

Programame



Concurso Regional Programame
Zaragoza
16 de Marzo de 2018

Cuaderno de Problemas



Índice de problemas

A. Tres en Raya	3
B. Dardos	4
C. Rutas de Aviones	5
D. Tiempos de Guerra	6
E. El Juego del Dominó	7
F. La Báscula	8
G. Escalera Alta	9
H. Guerra de Globos	10
I. Partida de ajedrez	11

Ejercicios realizados por:

- José Manuel Guallar (Colegio Montessori)
- Fernando Valdeón (Colegio Montessori)
- Santiago Faci (Colegio Montessori)

A. Tres en Raya



El juego de 3 en raya es un juego muy conocido en el que dos jugadores se enfrentan por colocar 3 de sus fichas formando una línea horizontal, vertical o en cualquiera de las diagonales. Lo habitual es que uno de los jugadores juegue con las fichas O (letra O) y otro con X.

En este caso son dos computadores los que juegan entre si para ver quién gana y concretamente se encuentran en el momento más crítico, que es cuando la partida aún no se ha decidido y sólo quedan 2 casillas por cubrir. Debemos terminar el algoritmo que permite a las máquinas jugar de forma que éstas intenten ganar la partida a toda costa, por lo que su movimiento siempre tenderá a ser el de hacer línea con 3 de sus fichas.

Hay que tener en cuenta que en caso de que no haya opción de ganar se colocará la pieza en la primera casilla que se encuentre, recorriendo el tablero empezando por la casilla superior izquierda y recorriendo fila a fila el mismo (de izquierda a derecha y de arriba a abajo).

Entrada

Como entrada se recibe, en la primera línea, el símbolo de la pieza del jugador al que le toca mover y, a continuación, el estado del tablero a falta de dos movimientos para terminar la partida (ésta termina cuando se llena el tablero).

Salida

Como salida se obtendrá un mensaje que indique quien gana (GANA O ó GANA X). En caso de que ninguno de los jugadores logre hacerse con la victoria se indicará NADIE GANA

Entrada de ejemplo

```
O
X _ X
X O O
_ O O
```

Salida de ejemplo

```
GANA O
```

B. Dardos

Los dardos es un juego muy sencillo con un sinfín de variantes. En nuestro caso tenemos que programar una máquina de dardos de forma que sea capaz de decidir cuál de los jugadores gana una partida a lo que se conoce como el juego del 101.

En este juego, los jugadores (un número entre 1 y 4), comienzan la partida con 101 puntos y los van perdiendo a medida que van acertando en las diferentes zonas de la diana (entre 1 y 20 para los exteriores, 25 y 50 para las dos partes centrales de la misma). Para ganar se debe llegar al 101 de forma exacta de forma que si, por ejemplo, un jugador estuviera a falta de 3 puntos e hiciera diana en un valor 5,

el valor “rebotaría” y se quedaría con un valor de 2, por lo que debería seguir jugando.



Entrada

Como entrada se recibe el número de jugadores, un número indicando el número de turnos que ha durado la partida y, a continuación, una lista con las tiradas de cada uno de los jugadores. Se tendrá en cuenta que los valores pueden multiplicarse por 2 ó por 3 en función de la zona de la diana donde se acierte con el dardo. En ese caso la puntuación vendrá acompañada de un x2 o un x3 en función de si la puntuación es doble o triple.

Salida

Se debe indicar el jugador que ha ganado de la siguiente forma: GANA JUGADOR x (Siendo x el número de jugador según la posición que ocupan sus tiradas en la entrada de datos). En caso de que ninguno de los jugadores consiga llegar al 101 exactamente, se indicará EMPATE

Entrada de ejemplo

```
2
11
1 10x2 3x3 3 4 4 5 12 12x3 10 3
1 11x3 3 4 2 1 2 25 25 1 1
```

Salida de ejemplo

```
GANA JUGADOR 1
```

C. Rutas de Aviones



El famoso buscador de vuelos Skyscanner quiere mejorar su servicio y ahora va a incorporar una opción para que sus usuarios puedan combinar vuelos entre diferentes ciudades y compañías y puedan saber si será posible encadenar dichos vuelos para llegar a su destino. Para ello, el buscador solicita la ciudad de origen y destino y todas las intermedias, además de las horas de salida duración de todos los vuelos. Así, el buscador podrá determinar si es posible o no encadenar dichos vuelos.

