Supongamos que las n peticiones fueron construidas de manera aleatoria y ninguna de dichas páginas fuera residente al inicio. Si repitiéramos C cuatro veces utilizando FIFO, en términos de n ¿cuál es el máximo valor que puede tomar m y provocar $4n$ fallos de página? Justifica tu respuesta.
101. (1) En cierto sistema se cuenta con una memoria virtual de m páginas residentes y una cadena de referencia de n peticiones de página $C = q_1, q_2, q_3, \dots, q_n$. Suponiendo que las n peticiones fueron construidas de manera aleatoria y ninguna de dichas páginas fuera residente al inicio, Pancho afirmó las siguientes proposiciones:
- (a) Si C se repite muchas veces seguidas y tenemos $m < n$, entonces entre más chica se vuelve la m ocurren más fallos de página.
 - (b) Si C se repite muchas veces seguidas, el menor número de fallos de página ocurre cuando $m \geq n$.
 - (c) Si C se repite cualquier número de veces seguidas, LRU y FIFO generan el mismo número de páginas fallidas.
- ¿Cuántas mentiras dijo Pancho? ¿Cuáles fueron? Justifica tu respuesta.
102. (6) Imaginemos que tenemos un lenguaje L libre de contexto y un lenguaje regular R ¿Será cierto que $L \cap R$ es libre de contexto? Justifica tu respuesta.
103. (6) Imaginemos que tenemos dos lenguaje L_0 y L_1 libres de contexto ¿Será cierto que $L_0 \cap L_1$ es libre de contexto? Justifica tu respuesta.
104. (6) Sabemos que si R_0 y R_1 son ambos lenguajes regulares entonces $R_0 \cap R_1$ también es regular. ¿Es $R_0 \setminus R_1$ regular? Justifica tu respuesta.
105. (1) Sean X e Y dos enteros sin signo de dos bits cada uno. El siguiente diagrama misterioso sirve para determinar si $X ? Y$. ¿Qué operador debería remplazar a “?” para que la operación especificada produzca siempre $Z = 1$?



106. (1) Un bloque de 2013 palabras de memoria es usado como almacenamiento dinámico para objetos de tamaño 3 palabras y tamaño 13 palabras. Se disponen de las siguientes operaciones:
- (a) alojar (n): Aloja un objeto de tamaño n en la memoria ($n=3$ o $n=13$). Regresa la dirección inicial de la memoria asignada.
 - (b) libera (y): Libera la memoria asignada al objeto cuya dirección inicial es y .

Supongamos que en cierto punto la operación alojar causa un error, debido a que no se dispone de un bloque de memoria contigua lo suficientemente grande. ¿Cuál es el mínimo número de palabras que pueden estar realmente en uso en la memoria?

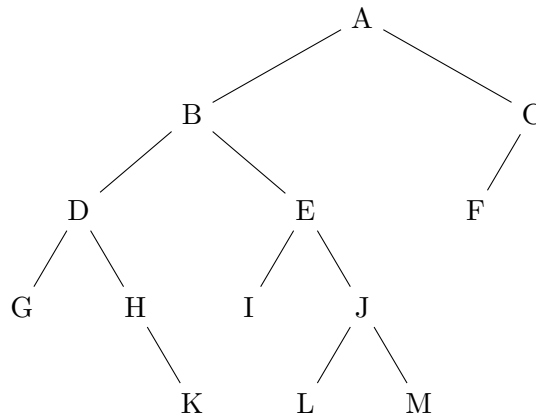
107. (5) Supongamos que $Q(x, y)$ significa "x es padre de y" y $H(x)$ significa "x es hombre". Si la expresión $G(x, y)$ es equivalente a:

$$\neg H(x) \wedge \exists v \exists w \exists z (Q(z, w) \wedge Q(z, v) \wedge (w \neq v) \wedge Q(w, x) \wedge Q(v, y))$$

Rellena la oración "Podemos decir que x es _____ de y".

