V Olimpiada de Informática del estado de Guanajuato
Solución del examen Teórico

I.- ¿Qué número debe seguir según la sucesión?

1) 7, 11, 13, 17, 19, 23, 29, ¿?
(*5 puntos*)

Respuesta:

Todos son números primos consecutivos, por lo que sin duda, el siguiente se trata del 31, que es el siguiente primo después del 29.

2) 1978, 1982, 1986, 1990, ¿?
(*5 puntos*)

Respuesta:

La diferencia entre el primer número y el segundo es 4, al igual que la diferencia entre el segundo y tercero y del tercero con el cuarto. Por lo que el siguiente número es 1994.

3) 2, 6, 18, 54, 162, ¿?
(*5 puntos*)

Respuesta:

Si multiplicamos el 2 por 3 tenemos 6, si multiplicamos 6 por 3 tenemos el 18, etc. Entonces el siguiente número es: 162 por 3 que es: 486.

4) 1, 1, 2, 4, 7, 13, 24, ¿?
(*5 puntos*)

Respuesta:

Sumando los 3 primeros números obtenemos el cuarto, sumando el 1, 2 y 4 obtenemos el 7, sumando el 2, 4 y 7 tenemos el 13 y sumando el 7, 13 y 24 se obtendría el 44.

5) 100, 121, 144, 169, 196, 225, ¿?
(*5 puntos*)

Notemos que 100 es cuadrado del 10, 121 es el cuadrado del 11, 144 es el cuadrado de 12, etc… Como 225 es cuadrado de 15, entonces el siguiente número es el cuadrado de 16 (todas las raíces son consecutivas) que es igual a 256.

II.- Resuelve los siguientes problemas:

6) En una OLIMPIADA DE INFORMÁTICA asistieron 45 concursantes. Todos manejaban al menos un lenguaje de programación ya sea C++, Java o Pascal. Se realizaron algunas preguntas a cada uno de los concursantes y se anoto cuantos jóvenes respondieron afirmativamente
a) ¿Sabes programar en C++? R = 26
b) ¿Y en Java? R = 20
c) ¿Y en Pascal? R = 20
d) ¿Y en C++ y en Java? R = 11
e) ¿Y en C++ y Pascal? R = 7
f) ¿Y en Java y Pascal? R = 8

¿Cuántos concursantes saben programar tanto en C++, como en Java y Pascal?

(*15 puntos*)

Respuesta:

Recurramos a un esquema de círculos como se muestra en la figura 1. Cada círculo encierra a una cantidad (por ahora variables desconocidas) de personas que conocen cada uno de los lenguajes. Cuando dos círculos se interceptan, significa que las personas que están en esa zona, son personas que conocen dos lenguajes. Si una persona

está en la intercepción de los tres círculos, significa que sabe los tres lenguajes.

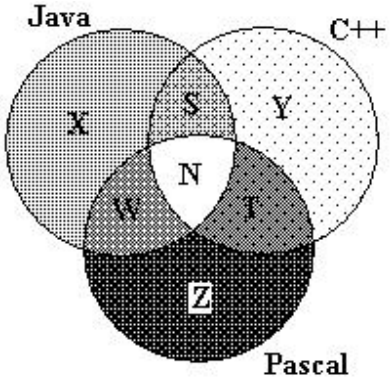


Figura 1

X es el número de personas que saben SOLAMENTE JAVA, N es el número de personas que saben LOS TRES LENGUAJES, etc...

De esta imagen se desprende las siguientes ecuaciones según el problema:

Como todas las personas son 45, se tiene la siguiente ecuación:
$X + Y + Z + W + S + T + N = 45$
Como 20 saben Java entonces:
$X + S + N + W = 20$
Como 26 saben C++ entonces:
$Y + S + N + T = 26$
Como 20 saben Pascal entonces:
$T + Z + N + W = 20$
De igual modo podemos pensar que 11 saben Java y C++ entonces:
$S + N = 11$
Análogamente se tienen las ecuaciones:
N + T = 7
N+ W = 8

Por tanto ahora podemos resolver el sistema de ecuaciones al tener 7 variables y 7 ecuaciones que las relacionan.