Debido a que los vuelos siempre se encuentran en hora local, el sistemas necesitará como entrada la diferencia horaria de todas las ciudades intermedias y destino con respecto a la origen. En caso de que la zona horaria de una ciudad destino sea la misma que la de origen, se indicará con el valor 0 para esa ciudad.

Como limitación para el buscador en esta primera versión sólo se podrán tener en cuenta vuelos cuya hora de salida sea la misma que la del vuelo origen.

Entrada

Como entrada se enviará la ruta indicando, para cada ciudad, la hora local de salida, el nombre de la ciudad y la duración en horas del vuelo a la siguiente ciudad. El sistema también necesitará la diferencia horaria de las ciudades intermedias y destino con respecto a la origen. Se pasará un 0 para indicar que no hay más datos de entrada.

Para la última ciudad no será necesaria la duración del viaje por lo que no se enviará en ningún caso. Podemos estar seguros de que, tanto las duraciones como las horas de salida de los vuelos, son horas exactas.

Salida

Como salida se obtendra un si ó no en función de si el avión puede realizar la ruta a tiempo o no.

Entrada de ejemplo

```
8h Madrid 1h - 12h Barcelona 9h - 16h Chicago
Barcelona 0
Chicago -7
0
```

Salida de ejemplo

```
si
```

D. Tiempos de Guerra



The Imitation Game (Descifrando el enigma) es una película que habla de la vida de Alan Turing un matemático británico, conocido en muchos casos como uno de los padres de la computación. Uno de sus logros fue descifrar los códigos generados por la máquina Enigma, usada por los nazis para enviar mensajes encriptado en la Segunda Guerra Mundial.

En nuestro caso, los mensajes son secuencias de caracteres, en las que se ocultan números. Estos números son el código ASCII de los caracteres encriptados del mensaje, con una peculiaridad: Si el número es mayor de 99, el valor son las 2 últimas cifras. Con los 100 primeros caracteres podemos representar cualquier mensaje:

L65Gdg99-T234066Ag1197

65 → A,

99 → c,

234066 → 66 → B,

1197 → 97 → a

Cada número que representa el código ascii de un carácter está separado del siguiente al menos por otro carácter no numérico.

Entrada

La entrada consiste en una cantidad variable de casos de prueba, cada uno en una línea. Cada caso de prueba consiste en una cadena de caracteres que representa un mensaje encriptado. En cada cadena de caracteres podemos encontrar números que representan el código Ascii de un carácter.

Los números escondidos en las cadenas de texto no son mayores a 10^6 .

Salida

Por cada caso de prueba el programa escribirá la cadena de caracteres obtenida a partir de los números Ascii encontrados en ella. En caso de que no contenga ningún número, escribe SIN MENSAJE.

Entrada de ejemplo

```
35065Carro-&184comb65#18067-3465te@#1182
65065Tro176pas65enem1183gas49:5050
AliADoS_C#pT:uraD+S+
```

Salida de ejemplo

```
ATACAR
ALAS12
SIN MENSAJE
```

E. El Juego del Dominó



En el juego del dominó se emparejan fichas atendiendo a los valores que contienen. Dichos valores van del 0 al 6, y cada ficha tiene 2 caras. Consiste en emparejar los valores de las fichas hasta que un jugador haya terminado con las suyas:

0:3 3:6 6:2 2:4 4:5

Reglas:

1. No hay fichas repetidas por cada partida.
2. Todos los jugadores empiezan con el mismo número de fichas ≥ 1 .
3. Cada jugador usa la primera ficha que pueda colocar en la mesa, evaluando sus fichas de izquierda a derecha.
4. Si una ficha se puede poner tanto a la izquierda de la mesa como a la derecha, se pondrá siempre a la izquierda.
5. El primer jugador que se quede sin fichas, gana (Esto podría ocurrir en el primer movimiento)
6. Si nadie puede poner fichas, el que menos fichas tiene, gana
7. Si nadie puede poner fichas, y no hay nadie con menos fichas que otro, hay empate.