108. (2) Había una vez en el mundo de las gráficas simples un árbol llamado Shady. Shady era muy tímido y no le gustaba mostrarse en las fiestas, reuniones, eventos deportivos, etcétera. Por esta razón, nadie sabía cómo era Shady físicamente, aunque es un hecho conocido que cuando una gráfica cumple 30 años combinatorios tiene que ir al doctor a realizarse unos recorridos. Es así que cuando Shady cumplió sus 30 años se realizó un recorrido inorden y uno preorden. Si sabemos que la lista $N = G, D, H, K, B, I, E$ representa el recorrido inorden de Shady y la lista $P = B, D, G, H, K, E, I$ representa el recorrido preorden de Shady, ¿serás capaz de dibujar a Shady? En caso de ser posible dibújalo y, en caso contrario, presenta tus argumentos.

109. (2) Si tenemos el siguiente árbol y sabemos que es un árbol de búsqueda binaria, ¿cuál es el nodo más chico?, ¿el cuarto más chico?, ¿y el noveno?



110. (5) Una implementación básica de la lógica difusa es utilizar valores lógicos 0,1 o 2, cada uno representando la noción de verdadero, falso y desconocido. De esta manera, ¿cuántas variables necesitaríamos para representar al menos 2013 valores diferentes?

111. (1) En los protocolos IP, ¿cuál de las siguientes opciones describe el propósito del Protocolo de Resolución de Direcciones?

- (a) Traducir direcciones Web a nombres de anfitrión.
- (b) Determinar la dirección IP del nombre de un anfitrión dado.

- (c) Determinar la ruta adecuada para un datagrama.
- (d) Determinar la dirección de hardware de una dirección IP dada.

112. (5) En la casa de la lógica de los predicados vivían tres predicados:

- (a) $(\forall xP(x) \vee \forall xQ(x)) \Rightarrow \forall x(P(x) \vee Q(x))$
- (b) $(\forall x(P(x) \vee Q(x)) \Rightarrow \forall xP(x) \vee \forall xQ(x))$
- (c) $(\exists xP(x) \vee \exists xQ(x)) \Rightarrow \exists x(P(x) \vee Q(x))$

¿Cuántas tautologías vivían en la casa de la lógica de los predicados? ¿Cuántos predicados satisficibles vivían en la casa de la lógica de los predicados?

113. (7) Supongamos que el siguiente programa en Scheme utiliza alcance estático o léxico para el manejo de variables.

```

1 (define m 5)
2 (define n 10)
3 (define (p1)
4   (display (list "En p1, n vale " n))
5   (newline))
6 (define (p2)
7   (display (list "En p2, m vale " m))
8   (newline)
9   (display (list "En p2, n vale " n))
10  (newline)
11  (p1))
12 (define (list "En el main, n vale " n))
13 (newline)
14 (p2 1)
15 (p1)

```

La salida de dicho programa toma la siguiente forma:

```

En el main, n vale ?
En p2, m vale ?
En p2, n vale ?
En p1, n vale ?
En p1, n vale ?

```

Completa los ? con los valores que deben de tomar en cada línea.

114. (7) Supongamos que el siguiente programa en Scheme se utiliza alcance dinámico para el manejo de variables.

```

1 (define m 5)
2 (define n 10)
3 (define (p1)
4   (display (list "En p1, n vale " n))
5   (newline))

```

```

6 (define (p2)
7   (display (list "En p2, m vale " m))
8   (newline)
9   (display (list "En p2, n vale " n))
10  (newline)
11  (p1))
12 (define (list "En el main, n vale " n))
13 (newline)
14 (p2 1)
15 (p1)

```

La salida de dicho programa toma la siguiente forma:

```

En el main, n vale ?
En p2, m vale ?
En p2, n vale ?
En p1, n vale ?
En p1, n vale ?

```

Completa los ? con los valores que deben de tomar en cada línea.