Quedando como:

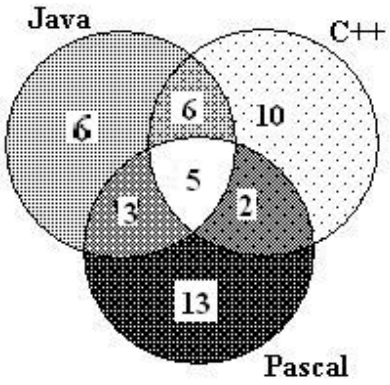


Figura 2

Entonces 5 saben los tres lenguajes.

7) En una OLIMPIADA DE INFORMÁTICA asistieron 80 concursantes. Todos manejaban al menos un lenguaje de programación ya sea C++ o Pascal. Se realizaron dos preguntas a cada uno de los concursantes, la primera: ¿Sabes programar en C++? A lo que 60 jóvenes respondieron afirmativamente. La segunda pregunta fue: ¿Sabes programar en Pascal? En esta ocasión sólo 35 jóvenes respondieron que si.

De los 80 concursantes, ¿cuántos saben programar tanto en C++ como en Pascal?

(*10 puntos*)

Hagamos su diagrama:

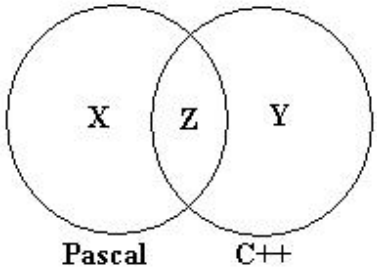


Figura 3

Ahora podemos anotar que
$X + Y + Z = 80$
$X + Y = 60$
$Y + Z = 35$

Resolviendo el sistema se tiene:

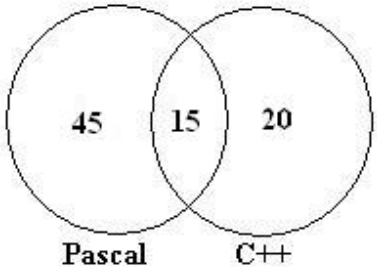


Figura 4

8) En una cuadrícula de NxN se acomodaron N^2 alumnos de la OMI. Antes de comenzar el examen se pidió que se desearan suerte unos con otros. Cada alumno saludo únicamente a los adyacentes a él. Los de las esquinas saludaron a 3, los que están en las orillas a 5 y los del centro saludaron a ocho. Por ejemplo:

X	X			
B	X			
X	X	X	X	X
		X	C	X
X	X	X	X	X
A	X			

Figura 5

Alguien vio que se dieron 72 apretones de mano. ¿Cuántos alumnos había en el salón?

(*15 puntos*)

Para este problema es necesario contar según la posición de los alumnos. Así pues, un alumno en una esquina sólo saluda a 3, mientras que en una orilla (que no sea esquina) se saluda a 5. Los restantes saludan a 8. Por lo tanto contemos cuantos hay de cada uno.

4 esquinas que saludan a 3
4 orillas (quitando esquinas) con N-2 alumnos cada una (porque quitamos las esquinas.
$(N-2)^2$ *alumnos del centro.*

Recordemos que estamos contando los saludos de cada alumno pero un saludo involucra a dos alumnos, por lo tanto debemos multiplicar por 2 para tener el número de saludos individual. Entonces sumando tenemos:

$4*3 + 5*4*(N-2) + 8*(N-2)^2 = 2*72$
$2*3 + 5*2*(N-2) + 4*(N-2)^2 = 72$
$6 + 10N - 20 + 4N^2 - 16N + 16 = 72$
$4N^2 - 6N + 2 = 72$
$4N^2 - 6N - 70 = 0$

Empleando la formula cuadrática:
$N = 5$

9) Hoy es sábado 14 de febrero de 2004, ¿qué día de la semana será el 16 de febrero de 2008?

Respuesta:

Veamos que cada 7 días se repite sábado. Como un año tiene 365 días, cuando no es

bisiesto. Dividamos esos días entre 7 y tenemos que hay 52 sábados con un día de sobra. Quiere decir que si el 14 de febrero fue sábado, entonces en el siguiente año será un lunes. En un año bisiesto se los días se desfasan dos días porque se tienen 366 días. En conclusión tenemos 3 años no bisiestos y uno si. Por tanto recorremos 5 días de la semana. Entonces, el 14 de febrero del 2008 será un jueves.

