

Concurso de Programación Ada Byron

Fase local - UCM



Cuadernillo de problemas



25 de enero de 2016

In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selections amongst them for the purposes of a calculating engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.

Ada Byron

Listado de problemas

A La fiesta del verano	3
B Fin de mes	5
C Temperaturas extremas	7
D Jugando al Buscaminas	9
E Racha afortunada	11
F El código de la T.I.A.	13
G Navegando sin teclado	15

Autores de los problemas:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Enrique Martín Martín (Universidad Complutense de Madrid)
- Isabel Pita Andreu (Universidad Complutense de Madrid)
- Antonio Sánchez Ruiz-Granados (Universidad Complutense de Madrid)
- Clara Segura Díaz (Universidad Complutense de Madrid)
- Alberto Verdejo López (Universidad Complutense de Madrid)

● A

La fiesta del verano

El segundo domingo de agosto se celebra la fiesta de mi pueblo. El ayuntamiento coloca grandes carpas en la playa y pone mesas para que comamos todos los vecinos. Las mesas se organizan por edades para evitar que los chavales se aburran con la conversación de sus padres y estos a su vez no tengan que aguantar el jaleo de los más pequeños. Las mesas son todas del mismo tamaño colocándose varias para un mismo tramo de edad si es necesario.

Cada año el ayuntamiento dedica un gran esfuerzo a calcular el número de mesas necesarias, de forma que sean lo más grandes posible, no sobre ningún hueco, y al mismo tiempo sean todas ellas del mismo tamaño. Este año han decidido informatizar el proceso. ¿Puedes ayudarles?



Entrada

La entrada está formada por varios casos de prueba. Cada caso consiste en una lista de valores con el número de personas en cada tramo de edad, acabada con el valor cero. La última lista tendrá solamente el valor cero, y no debe ser procesada.

Cada tramo estará formado por al menos una persona y el número total de personas nunca será mayor que 10^{18} .

Salida

Para cada caso de prueba se indicará el número de mesas a utilizar. Recuerda que se desea que las mesas tengan el mayor tamaño posible sin que sobre ningún hueco.

Entrada de ejemplo

```
10 25 30 0
60 45 0
7 0
0
```

Salida de ejemplo

```
13
7
1
```


● B

Fin de mes

A mí no me asusta el fin del mundo; me asusta el fin de mes, porque no siempre consigo que mis ingresos lleguen conmigo. Los gastos se acumulan, y no sé qué más hacer para estirar mi triste sueldo.

Creo que el primer paso para mejorar mi situación es hacer una estimación de lo bien o lo mal que me va a ir un mes, en función de los ingresos y los gastos previstos. Sé cuánto dinero tengo en el banco al principio, y sé cuánto va a variar. ¿Me ayudas a saber si llegaré a fin de mes con dinero en el banco?



Entrada

La entrada comienza con un número que indica cuántos casos de prueba vendrán a continuación. Cada caso contiene dos números, $-10.000 \leq s, c \leq 10.000$ indicando, respectivamente, el saldo en mi cuenta bancaria el primer día del mes, y el cambio estimado (ingresos menos gastos) durante el mes.

Salida

Para cada caso de prueba, se escribirá “SI” si llego a fin de mes con saldo mayor o igual que cero, y “NO” en otro caso.

Entrada de ejemplo

```
4
100 -10
-10 -100
-10 100
100 -1000
```

Salida de ejemplo

```
SI
NO
SI
NO
```

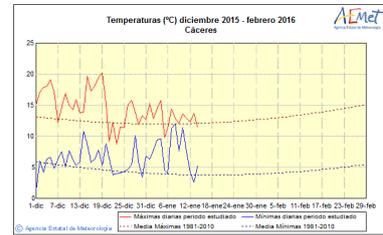



Temperaturas extremas

La Agencia Estatal de Meteorología (AEMET) actualiza diariamente el Banco Nacional de Datos Climatológicos, en el que se almacenan las observaciones climatológicas (precipitación y temperatura) realizadas en España desde hace unos 150 años.