115. (1) ¿Cuáles de las siguientes afirmaciones, que se refieren a datagramas enviados en un nodo dentro de una red que utiliza el protocolo IPv4, son ciertas?
- (a) Los datagramas enviados desde el origen deben tener un tamaño menor o igual a la máxima unidad de transmisión de todos los enlaces que forman la ruta hacia el destino.
 - (b) Los datagramas pueden ser fragmentados durante el enrutamiento.
 - (c) Los datagramas sólo pueden ser fragmentados en el origen de la ruta.
 - (d) Los datagramas sólo se reensamblan al llegar al destino.
116. (5) Un circuito lógico tiene como entrada un número de tres bits: b_0 , b_1 y b_2 , donde b_0 es el bit menos significativo y b_2 es el bit más significativo. La salida de dicho circuito es 1 cuando las entradas son los números 1, 3, 5 y 6. Da una expresión booleana que represente la salida de dicho circuito.
117. (5) Sea Q el siguiente problema: Dado un programa y una entrada para el mismo, determinar si la salida es la expansión decimal de π . ¿Es Q decidible? Argumenta tu respuesta.
118. (2) En unas ruinas antiguas fue encontrado el siguiente pseudocódigo:

```

1 f(k) {
2   x=3;
3   for i=1 to k
4     x=x*x;
5   return x;
6 }

```


Tu trabajo como arqueoprogramador es determinar cuál de los siguientes valores para k es el mínimo que cumple que, dada una n entera y positiva, se tiene que $f(k) \geq n$.

- (a) $\lceil \log_2(\log_3(n)) \rceil$
- (b) $\lceil \log_3(\log_2(n)) \rceil$
- (c) $\lfloor \log_2(\log_3(n)) \rfloor$
- (d) $\lfloor \log_3(\log_2(n)) \rfloor$

119. (2) Supongamos que tenemos un árbol binario balanceado que cumple la propiedad de un *heap-max* y cuyo recorrido en preorden es $Q=[10,8,3,7,5,2,4]$. ¿Cuál sería la lista asociada a su recorrido preorden después de haber realizado una operación delete-max?
120. (1) Una instrucción privilegiada puede ser ejecutada únicamente mientras el hardware se encuentra en modo kernel. ¿Cuál de las siguientes instrucciones tiene menos probabilidad de ser una instrucción privilegiada?
- (a) Enviar una salida a la impresora.
 - (b) Detener el CPU
 - (c) Aumentar el valor del PC (“*program counter*” o contador del programa).
121. (1) Si en general sólo existe un registro que contiene el PC (“*program counter*” o contador del programa), ¿qué funcionalidad o concepto hace posible que dos procesos se ejecuten con dos valores del PC completamente distintos de manera no secuencial en una computadora con un único procesador?
122. (1) En el pueblo Paralelotongo existían dos hermanos Pancho y Mariana. Pancho y Mariana eran muy muy parecidos, pero una de las diferencias más grandes entre ellos era que mientras Pancho no compartía sus cuadernos con otros niños, Mariana compartía sus cuadernos con otras niñas de manera muy frecuente. Determina los valores X e Y de la siguiente analogía: Cuadernos es a memoria, lo que Pancho es a X y Mariana es a Y .
123. (5) El Profesor Y ha diseñado una máquina parabolimetapositrónica, la cual él garantiza es capaz de determinar en tiempo polinomial si una máquina de Turing se detiene sin importar el contenido de su cinta de entrada. ¿Será cierto lo que el Profesor Y garantiza?
124. (1) Consideremos la siguiente cadena de referencia de páginas: 1, 2, 3, 4, 5, 2, 1, 5, 7, 8, 9, 10, 1, 2, 3, 4, 5.
Suponiendo que hay 5 marcos de página disponibles y que al inicio se encuentran vacíos ¿Cuántos fallos de página tendrían lugar si utilizamos LRU como protocolo de remplazo?
125. (1) Consideremos la siguiente cadena de referencia de páginas: 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7. Si sólo tenemos 6 marcos de página disponibles, que al inicio se encuentran vacíos, y utilizamos LRU, ¿cuántos fallos de página tienen lugar? Más importante, ¿por qué esto pasa muy raramente en los sistemas operativos reales?