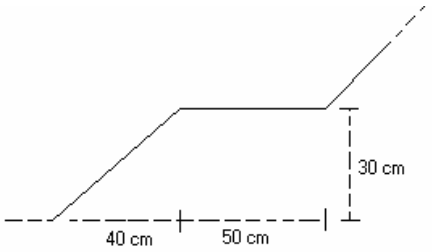
(*7 puntos*)

10) Cierta persona nació el 14 de febrero de 1980, ¿qué día de la semana nació?

(*7 puntos*)

Veamos que cada 7 días se repite sábado. Como un año tiene 365 días, cuando no es bisiesto. Dividamos esos días entre 7 y tenemos que hay 52 sábados con un día de sobra. Quiere decir que si el 14 de febrero fue sábado, entonces en el siguiente año será un lunes. En un año bisiesto se los días se desfasan dos días porque se tienen 366 días. En conclusión tenemos 24 años, de los cuales 6 son bisiestos y 18 no. Por tanto recorremos 12 días de la semana por los bisiestos y 18 por los otros. Por lo tanto el 14 de febrero del 1980 fue un jueves.

11) La casa de una familia de personas muy altas tiene una escalera algo extraña, los escalones no son como los que conocemos normalmente, sino que tienen la siguiente forma:



La escalera completa tiene 73 escalones (el último escalón no tiene la parte horizontal). ¿Cuánta alfombra se necesita para poder cubrir toda la escalera?

(*7 puntos*)

Respuesta:

*Podemos calcular la pendiente aplicando el teorema de Pitágoras. La pendiente se calcula como: la raíz de $40^2 + 30^2 = 2500$ Entonces la pendiente mide 50. Como tenemos 73 escalones y 74 pendientes (antes de llegar al piso hay una pendiente) entonces tenemos (73+74)*50 = 147*50=7,350 cm*

12) Tres señores y dos chavitos van de excursión a un bosque; luego de caminar por algunas horas se encuentran con un río que deben cruzar para continuar su camino, afortunadamente encuentran de su lado del río una vieja balsa que pueden utilizar. Luego de analizarla por un rato, el más experimentado se da cuenta de que la balsa sólo puede soportar a la vez el peso de un solo señor o de los dos chavitos. ¿Cómo deben utilizar la balsa para poder cruzar todos el río? (supón que todos saben "manejar" la balsa).

(*10 puntos*)

Respuesta:

1) Cruzan los dos chicos.
2) Se regresa uno de los niños.
3) Cruza un adulto.
4) Regresa el niño que se había quedado.

Entonces ya cruzo un señor y estamos como al principio. Repetimos los pasos 14 hasta acabar de pasar todos.

13) En la siguiente figura se muestra el diagrama de un cuarto en el que se encuentran 6 cajas muy pesadas (cuadros sombreados). Describe la manera en la que una persona colocada en el cuadro "inicio" puede llegar al cuadro marcado como "salida" si solamente puede EMPUJAR UNA CAJA A LA VEZ en cualquier dirección (NO PUEDEN JALAR cajas).

(*12 puntos*)

Figura 6

Respuesta:

Primero empujamos, en uno, a los cuadros derecho e izquierdo de la primera línea como e muestra en la figura 7. Ahora hacemos la caja central a un lado y repetimos la operación para la siguiente línea.

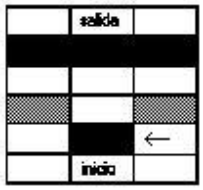


Figura 7

III Diseño de circuitos.

SOLUCIÓN PENDIENTE

IV.- Analiza las siguientes instrucciones para poder responder los problemas correspondientes.

Te encuentras ante un programa que juega gato, el clásico juego de dos personas en la cual se trazan cuatro líneas de la forma #.

Tú dispones de una serie de funciones las cuales debes de emplear en una función llamada TURNO, Esta función es llamada cada vez que toca tu turno.

Debes considerar:

1) El juego termina cuando alguien logra una línea de tres o ya no hay espacios libres.

2) Una función que tiene éxito en su propósito termina tu turno.