Nuria está estudiando la relación entre la variabilidad de la temperatura y el estado hídrico del suelo en una región de secano en España. Para ello, acude al Banco y solicita los registros de temperaturas en dicha zona año a año desde que existen registros.

En la primera fase de su estudio pretende determinar el número de picos y valles en las temperaturas en un determinado periodo de tiempo. Una temperatura se considera pico (resp. valle) en una secuencia cuando la anterior y la siguiente en la secuencia son estrictamente menores (resp. mayores).



Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación.

Cada caso de prueba se compone de dos líneas. La primera de ellas tiene un único entero con el número de temperaturas registradas (mayor que 0 y menor o igual que 10.000), mientras que la segunda línea contiene la secuencia de temperaturas (números enteros entre -50 y 60 grados centígrados).

Salida

Para cada caso de prueba se escribirá el número de picos y de valles, separados por un espacio y seguidos de un salto de línea.

Entrada de ejemplo

```
4
5
7 5 3 8 9
8
8 9 7 6 5 3 4 2
2
3 -5
8
4 -1 5 3 7 7 6 8
```

Salida de ejemplo

```
0 1
2 1
0 0
1 3
```




Jugando al Buscaminas

Hoy vamos a jugar al *Buscaminas*. Seguro que lo conoces. En un tablero dividido en celdas se esconde cierto número de bombas. El objetivo es encontrarlas todas. Para ello hay que destapar todas las celdas que no esconden bombas. Como ayuda, cuando se descubre una celda sin bombas, aparece en ella el número de bombas en las (como mucho) 8 celdas colindantes. Así, si al descubrir una celda aparece un 3 eso significa que de las 8 celdas que hay alrededor 3 de ellas esconden una bomba y 5 no. Cuando se descubre una celda que no tiene bombas alrededor, en vez de un 0 la celda se deja vacía y se descubren de forma automática las 8 celdas colindantes (ahorrando al jugador tener que descubrirlas por sí mismo sin ningún riesgo). Esto provoca que en ocasiones se descubran muchas celdas de golpe. Si se descubre una celda con una bomba, se pierde la partida.



La figura muestra el estado del juego después de haber descubierto consecutivamente las cuatro esquinas del tablero.

Lo que queremos es conocer el estado del juego después de que el jugador haya descubierto una serie de celdas.

Entrada

La entrada constará de una serie de casos de prueba. Cada caso comienza con una línea con los números de filas F y columnas C del tablero ($1 \leq F, C \leq 50$). A continuación aparecen F líneas, cada una con C caracteres: un * significa que en la celda correspondiente se esconde una bomba; las casillas vacías se indican con -. Después aparece el número K de celdas a descubrir, seguido de K líneas que indican las coordenadas de cada celda descubierta: una fila entre 1 y F y una columna entre 1 y C . Pueden intentarse descubrir celdas ya descubiertas anteriormente (quizás de forma automática), en cuyo caso el estado del tablero no cambia. Si al descubrir una celda aparece una bomba, esa será la última celda a descubrir que aparezca en la entrada para ese caso.

Salida

Para cada caso de prueba, se escribirá el tablero después de haber descubierto todas las celdas indicadas en la entrada (y las que lo hayan hecho de forma automática). Por cada celda descubierta se escribirá un número indicando el número de bombas que hay alrededor, salvo que este sea 0, que se escribirá -. Las celdas sin descubrir se indicarán con X.

Si durante el juego se ha descubierto una bomba, se escribirá **GAME OVER** en vez del tablero final.

Entrada de ejemplo

```

8 8
-*---*--
-----
**-----
---*----
---*----*
-*--*---
----*---
-----
4
1 1
1 8
8 8
8 1

```

Salida de ejemplo

```
1XXXXX1-  
XXXX111-  
XXXX1---  
XXXX2-11  
XXXX311X  
XXXXX211  
1112X2--  
---1X1--
```



Racha afortunada

Vladimir es un jugador de *póquer* profesional que lleva muchos años perfeccionando su técnica de juego. Sabe que la práctica y la constancia son la clave para llegar a dominar cualquier cosa y, desde el principio, lleva registros exhaustivos de todas sus partidas.