126. (1) Jorge es el encargado de organizar un concurso de batallas pokemón, pero no cuenta con muchas habilidades en el diseño web. Irónicamente, una de sus responsabilidades es mantener la página web de dicho concurso, en la cual sube los resultados de las diferentes batallas que tienen lugar en los torneos. Leonardo es el fan número uno de dicho concurso en la Facultad de Ingeniería, por lo que mínimo consulta dicha página 15 veces al día. Además, el concurso es seguido por miles de espectadores en el Distrito Federal. Un día Jorge quedó de subir ciertos resultados a las 5 de la tarde. Leonardo naturalmente revisó la página a dicha hora y no encontró los resultados, por lo que rápidamente telefoneó a Jorge para reclamar, a lo que Jorge respondió que ya había subido los resultados para cuando Leonardo revisó la página. Leonardo inmediatamente pulsó el botón de actualizar del explorador y logró visualizar los resultados de las batallas. ¿Qué pudo haber salido mal la primera vez que Leonardo intentó acceder a la página web?
127. (1) Existen dos maneras de reconocer a un servidor, por su *hostname* o por su dirección IP. Las personas prefieren el primero mientras que los ruteadores prefieren el segundo. La principal tarea del DNS es traducir unas en otras. En la actualidad se utilizan muchos servidores DNS implementados en jerarquías para resolver estas peticiones. ¿Por qué no se utiliza un diseño más simple donde sólo se utilizara un servidor en vez de cientos distribuidos por el mundo? Proporciona dos razones.
128. (1) Supongamos que un equipo tiene que enviar solamente cinco paquetes distintos 1,2,3,4 y 5 y que su ventana de envío es lo suficientemente grande para haber enviado todos los paquetes a la vez. Antes de que su temporizador se termine, dicho equipo recibe una señal ACK(2) y posteriormente una señal ACK(4) del equipo receptor. Más tarde el temporizador llega a su fin. Si se está utilizando un protocolo *Go Back-N*, ¿cuántos paquetes tiene que reenviar dicho equipo?
129. (1) En los protocolos de transferencias confiables *Selective Repeat* y *Go Back N* por ejemplo, se utiliza una ventana de tamaño m para manejar el envío y recepción de paquetes y se numeran los paquetes a enviar utilizando n números distintos del 0 al $n - 1$. ¿Cómo tiene que ser n respecto a m para que no haya ambigüedades con el manejo de la ventana?
130. (4) Supongamos que tenemos la siguiente tabla que representa el ejemplar (*instance*) de cierta relación $R(A,B,C)$ en una base de datos.

A	B	C
1	a	2
1	a	3
1	b	4
1	c	5

¿Cuál(es) de las siguientes dependencias funcionales **no** puede(n) formar parte de las asociadas a R ?

- (a) $C \rightarrow A$ (b) $B \rightarrow A$ (c) $A \rightarrow C$ (d) $A \rightarrow B$

131. (2) Cierta satélite ha recopilado la información geológica de una zona cuadrada, previamente cuadrículada en parcelas, planeada para minar piedras preciosas. Dicha información se encuentra representada en un arreglo bidimensional de ceros (0) y unos (1), donde los ceros y unos representan la inexistencia o existencia, respectivamente, de dichas piedras preciosas en esa parcela. La compañía minera maneja el siguiente sistema de costos: abrir una mina en una parcela cuesta 10 mil dólares, mientras que expandir la mina de una parcela a alguna parcela adyacente (que comparta un vértice) cuesta 9 mil dólares. La compañía minera planea minar todas las parcelas donde existan piedras preciosas gastando la menor cantidad de dinero posible. Tu tarea es elaborar un programa que determine el número de minas a abrir. Tu programa debe cumplir las siguientes condiciones:

- (a) La entrada consiste de varios casos de prueba. Cada caso de prueba contiene una línea con un entero $0 \leq n \leq 25$, que representa que la zona ha sido cuadrículada en n^2 parcelas.
- (b) El caso de prueba donde $n = 0$ representa el final de la entrada.
- (c) Las siguientes n líneas contienen, cada una, una cadena de ceros y unos de longitud n representado la información recopilada por el satélite.
- (d) La salida de cada caso de prueba debe mostrarse en una línea independiente con el formato mostrado a continuación.
- (e) Ejemplo de entrada:

```

6
100100
001010
000000
110000
111000
010100
8
01100101
01000001
00011000
00000010
11000011
10100010
10000001
01100000

```

- (f) Ejemplo de salida:

```

Para el caso de prueba 1 se necesitan 3 minas.
Para el caso de prueba 2 se necesitan 6 minas.