Las funciones son:

Bloquea ( ): Busca una posición en la cual evitas que al siguiente turno tu oponente forme una línea de tres. Si encuentra una posición, entonces marca y termina tu turno. De lo contrario, no hace nada.

Línea ( ): Busca una posición tal que en una línea haya dos espacios vacíos y un símbolo tuyo. Si encuentra la posición,

entonces marca y termina tu turno. De lo contrario, no hace nada.

Gana ( ): Busca una posición en la cual al colocar tu símbolo, formas una línea de tres. Si la encuentra, entonces la marca y GANAS de lo contrario, no hace nada.

Marca ( ): Busca una posición libre escogida al azar y la marca terminando tu turno.

Trampa ( ): Busca una posición tal que genere una jugada doble. Una jugada es doble si para el siguiente turno puedes formar una línea de tres sin importar donde tire tu oponente. Si encuentra una posición, entonces marca la posición y termina tu turno. En caso contrario, no hace nada.

Programa de ejemplo:

```
TURNO( )
        {
        Bloquea( )
        Trampa( )
        Marca( )
        }
```

16) Tú eres el jugador uno y tú pones X. Debes escribir una función que nunca pierda, dependiendo de las condiciones iniciales que se te den en cada inciso.

a)

O		O
X	X	O
		X

b)

	O	
X	X	O

(*5 puntos cada inciso*)

Respuesta:

a) Observemos que para este caso no podemos ganar haciendo una línea en nuestro turno inmediato. Por tanto, no nos sirve de nada colocar la función "Gana" Por otro lado, podemos hacer una "Trampa" pero eso dejaría libre la casilla superior central y nuestro oponente nos ganaría en su siguiente turno. En conclusión, Primero debemos emplear la función "Bloquea". Por último empleamos la función "Marca" pues independientemente después del turno que haga nuestro adversario, sólo nos queda una opción.

El programa queda como:

```
TURNO( )
        {
        Bloquea()
        Marca()
        }
```

b) Observemos que es posible colocar una trampa en las casillas de la izquierda sin riesgo de que nos ganen al siguiente turno. Por tanto, basta con colocar la función "Trampa" y "Gana". Recordemos que las funciones no especifican como escogen sus opciones, sólo garantizan que si existe al menos una opción, entonces aplican su objetivo, aunque no dicen que pasa si hay más de una opción. Por ello, es importante checar el orden de las funciones ya que si colocamos primero "Trampa" podemos perder como se muestran en el esquema:

	O	
X	X	O
X		

	O	**O**
X	X	O
X		

	O	O
X	X	O
X		**X**

Para nuestro primer turno se pudo colocar una trampa, nuestro oponente bloquea pero para nuestro siguiente turno, también se coloca una trampa. Finalmente se pierde con:

O	**O**	**O**
X	X	O
X		X

Entonces la función debe quedar como:

```
TURNO( )
        {
        Gana()
        Trampa()
        }
```

17) Escribe un programa para ganar la mayor cantidad posible de juegos de gato sin importar como empiece el tablero.

(*15 puntos*)

Ahora ordenemos nuestras prioridades. Lo más importante es "GANAR" aunque algunos digan que ganar no lo es todo✍ (es lo único). Además si no podemos ganar en cada turno es importante asegurarse de no perder en el siguiente turno. Por tanto, después de intentar ganar, debemos intentar bloquear las opciones de "gane" del otro opositor. Si no podemos bloquear, pues nos quedan tres sopas: tirar al azar (la cual no parece tan buena idea), tirar tratando de formar una línea y la tercera es tratar de generar una trampa. De todas estas tres, la mejor es tratar de generar una trampa, ya que si formamos una línea es posible que nos la "tapen" en el siguiente turno. Por último, empleamos la función línea y la función marca como últimos recursos en ese orden.

La función turno quedaría como:

```
TURNO( )
        {
        Gana()
        Bloquea()
        Trampa()
        Línea()
        Marca()
        }
```

V. La máquina de Turing es un aparato que consta de una cinta dividida en celdas en las que se pueden escribir letras o dígitos, tiene una cabeza capaz de moverse a lo largo de la cinta ya sea hacia la derecha o hacia la izquierda y de escribir sobre ella.