Entrada

Está compuesta por varios casos de prueba. Cada caso de prueba representa una partida y comienza por un número (2-6) indicando el número de jugadores de la partida, y a continuación una línea por cada jugador con las fichas separadas por un espacio. Una línea con un 0 indica la ausencia de más casos de prueba

Salida

Por cada caso de prueba el programa debe escribir el jugador que gana, o empate, si nadie puede poner ficha y no hay un jugador con menos fichas que los demás.

Entrada de ejemplo

```
2
0:5
6:6
3
1:4 2:5
5:1 6:6
3:5 6:2
4
1:1 3:6 0:1
3:1 4:5 3:3
1:4 6:2 4:6
2:5 5:5 5:0
0
```

Salida de ejemplo

```
JUGADOR 1
EMPATE
JUGADOR 4
```

F. La Báscula



En los gimnasios hay personas que hacen ejercicio para perder peso, aunque también hay personas que entrenan para ganarlo. Por norma general, si se queman más calorías de las que se ganan alimentándonos, se pierde peso y viceversa. Aunque la relación entre calorías quemadas o consumidas y el peso depende de muchos factores, se suele dar por hecho que 1 kg de masa equivale a unas 8000 calorías.

Para alcanzar el peso esperado puede que se deba perder peso o ganarlo. El que quiera pesar menos, lo habrá conseguido si después de los días de gimnasio pesa igual o menos a lo esperado, y quien quiera ganar peso deberá pesar lo mismo o más de lo esperado.

Entrada

La entrada comienza con un número indicando cuantos casos de prueba hay que procesar.

Cada caso de prueba está compuesto por 3 números en la misma línea: d , i , f , indicando la cantidad de días que ha ido al gimnasio, el peso del primer día, y el peso final esperado. (Peso inicial siempre es distinto a peso esperado, y no son negativos).

A continuación, seguirán d líneas con dos números p , g , indicando las calorías perdidas en el gimnasio y las ganadas alimentándose.

Todos los números son enteros menores a 10^6 .

Salida

Por cada caso de prueba, el programa escribirá una línea indicando si consigue alcanzar su peso esperado o no.

Entrada de ejemplo

```
3
2 70 68
9000 1000
10000 2000
4 85 87
4000 8000
1500 7034
2030 9041
1465 3456
2 80 82
1000 9000
1001 9000
```

Salida de ejemplo

```
CONSEGUIDO
CONSEGUIDO
NO CONSEGUIDO
```

G. Escalera Alta



En el póker una escalera se compone de 5 cartas seguidas y consecutivas.
Si además son del mismo color, se conoce como escalera de color.

Cada color está compuesto de 13 cartas: 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A

El As (A) en póker representa la carta más alta y también puede representar la carta más baja. El As puede terminar una escalera después de la K, o empezarla antes del 2.

En nuestra baraja vamos a jugar con un solo color, así que todas las escaleras serán de color.

Entrada

La entrada está compuesta por múltiples casos de prueba, cada uno en una línea.

Cada caso de prueba indicará las cartas que tenemos (mínimo 1, máximo 13) separadas por un espacio.

Las cartas están siempre ordenadas y no hay cartas repetidas.

Salida

Para cada caso de prueba se escribirá en una línea la escalera más alta que se puede conseguir, o se indicará que no hay escalera.

Entrada de ejemplo

```
A 2 3 4 5 7 8 10 J K
2 3 4 5 6 7 8 9 10 J Q K A
2 5 6 7 8 9 10 J K A
2 3 4 5 7 8 9 10 Q K A
3 4 5 6
Q
```

Salida de ejemplo

```
ESCALERA 5
ESCALERA A
ESCALERA J
NO HAY ESCALERA
NO HAY ESCALERA
NO HAY ESCALERA
```

H. Guerra de Globos



Quién no recuerda las batallas con globos de agua que se libraron en el parque cuando éramos niños. Y cómo se sigue divirtiendo Jaimito mojando a los otros niños que juegan en el parque.