```

132. (7) Existe una pregunta muy famosa para los *testers* que desean entrar a trabajar a la empresa Scopeta: Si tenemos los siguientes códigos en C:

Programa 1

```

1 int suc ();
2 int main () {
3     int n;
4     int k;
5     n=10;
6     k=suc ();
7     return 0;
8 }

```

Programa 1 (continúa)

```

9 int suc () {
10     return n+1;
11 }
16

```

Programa 2

```

1 int suc ();
2 int n;
3 int k;
4 int main () {
5     n=10;
6     k=suc ();
7     return 0;
8 }

```

Programa 2 (continúa)

```

9 int suc () {
10     return n+1;
11 }
16

```

Les preguntan a los candidatos cuál o cuáles de estos códigos son rechazados por el compilador y por qué razón. ¿Cuál debe ser la respuesta para ser contratados?

133. (2) El Conde Contador te ha contratado para que elabores un programa que le ayude a calmar sus obsesiones. Últimamente se ha preguntado de cuántas formas distintas puede expresarse un entero no negativo n como suma de 5 enteros ordenados no negativos. Por ejemplo, el 1 puede expresarse de 5 formas distintas: $1 + 0 + 0 + 0 + 0$, $0 + 1 + 0 + 0 + 0$, $0 + 0 + 1 + 0 + 0$, $0 + 0 + 0 + 1 + 0$ y $0 + 0 + 0 + 0 + 1$. Después de un sin fin de cuentas el conde, logró descubrir que existen 687,574,403,880 maneras distintas de expresar al 2013 como suma de cinco enteros ordenados no negativos. Al lograr dicho resultado no le fue muy difícil calcular todas las maneras distintas para expresar a $n < 2013$ como suma de cinco enteros no-negativos ordenados. Lamentablemente dichos cálculos fueron robados y el conde ha perdido la cabeza ante la idea de volver a realizar tantos cálculos. Tu tarea es simple, tienes que implementar una función bajo el prototipo mostrado a continuación que sea capaz de calcular las maneras distintas de expresar a $n < 2013$ como suma de cinco enteros no negativos ordenados.

```
type cuenta(int n)
```

- (a) `type` representa el tipo del número que vas a regresar: escoge adecuadamente.
- (b) `n` representa el número del cual hay que calcular las maneras de expresarlo como suma de cinco enteros no negativos ordenados.
134. (2) En el mundo de Grafocea se está realizando un censo para contar la cantidad de árboles que existen en cierta ciudad. Por ello te han contratado para que implementes una función en C/C++ que reciba la lista de aristas (representada por un arreglo) y el número de vértices que describen a cierta gráfica simple, y determines si se trata de un árbol o no. Los n vértices de una gráfica simple que vive en Grafocea se representan con los enteros del 0 al $n - 1$ y toda

gráfica tiene menos de 2013 vértices. Puedes suponer que en la lista de aristas no hay dos representaciones de la misma arista. Tu función tiene que cumplir con el siguiente prototipo y trabajar con la siguiente estructura:

```

1 typedef struct aristaS {
2     int x; //Vertice 1
3     int y; //Vertice 2
4 } arista;
5 bool esArbol(arista *lista , int tamlista , int numvertices);

```

donde

- lista es un apuntador al arreglo de aristas.
- tamlista es el tamaño de la lista de aristas.
- numvertices es el número de vértices que contiene la gráfica.

Puedes hacer uso de las funciones auxiliares y estructuras/tipos que tú gustes.