Esta maquina, como cualquier computadora, se puede programar y la manera de hacerlo es escribiendo una lista de instrucciones que debe realizar la máquina. Las instrucciones deben estar en el siguiente formato:

edo_Ini, val_Ini, nvo_Edo, nvo_Val, mov

El *edo_Ini* es el nombre del estado en el que podría encontrarse la máquina, el *val_ini* es el valor que tiene escrito la cinta en la posición donde se encuentra la cabeza en ese momento. Si la máquina se encuentra en ese estado y la cinta tiene ese valor entonces la máquina cambiará su estado a "*nvo_Edo*" y cambiará el valor que está escrito en la cinta por "*nvo_Val*", finalmente la máquina hará el movimiento indicado por "*mov*", los posibles movimientos son D para "derecha" e "I" para izquierda. Los posibles valores son

"0", "1" y "B" donde "B" indica "celda en blanco".

La máquina se detiene cuando no encuentra ninguna instrucción dentro del

programa que le indique qué hacer en su estado actual.

Ejemplo:

El siguiente ejemplo es un programa que mueve la cabeza de la máquina de Turing hacia la derecha hasta que pase el primer "0":
Estado de inicio del programa: *Inicio*

Inicio,	1,	movDerecha,	1,	D
Inicio,	0,	fin,	0,	D
movDerecha,	1,	movDerecha,	1,	D
movDerecha,	0,	fin,	0,	D

18) Escribe un programa que mueva la cabeza de la máquina de Turing hacia la izquierda hasta que pase el primer '1'.

(*5 puntos*)

Respuesta:

El programa quedaría como:

inicio, 1, fin, 1, I
inicio, 0, inicio, 0, I

Este programa es muy sencillo, simplemente entra en un ciclo que termina únicamente cuando se encuentra un "1". El ciclo es la segunda línea, el "detector de unos" es la primera línea, cuando lo encuentra simplemente cambia el estado de la maquina a un estado que no esta asociado con una instrucción lo que provoca que la maquina se detenga.

19) Escribe un programa que "recorra" hacia la izquierda todos los 1's que se encuentran a la derecha de la cabeza de tal manera que no haya "huecos" entre series de 1's, por ejemplo, si la cinta es:

100000011111000001111000

y la cabeza se encuentra en la parte de la izquierda (primer 1), entonces al final del programa la cinta debe quedar así:

111111111000000000000000

El estado inicial es "Inicio" y puedes suponer que la cabeza de la máquina siempre empieza en una celda donde hay un 1.

(*27 puntos*)

El programa queda como:

1) inicio, 1, buscandoCero, 1, D
2) buscandoCero, 1, buscandoCero, 1, D
3) buscandoCero, 0, buscandoUno, 0, D
4) buscandoUno, 0, buscandoUno, 0, D
5) buscandoUno, 1, recorriendoUno, 0, I
6) recorriendoUno, 0, recorriendoUno,0, I
7) recorriendoUno, 1, colocandoUno, 1, D
8) colocandoUno, 0, buscandoUno, 1, D

Empezamos a buscar el primer cero, ya que los 1's del principio deben dejarse intactos. La segunda instrucción es el ciclo que busca el primer cero. Después del primer cero, todos los unos de la derecha deben moverse. La cuarta instrucción es el ciclo que busca el primer uno. Al encontrar un uno, comenzamos a recorrerlo (colocamos un cero en su lugar). La sexta instrucción es para el "recorrido", simplemente buscamos el primer uno a la izquierda (es el "tope" a la izquierda). Si la instrucción 7 se ejecuta

entonces ya encontramos "tope", hay que colocar el "1" a la derecha de donde estamos (Instrucción 8) y volvemos a buscar un uno.