Como todos los jugadores, tiene días mejores y días peores. Forma parte del juego, a veces se gana y a veces se pierde. Sin embargo, mirando sus registros se ha dado cuenta de que a veces tiene rachas muy buenas en las que gana casi siempre y le gustaría identificarlas.



En particular, teniendo en cuenta las ganancias o pérdidas de cada día, le gustaría conocer qué día debería haber empezado a jugar y qué día debería haberlo dejado para maximizar sus ganancias, suponiendo que jugase todos los días intermedios. ¿Puedes ayudarlo?

Entrada

El primer número de la entrada indica el número de casos de prueba que aparecen a continuación.

Cada caso de prueba está compuesto por dos líneas. La primera contiene el número de días anotados en el registro (al menos 1 y como mucho 100.000), y la segunda línea las ganancias (números positivos) o pérdidas (números negativos) de cada día. Cuando las ganancias o las pérdidas de un día llegan a 10.000, lo deja antes de que la suerte le abandone, o de que tenga que volver a casa desnudo tapándose con un barril.

Salida

Para cada caso de prueba se deberá imprimir una línea con dos números que delimitan el intervalo de días consecutivos que Vladimir debería haber jugado para maximizar sus ganancias (suponiendo que sólo jugase esos días). Ten en cuenta que Vladimir juega tanto el día de inicio como el día de fin del intervalo, y que los días empiezan a numerarse en 1.

En caso de existir varios intervalos en los que Vladimir podría obtener el máximo beneficio posible, se debe elegir el de duración más corta y, a igualdad de duración, el que empiece primero. Al fin y al cabo, a nadie le gusta trabajar más de lo necesario y todos preferimos terminar cuanto antes.

Vladimir es un profesional, por lo que se puede asumir que al menos habrá un día en el que haya ganado algo.

Entrada de ejemplo

```
3
4
-10 4 8 -2
5
-3 0 7 -20 -7
11
30 -7 1 -25 -20 25 10 3 -1 8 -100
```

Salida de ejemplo

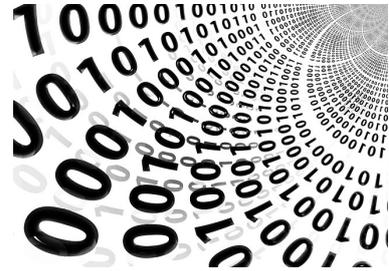
```
2 3
3 3
6 10
```




El código de la T.I.A.

Ante la nueva oleada de ataques a las redes de otras agencias de investigación, la T.I.A. ha decidido tomarse en serio la transmisión de sus mensajes entre las distintas oficinas y ha contratado los servicios de un especialista que se hace llamar Pepe Gotera.

A la vista de la solución del supuesto experto, salta a la vista que vive anclado en el pasado. Lo único que ha hecho ha sido asignar a cada símbolo un número de entre uno y tres dígitos y ha instruido a los agentes especiales para que, cuando tengan que mandar un mensaje, simplemente sustituyan las letras y símbolos del mensaje por esos códigos, dando lugar a una secuencia de números, uno detrás de otro.



Por ejemplo, en una codificación donde la A es sustituida por el 12 y la B es sustituida por el 42, el grupo musical ABBA se codifica 12424212.

Pepe Gotera además de dar el método de encriptado ha proporcionado distintas tablas de codificación que los agentes irán utilizando a lo largo del tiempo. Sin embargo, nuestro “experto” no ha oído hablar de los códigos de Huffman¹ ni nada que se le parezca, por lo que cuando le ha llegado a Otilio el primer mensaje cifrado con una de esas tablas, se ha dado cuenta de que en realidad hay varios posibles descifrados distintos. El pánico se ha instalado en las oficinas centrales de la T.I.A. y cuando le han pedido explicaciones a Pepe Gotera ha dicho que, dado que ninguno de los códigos de las letras contiene ceros, se puede decir a los agentes que utilicen ceros para separar palabras.

El Super no está aún convencido de la efectividad de la solución, así que nos ha pedido ayuda. Dada la tabla de símbolos y un texto cifrado, ¿de cuántos mensajes distintos podría provenir?