135. (2) Los pingüinos son animales fascinantes. Cuando encuentran una pareja suelen quedarse con ella de por vida. Fernando el pingüino es un pingüino con muchos complejos. Uno de ellos es que no quiere que su pareja sea más alta que él, pero tampoco quiere acabar solo, por lo que si todas las pingüinas son más altas que él, está dispuesto a hacer una excepción. Otro de sus complejos es que siempre ha soñado con tener una pareja lo menos chaparra posible. Fernando te ha pagado 10 pescados frescos para que elabores una función en C/C++ que dado un arreglo de tamaño n , con las estaturas de todas las pingüinas de su isla, ordenadas de menor a mayor, devuelva la estatura de la pareja ideal de Fernando. Dicha función necesita ejecutarse en un orden de $\log n$ por las exigencias computacionales que se presentan en la Antártica. Tu función tiene que cumplir con el siguiente prototipo:

```
float estaturaIdeal(float *lista , int tamlista , float estaturaFer);
```

donde:

- lista es un apuntador al arreglo de las estaturas de las pingüinas.
- tamlista es el tamaño del arreglo anterior.
- estaturaFer representa la estatura del pingüino Fernando.

136. (2) Todos hemos oído hablar del problema de las 8 reinas. Dicho problema consiste en colocar 8 reinas sobre de un tablero de ajedrez de manera que ninguna ataque a otra. Tu tarea es un poco más complicada, pues debes elaborar una función en Haskell que resuelva dicho problema, pero sustituyendo el 8 por una $0 < n \leq 20$ arbitraria, es decir, que si existe un acomodo que cumpla dichas restricciones, especifique las posiciones de al menos un acomodo. Tu función tiene que cumplir con el siguiente prototipo y trabajar con la siguiente estructura:

```

type Queen = (Int , Int)
pReinas :: Int -> Maybe[Queen]

```

donde:

- Queen Es el tipo que representa la posición de una reina por sus coordenadas.
- pReinas Recibe un entero que representa el tamaño del tablero y devuelve una lista de tipos Queen si existe solución o Nothing en caso contrario.

Puedes elaborar las funciones y estructuras auxiliares que gustes.

137. (2) Una de las grandes ventajas de Haskell, y otros lenguajes funcionales, es que acepta funciones como parámetros de una función. Para familiarizarte con esto Funcio te ha pedido que elabores una función que reciba un árbol, no necesariamente binario, que devuelva el mismo árbol pero con la función aplicada a sus nodos. Tu función tiene que cumplir con el siguiente prototipo y trabajar con la siguiente estructura:

```
data ArbolN a = VoidN | NodoN a [ArbolN a] deriving (Show, Eq)
aplicaArbolN :: Eq a => (a->b)->ArbolN a->ArbolN b
```

138. (2) Estudiar cualquier cosa relacionada con la ciencia puede llegar a ser fatal para las mentes de algunos alumnos. Uno de dichos alumnos enloquecidos te ha pedido que le ayudes con un problema. Este alumno siempre ha sufrido de claustrofobia aguda, por lo que cuando maneja cadenas en Java inserta cuatro espacios entre cada par de palabras consecutivas. El problema fue que él se encargó de declarar un arreglo de caracteres con espacio suficiente para realizar esto, pero al momento de instanciarlo su compañero normal lo rellenó con sólo un espacio entre cada par de palabras consecutivas. Tu tarea es desarrollar un método que reciba una cadena de palabras y la longitud que tendría para una persona normal, y debe sustituir cada espacio en dicha cadena por cuatro espacios consecutivos. Dicha sustitución se debe realizar sin ocupar nada de espacio adicional (*in place*). Tu método tiene que cumplir con el siguiente prototipo:

```
public static void remplazaEspacios(char[] str, int longitud);
```

donde

- `str` es la cadena a corregir.
- `longitud` es la longitud que tendría la cadena para una persona normal.

139. (2) El pueblo Tableroix, como su nombre lo dice, está compuesto por muchas villas distribuidas en forma de tablero. Actualmente empezó a correr una epidemia que es muy contagiosa: Cuando pasa un mes, toda villa contagiada instantáneamente contagia a todas las villas de la misma fila y de la misma columna a las que pertenece. El centro de salud de Tableroix te ha pedido que elabores un método en Java que reciba una matriz de enteros representando a Tableroix, donde un 1 representa que dicha villa se encuentra infectada y un 0 lo contrario. Tu método debe modificar dicha matriz para que represente como se verá Tableroix a un mes de hoy. Tu método debe tener el siguiente encabezado:

```
public static void contagia(int[][] pueblo)
```

donde

- `pueblo` la matriz de 1's y 0's que representa la situación actual de Tableroix.

