

# Olimpiada Informática Española

## Regional de Madrid



### Cuadernillo de problemas



Facultad  
de  
Informática



UNIVERSIDAD COMPLUTENSE  
MADRID

12 de febrero de 2021

*Every great developer you know got there by solving problems  
they were unqualified to solve until they actually did it*  
**Patrick McKenzie**

12 de febrero de 2021

## Listado de problemas

<b>A</b>	<b>¿En qué volumen?</b>	<b>3</b>
<b>B</b>	<b>Codificación límite</b>	<b>5</b>
<b>C</b>	<b>¿Se ha colado!</b>	<b>7</b>
<b>D</b>	<b>Comienza la temporada</b>	<b>9</b>
<b>E</b>	<b>Pepe Casanova</b>	<b>11</b>
<b>F</b>	<b>Puntos de silla</b>	<b>13</b>
<b>G</b>	<b>Ratones en un laberinto</b>	<b>15</b>
<b>H</b>	<b>¿Cuándo seré rico?</b>	<b>17</b>
<b>I</b>	<b>El carpintero Ebanisto</b>	<b>19</b>

Autores de los problemas:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Isabel Pita Andreu (Universidad Complutense de Madrid)
- Clara Segura (Universidad Complutense de Madrid)
- Alberto Verdejo (Universidad Complutense de Madrid)





## ¿En qué volumen?

En muchos jueces on-line (*¡Acepta el reto!* entre ellos) cada problema tiene un identificador único para poderlo referenciar de manera unívoca dentro del sistema. Los identificadores son números naturales correlativos, y el primer problema recibe el número 100.



Empezar en 100, en lugar de en 1 (o en 0), no es un capricho. Los problemas se “archivan” en volúmenes, cada uno compuesto por 100 problemas. Al asignar el número 100 al primer problema, es fácil saber en qué volumen está cualquier problema a partir de su identificador. En concreto, el primer volumen de problemas contiene a aquellos que tienen como identificador los números entre 100 y 199, el volumen 2 contiene los problemas con identificadores 200...299, etcétera.

Dado un problema, ¿en qué volumen está?

### Entrada

La entrada comienza con un número que indica cuántos casos de prueba vendrán a continuación. Cada uno será un número entre 100 y 999.999.

### Salida

Para cada caso de prueba, el programa deberá escribir a qué volumen pertenece el problema con ese identificador.

### Entrada de ejemplo

```
2
100
306
```

### Salida de ejemplo

```
1
3
```



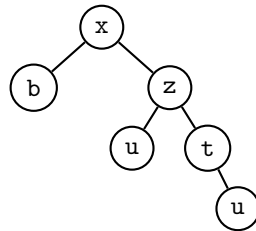
## ● B

# Codificación límite

El envío de mensajes cifrados para evitar miradas indiscretas se lleva estudiando desde la antigüedad. El método más simple consiste en manejar tablas de traducción que contienen, para cada letra, por qué otra letra se sustituirá en el mensaje cifrado.

Existe otro mecanismo simple que consiste en simplemente añadir caracteres aleatorios entre las letras del mensaje. El método que hoy proponemos utiliza este sistema. Además, requiere un pequeño esfuerzo adicional por parte del lector pues el mensaje recompuesto *no* contiene espacios separando las palabras, por lo que deberá ser él el que infiera, en el momento de leer, dónde empieza y termina cada una.

El procedimiento comienza con un mensaje cifrado como el siguiente: “xb..zu..t.u..”. Ese mensaje lo interpretaremos como un árbol binario de caracteres donde el primer carácter simboliza la raíz y a continuación aparecen el hijo izquierdo y el hijo derecho, teniendo en cuenta que el árbol vacío está representado por un punto ‘.’. El mensaje del ejemplo representa el siguiente árbol (donde se han omitido los árboles vacíos):



El mecanismo de codificación límite lo que hace es quedarse con el límite o frontera del árbol (las hojas de izquierda a derecha), y escribirlas.