Ejemplo: los símbolos "<" y ">" representan la posición actual de la cabeza de la máquina. A la derecha se encuentra el "estado actual de la máquina"

Cinta inicial: 1 0 0 1 1 1 0 1 0 1 0 0
Estado inicial: inicio

Ejecución:

<1>0 0 1 1 1 0 1 0 1 0 0 : inicio
1<0>0 1 1 1 0 1 0 1 0 0 : buscandoCero
1 0<0>1 1 1 0 1 0 1 0 0 : buscandoUno
1 0 0<1>1 1 0 1 0 1 0 0 : buscandoUno
1 0<0>0 1 1 0 1 0 1 0 0 : recorriendoUno
1<0>0 0 1 1 0 1 0 1 0 0 : recorriendoUno
<1>0 0 0 1 1 0 1 0 1 0 0 : recorriendoUno
1<0>0 0 1 1 0 1 0 1 0 0 : colocandoUno
1 1<0>0 1 1 0 1 0 1 0 0 : buscandoUno
1 1 0<0>1 1 0 1 0 1 0 0 : buscandoUno
1 1 0 0<1>1 0 1 0 1 0 0 : buscandoUno
1 1 0<0>0 1 0 1 0 1 0 0 : recorriendoUno
1 1<0>0 0 1 0 1 0 1 0 0 : recorriendoUno
1<1>0 0 0 1 0 1 0 1 0 0 : recorriendoUno
1 1<0>0 0 1 0 1 0 1 0 0 : colocandoUno
1 1 1<0>0 1 0 1 0 1 0 0 : buscandoUno
1 1 1 0<0>1 0 1 0 1 0 0 : buscandoUno
1 1 1 0 0<1>0 1 0 1 0 0 : buscandoUno
1 1 1 0<0>0 0 1 0 1 0 0 : recorriendoUno
1 1 1<0>0 0 0 1 0 1 0 0 : recorriendoUno
1 1<1>0 0 0 0 1 0 1 0 0 : recorriendoUno
1 1 1<0>0 0 0 1 0 1 0 0 : colocandoUno
1 1 1 1<0>0 0 1 0 1 0 0 : buscandoUno
1 1 1 1 0<0>0 1 0 1 0 0 : buscandoUno
1 1 1 1 0 0<0>1 0 1 0 0 : buscandoUno
1 1 1 1 0 0 0<1>0 1 0 0 : buscandoUno
1 1 1 1 0 0<0>0 0 1 0 0 : recorriendoUno
1 1 1 1 0<0>0 0 0 1 0 0 : recorriendoUno
1 1 1 1<0>0 0 0 0 1 0 0 : recorriendoUno
1 1 1<1>0 0 0 0 0 1 0 0 : recorriendoUno
1 1 1 1<0>0 0 0 0 1 0 0 : colocandoUno
1 1 1 1 1<0>0 0 0 1 0 0 : buscandoUno
1 1 1 1 1 0<0>0 0 1 0 0 : buscandoUno
1 1 1 1 1 0 0<0>0 1 0 0 : buscandoUno
1 1 1 1 1 0 0 0<0>1 0 0 : buscandoUno
1 1 1 1 1 0 0 0 0<1>0 0 : buscandoUno
1 1 1 1 1 0 0 0<0>0 0 0 : recorriendoUno
1 1 1 1 1 0 0<0>0 0 0 0 : recorriendoUno
1 1 1 1 1 0<0>0 0 0 0 0 : recorriendoUno
1 1 1 1 1<0>0 0 0 0 0 0 : recorriendoUno
1 1 1 1<1>0 0 0 0 0 0 0 : recorriendoUno
1 1 1 1 1<0>0 0 0 0 0 0 : colocandoUno
1 1 1 1 1 1<0>0 0 0 0 0 : buscandoUno
1 1 1 1 1 1 0<0>0 0 0 0 : buscandoUno
1 1 1 1 1 1 0 0<0>0 0 0 : buscandoUno
1 1 1 1 1 1 0 0 0<0>0 0 : buscandoUno
1 1 1 1 1 1 0 0 0 0<0>0 : buscandoUno
1 1 1 1 1 1 0 0 0 0 0<0>: buscandoUno

20) ¿Qué tendrías que modificar del programa anterior para que funcione aún si no conocemos la posición de la cabeza(es decir, si no sabemos si la cinta tiene un 1 o un 0 en esa posición)? (basta con que describas la modificación, no importa que no escribas el programa)

(*5 puntos*)

Respuesta:

Para el último problema, basta agregar una condición que coloque la cabeza de la máquina sobre un "1", después el mismo programa debe funcionar:

inicio, 0, inicio, 0, D

Es decir, "mientras te encuentres sobre un 0, muévete a la derecha". Nota que cuando la máquina encuentre un "1", el estado será

inicio, 1

Por lo que entra en el programa original.