140. (2) Gala y Vinka vivían en la cabeza de una lista ligada. Un día decidieron correr hacia adelante para explorar su mundo, pero Vinka era más rápida que Gala y avanzaba dos nodos en lo que Gala avanzaba uno. Misteriosamente, después de un tiempo, Vinka volvió a alcanzar a Gala, lo que significaba que vivían en una lista ligada que contenía un ciclo. Tu tarea es elaborar un método en Java que reciba la cabeza de la lista ligada en donde Gala y Vinka vivían, el nodo donde se encontraron y devuelva el primer nodo de la lista ligada que pertenece al ciclo que contiene. Tu método debe tener el siguiente encabezado:

```
LinkedListNode encuentraInicioCiclo(LinkedListNode cabeza ,
                                     LinkedListNode encuentro)
```

donde

- `cabeza` es el nodo cabeza de la lista ligada donde vivían Vinka y Gala.
- `encuentro` es el nodo donde se encontraron a causa del ciclo.

141. (2) Para cierto juego llamado "Scroggle", si se tienen dos palabras q_1 y q_2 que utilizan únicamente letras minúsculas, es de muchísima utilidad saber cuál es la cadena x de mayor longitud tal que existe una permutación de x que sea subsecuencia de q_1 y que existe una permutación (no necesariamente la misma) de x que sea subsecuencia de q_2 . Tú has decidido volverte un jugador profesional de "Scroggle" por lo que te has propuesto desarrollar una función en C/C++ que determine la longitud de dicha cadena x . Tu función recibirá un apuntador al arreglo de caracteres que representa a q_1 con su longitud y un apuntador al arreglo de caracteres que representa a q_2 con su longitud. Debe devolver la longitud de la máxima cadena x que cumple lo anteriormente dicho. Tu función debe cumplir el siguiente prototipo:

```
int scroggle(char *a, int longa, char *b, int longb)
```

142. (3) Una hormiga se encuentra parada en la esquina inferior izquierda de un tablero de $m \times n$ con $1 \leq m, n \leq 10$. Si solamente se puede mover hacia la derecha o hacia arriba y siempre caminando por las orillas de las casillas de dicho tablero, ¿cuántas maneras distintas tiene de hacerlo? Elabora un método en Java que reciba las longitudes del tablero y regrese un entero con la respuesta a dicha pregunta. Tu método debe tener el siguiente encabezado:

```
public int cuentaCaminos(int m, int n)
```

143. (3) Una compañía de construcción muchas veces se queda con vigas de madera de sobra, las cuales son utilizadas para formar triángulos que soporten estructuras. Pero para ello es necesario saber qué tipo de triángulo formarían. Tu trabajo en la solución de este problema es implementar un método en Java que reciba tres enteros, representando las longitudes de las vigas, y devuelva 0 si el triángulo que se forma es degenerado, 1 si es acutángulo, 2 si es rectángulo y 3 si es obtusángulo. Tu método debe tener el siguiente encabezado:

```
public int determinaTriangulo(int a, int b, int c)
```

144. (2) Para cierto juego de cartas muchas veces es bueno saber si dada una mano de n cartas ordenadas es posible sumar cierto entero s . Te has propuesto diseñar una función en C/C++ que reciba un arreglo (apuntador) de enteros, ordenado de menor a mayor, su longitud y un entero s . Dicha función debe regresar *true* si existe una pareja de enteros dentro del arreglo cuya suma sea igual a s y *false* en caso contrario. Tu función debe cumplir el siguiente prototipo y ejecutarse en tiempo lineal con respecto a n :

```
bool posibleSuma(int *a, int n, int s);
```

donde

- `a` Es el arreglo ordenado de menor a mayor.
- `n` Es el tamaño del arreglo.
- `s` Es el entero que queremos sea la suma de dos enteros del arreglo.