## Entrada

La entrada consiste en diversas líneas, cada una con un mensaje codificado utilizando la codificación límite. Se garantiza que el mensaje será un árbol binario válido, que no tendrá más de 5.000 caracteres y cuya altura no será mayor de 3.000. Los nodos del árbol contienen caracteres de la ‘a’ a la ‘z’.

## Salida

Para cada caso de prueba, escribir una línea con el mensaje descifrado.

## Entrada de ejemplo

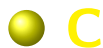
```
abh...ko..nl...a..  
xb..zu..t.u..
```

## Salida de ejemplo

```
hola  
buu
```







# ¡Se ha colado!

La carnicería de mi barrio trae el mejor producto de la zona, por lo que siempre está llena de gente. Claudio, el dueño, hace tiempo que decidió colocar un expendedor de números para poder atender a los clientes ordenadamente y sin que las discusiones entre ellos perturbasen su quehacer diario. Como no tenía suficiente dinero, no ha podido comprar además un contador digital, así que confía en que los propios clientes formen una cola respetando la numeración y no suele pedir el número cuando les atiende; circunstancia que estos han aprovechado para colarse en numerosas ocasiones.



Por ello, le ha pedido a su hija Claudia que haga una inspección sorpresa diaria. En el momento de la inspección, Claudia les pide a todos los que están en ese momento en la cola que enseñen su número y a los que está segura de que se han colado los expulsa de la tienda. Luego, cuando cierran la tienda, le cuenta a su padre a cuántos ha expulsado ese día.

## 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. En la primera aparece un único entero con el número de personas que están en la cola cuando Claudia hace la inspección (entre 1 y 500.000). En la segunda aparecen los números que tienen cada una de esas personas según están en la cola, números entre 1 y  $10^6$ , todos distintos (nadie ha llegado aún hasta el extremo de falsificar los números). Eso sí, pueden faltar números, porque hay personas que ya han sido atendidas o personas que se cansaron de esperar y se marcharon.

## Salida

Para cada caso de prueba, el programa escribirá el número de personas que Claudia ha expulsado.

### Entrada de ejemplo

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

### Salida de ejemplo

```
0
3
```





# Comienza la temporada

En el club deportivo del colegio les gusta aprovechar las equipaciones de los chicos de una temporada para la siguiente. De esta forma tienen más dinero para el mantenimiento de las instalaciones y no necesitan incrementar las cuotas. Al final de temporada, todos los jugadores deben devolver al club sus equipaciones, debidamente lavadas, planchadas y retirado el dorsal. Al comenzar la siguiente temporada el club toma nota de las tallas que necesitan los chicos y procede a repartir las equipaciones entre ellos, que volverán a poner el dorsal que les corresponda.



Al comenzar la temporada tenemos alumnos que se incorporan nuevos, en concreto los del primer curso de alevín, y normalmente los jugadores del curso anterior han cambiado de talla. Por este motivo es raro el año en que no hay que comprar alguna nueva equipación. Desde el club deportivo quieren optimizar sus recursos y nos han pedido que calculemos el mínimo número de equipaciones que han de comprarse para que todos los alumnos puedan jugar, teniendo en cuenta que si no hay suficientes equipaciones de una talla los alumnos pueden utilizar también las de una talla más. Irán un poco más holgados, pero no les impide moverse ni se les llega a caer como ocurriría con las tallas más pequeñas o las todavía más grandes.

## Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en que se indica el número  $N$  de jugadores de esta temporada y el número  $M$  de equipaciones con las que contamos ( $1 \leq N, M \leq 100.000$ ). A continuación aparece dos líneas: la primera tiene  $N$  valores con la talla que necesita cada chico y la segunda tiene  $M$  valores con las tallas de las equipaciones con las que contamos. Las tallas son valores entre 1 y 100.

## Salida

Para cada caso de prueba se escribe en una línea el mínimo número de equipaciones que debemos comprar.