Entrada

La entrada estará compuesta por distintos casos de prueba, terminados con una línea con un 0.

Cada caso de prueba está compuesto por tres líneas. La primera contiene el número N de símbolos que tiene la tabla de codificación. La segunda línea contiene N números correspondientes a los códigos de cada símbolo. Se garantiza que no hay números repetidos y que ninguno de ellos tiene ceros.

La tercera línea de cada caso de prueba tendrá el mensaje cifrado (una sucesión de entre 1 y 1.000 dígitos) codificado con la tabla anterior. En caso de tener ceros, éstos nunca estarán al principio ni al final del mensaje.

Salida

Para cada caso de prueba se escribirá en una línea independiente el número de mensajes que pueden dar el texto cifrado dado. Como puede haber muchos, se dará el resultado módulo 1.000.000.007.

Entrada de ejemplo

```
3
1 2 22
1221
3
1 2 22
12021
3
1 2 22
12321
0
```

¹Los códigos de Huffman son *códigos prefijo*, es decir, que cumplen que la codificación de un símbolo nunca es prefijo de la codificación de ningún otro símbolo.

Salida de ejemplo

2
1
0



Navegando sin teclado

Lo de ayer fue un drama. Con los años que llevas criticando a todos los que abusan del ratón en lugar de utilizar las teclas rápidas, te tuvo que pasar a ti. Maldito teclado, que se tuvo que romper. Y encima en día festivo, imposible comprar otro. Todo el día utilizando el ordenador a través del lento e incómodo ratón. Menos mal que lo único que hiciste fue navegar por internet y pudiste ir de una página a otra utilizando los enlaces entre ellas.



Una persona normal con el orgullo herido compraría un teclado de reserva. Pero tú no. Con un flamante teclado nuevo vas a hacer un programa que te ayude a decidir la manera más rápida de navegar si te vuelves a ver en la misma situación.

Para eso has recopilado todas las páginas web que visitas y has hecho una lista de los enlaces que tiene cada una de ellas. Teniendo en cuenta la página de inicio que tienes en el navegador quieres saber cuál es la forma más rápida de llegar a una página determinada utilizando únicamente esos enlaces.

Como estás obsesionado por la velocidad, el programa tendrá en cuenta que el tiempo de carga de cada página puede ser distinto y que una vez cargada, encontrar y pinchar en cada enlace depende de lo lejos que esté del principio de la página y de lo oculto que esté. Venga, ¡manos a la obra!

Entrada

La entrada estará compuesta por distintos casos de prueba terminados por una línea con un 0.

Cada caso de prueba representa un escenario de navegación distinto, que incluye el conjunto de páginas a visitar, sus tiempos de carga y los enlaces que hay entre ellos.

La primera línea de cada caso contiene el número N de páginas manejadas ($1 < N \leq 1.000$). La página de inicio del navegador es la 1 y la página que queremos visitar la N .

A continuación viene una línea con N números que contiene el tiempo de carga (en milisegundos) de cada una de las páginas.

Aparece después una línea con un número M indicando el número de enlaces entre páginas que se han identificado, tras lo que aparecen M líneas cada una describiendo un enlace con la página en la que está, la página a la que va y el tiempo (en milisegundos) que se tarda desde que la página carga hasta que el usuario encuentra el enlace y lo pulsa.

Se garantiza que todos los tiempos serán menores a 10 segundos y que en caso de que una página tenga más de un enlace a la misma página, únicamente se proporciona aquel que puede utilizarse antes.

Salida

Para cada caso de prueba se escribirá el número mínimo de milisegundos que se tardará en poder navegar desde la página 1 hasta la página N contando el tiempo de carga de todas las páginas (incluidas la 1 y la N) y el tiempo de localización y pulsación de cada enlace.

Si no se puede alcanzar la página, se escribirá IMPOSIBLE.

Entrada de ejemplo

```
4
10 5 15 8
5
1 2 10
1 3 20
1 4 100
2 4 20
3 4 20
4
10 5 15 8
4
1 2 10
1 3 20
2 3 40
4 2 10
0
```

Salida de ejemplo

```
53
IMPOSIBLE
```