145. (2) Un arreglo a de tamaño n es k ordenado si para todo $i < n - k$ se cumple que $a_i \leq a_{i+k}$. Tu objetivo es desarrollar una función que dado un arreglo k ordenado y su longitud, ordene el arreglo de menor a mayor. Tu función debe cumplir el siguiente prototipo y ejecutarse en tiempo $O(n \log k)$:

```
void ordenaK(int *a, int n, int k);
```

donde

- a Es el arreglo k ordenado de menor a mayor.
- n Es el tamaño del arreglo.

Puedes implementar las funciones auxiliares y estructuras/tipos que gustes.

146. (2) Un arreglo a de tamaño nk es k ordenado estricto si para todo $i < n - k$ se cumple que $a_i < a_{i+k}$. ¿Cuál es la máxima distancia que puede haber entre la posición actual de un elemento del arreglo y su posición en el mismo arreglo ordenado de menor a mayor de manera estándar? Da una respuesta argumentada en función de n y k .

147. (3) Una rana está parada en el 0 de la recta numérica. Cada día la rana da un salto de longitud " k " a la derecha, donde k es un entero positivo que permanece fijo durante el transcurso del tiempo pero que nosotros no conocemos. Cada día tenemos permitido verificar un número de la recta numérica para ver si la rana se encuentra ahí. ¿Es posible crear un algoritmo que garantice eventualmente adivinar(algún día) el número en el que se encuentra la rana?

148. (3) A un tablero de 2013×2013 se le han quitado la casilla superior derecha, la superior izquierda y la inferior derecha. Demuestra que no es posible llenarlo con fichas de 3×1 de manera que ninguna se salga del tablero y no se encimen entre ellas. Las fichas se pueden voltear y rotar.

149. (3) ¿Cuántos primos q cumplen que $2^{2^n} = q + 1$ con n un entero positivo? Argumenta tu respuesta, es decir, demuestra que son exactamente esos.

150. (3) Chiu tiene el siguiente tablero de 6×6 , existen 3 posibles movimientos para Chiu. Él puede elegir entre cualquier fila, cualquier columna y cualquier subtablero de 2×2 del tablero y sumar o restar un entero cualquiera a todos los elementos (a todos sumarle k o a todos restarles k) que pertenezcan a la fila, columna o subtablero de 2×2 elegido. Utilizando solamente estos movimientos, ¿será posible que, eventualmente, Chiu logre convertir todos los números del tablero al número 2013?

1	2	3	4	5	6
12	11	10	9	8	7
13	14	15	16	17	18
24	23	22	21	20	19
25	26	27	28	29	30
36	35	34	33	32	31

Parte III
Bibliografía

Bibliografía

- [1] Andreescu, T., Andrica, D., Feng, Z., *104 Number Theory Problems*, Birkhäuser Boston, 2007.
- [2] Bondy, A., *Graph Theory*, Springer, 2008.
- [3] Cormen, T. H., *Introduction to algorithms*, MIT Press, 2001.
- [4] Engel, A., *Problem-solving strategies*, Springer-Verlag, 1998.
- [5] Halim, S., Halim, F., *Competitive Programming 3*, lulu, 2013.
- [6] Kurose, J.F., Ross, K.W., *Computer Networking: A Top-Down Approach*, Pearson, 2012.
- [7] Russel, S., Norvig, P., *Inteligencia Artificial: Un Enfoque Moderno*, Pearson, 2004.
- [8] Sedgewick, R., Wayne, K., *Algorithms*, Addison-Wesley Professional, 2011.
- [9] Silberschatz, A., Sudarshan, S., Korth, H.F., *Fundamentos de bases de datos*, McGraw-Hill, 2006.
- [10] Silberschatz, A., Galvin, P.B., Gagne, G., *Operating system concepts*, Wiley, 2012.
- [11] Tucker, A., Noonan, R., *Programming Languages*, McGraw-Hill, 2006.
- [12] Viso Gurovich, E., *Introducción a la teoría de la computación (autómatas y lenguajes formales)*, Facultad de Ciencias, 2008.