Jaimito está subido en un árbol alto que hay en el centro del parque y tira globos con la intención de mojar al mayor número de niños posible.

Desde la altura del árbol, Jaimito es capaz de alcanzar cualquier lugar del parque, por lo que va variando el lugar donde tira el globo dependiendo de la cantidad de niños que hay alrededor.

El parque es cuadrado (4x4 casillas) y los globos pueden mojar a cualquier niño que se encuentre a una casilla de distancia del lugar del impacto.

Entrada

La entrada está compuesta por un número variable casos de prueba. Cada caso de prueba está definido por 4 líneas con 4 caracteres. Cada uno de estos cuadrados (4x4) representa el parque.

Dentro de estos cuadrados, la (G) marca el lugar del impacto del globo y la (N) los lugares donde hay un niño. Si no hay nadie se indica con una almohadilla (#).

Después de cada caso de prueba hay una línea con 4 guiones para facilitar la lectura.

Salida

Por cada caso de prueba, el programa debe mostrar la cantidad de niños que son mojados por el impacto del globo.

Entrada de ejemplo

```
NNNN
NNGN
NNNN
NNNN
----
GNN#
NNN#
NN##
#NN#
----
#NNN
N###
N#G#
N###
----
```

Salida de ejemplo

```
8
3
0
```

I. Partida de Ajedrez



IBM está pensando en desarrollar una nueva versión de un supercomputador capaz de competir contra los grandes maestros del ajedrez. Será el Deep Blue 2 y necesitan tu ayuda para mejorar el algoritmo que detecta cuando y cómo comerse las fichas del adversario cuando se presente la ocasión.

Para ello, primero hay que tener en cuenta cómo se puede mover cada una de las fichas:

- Peón (P): Se puede comer a la ficha que se encuentra justo en la casilla siguiente en una de las dos diagonales
- Torre (T): Se puede comer a cualquier ficha que se encuentre justo en la línea horizontal o vertical a cualquier distancia
- Alfil (A): Se puede comer a cualquier ficha que se encuentre en cualquier sentido de las diagonales y a cualquier distancia
- Rey (R) : Se puede comer a cualquier ficha que se encuentre en cualquiera de las casillas de al lado
- Reina (I): Se puede comer a cualquier ficha que se encuentre en cualquier de las diagonales y en las líneas horizontal o vertical, y a cualquier distancia
- Caballo (C) : Se puede comer a cualquier ficha que se encuentre en la casilla que forme una L en cualquiera de sus formas desde la casilla donde él se encuentre (2 casillas para el lado largo y 1 casilla para el corto). Es la única pieza que puede desplazarse aunque haya otras piezas en el camino.

Las fichas podrán ser blancas (B) o negras (N). Por ejemplo, un peón blanco vienen representado por PB y un caballo negro vendrá como CN.

Entrada

Como entrada se reciba en una línea el color al que te le toca jugar seguido del estado del tablero justo antes del movimiento, teniendo en cuenta que siempre será una situación en la que se puede comer alguna ficha del adversario.

Salida

Como salida se obtiene el estado del teclado justo después del movimiento del jugador al que le tocaba el turno.

Entrada de ejemplo

```
BLANCAS
TN CN AN IN RN AN CN TN
PN PN PN PN PN PN XX PN
XX XX XX XX XX XX XX XX
XX XX XX XX XX XX PN XX
XX XX XX XX XX XX XX XX
PB XX XX XX XX CB XX XX
XX PB PB PB PB PB PB PB
TB CB AB RB IB AB XX TB
```

Salida de ejemplo

TN	CN	AN	IN	RN	AN	CN	TN
PN	PN	PN	PN	PN	PN	XX	PN
XX							
XX	XX	XX	XX	XX	XX	CB	XX
XX							
PB	XX						
XX	PB						
TB	CB	AB	RB	IB	AB	XX	TB