## Entrada de ejemplo

```
5 4
12 12 12 12 12
13 14 12 10
3 4
13 10 12
15 11 12 14
3 3
10 14 17
6 4 50
```

## Salida de ejemplo

```
3
0
3
```





# Pepe Casanova

Pepe Casanova es un ligón de los de antaño, que intenta encandilar a las chicas con canciones románticas. A tal efecto, y de cara al veraneo en una playa del sur, decide conseguir una cinta para el radiocasete de su coche con las mejores canciones de amor.



Pepe es muy peculiar en sus gustos, y además anda algo escaso de dinero, por lo que en lugar de comprar una de tantas recopilaciones que circulan por el mercado discográfico, quiere grabársela él mismo. Rebuscando entre sus viejos vinilos, ha confeccionado una lista con sus canciones favoritas, apuntando la duración individual de cada una. Lamentablemente, su cinta de dos caras no tiene capacidad suficiente para contener todas las canciones, así que Pepe ha otorgado una puntuación a cada canción (cuanto más le gusta, mayor es la puntuación).

¿Puedes ayudar a Pepe a conseguir la mejor cinta (aquella cuya suma de puntuaciones de las canciones grabadas sea lo mayor posible), teniendo en cuenta que no puede repetir canciones, las escogidas han de caber enteras y no es admisible que una canción se corte a la mitad al final de una cara de la cinta?

## Entrada

La entrada está formada por una serie de casos de prueba. Cada caso comienza con el número  $N$  de canciones ( $1 \leq N \leq 30$ ) en la lista de Pepe. Después aparece la duración de cada una de las dos caras de la cinta (como mucho 1.000), y a continuación aparecen  $N$  líneas cada una con dos números enteros que representan la duración y la puntuación de cada una de las canciones. La puntuación dada nunca es mayor de 10.000.

La entrada termina con un 0.

## Salida

Para cada caso de prueba se escribirá una línea con la puntuación total máxima conseguida al grabar algunas canciones completas en las dos caras de la cinta.

## Entrada de ejemplo

```
4
90
50 80
40 20
40 50
60 10
0
```

## Salida de ejemplo

```
150
```

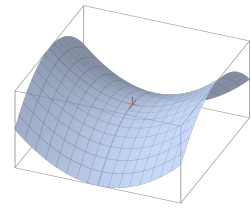




# Puntos de silla

Los *puntos de silla* de una matriz son aquellos elementos de la matriz que cumplen que son menores o iguales que el resto de elementos de su fila y mayores o iguales que el resto de elementos de su columna, o al revés, mayores o iguales que el resto de elementos de su fila y menores o iguales que el resto de elementos de su columna.

Reciben este nombre por su similitud con una silla de montar a caballo; aunque el caso de la silla de montar es mucho más particular (los elementos a partir del punto de silla están ordenados).



Dada una matriz, queremos saber si tiene puntos de silla o no.

## Entrada

La entrada está compuesta por diferentes casos de prueba. Cada caso comienza con una línea con dos enteros, el número  $F$  de filas y el número  $C$  de columnas de la matriz (entre 1 y 300). A continuación, aparecen  $F$  líneas cada una con  $C$  números enteros, que representan los valores de la matriz.

La entrada termina cuando el número de filas y columnas de un caso sea 0. Este último caso no debe ser procesado.

## Salida

Para cada caso de prueba se escribirá **SI** si la matriz tiene puntos de silla y **NO** en caso contrario.

## Entrada de ejemplo

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

## Salida de ejemplo

```
SI
NO
```







# Ratones en un laberinto

Un grupo de ratones de laboratorio está siendo entrenado para escapar de un laberinto. El laberinto está compuesto por una serie de celdas de las que salen pasadizos para unir las con otras. La longitud de cada pasadizo varía por lo que el tiempo que tardan los ratones en atravesar cada uno de ellos es distinto. Además, las puertecillas por las que los roedores acceden a los pasadizos se abren solo hacia dentro, por lo que se pueden utilizar únicamente en un sentido. Los investigadores, eso sí, a veces colocan dos pasadizos entre el mismo par de celdas, uno para cada sentido. En ese caso, aunque la longitud de cada uno sea la misma, el tiempo necesario para atravesarlos puede ser distinto por los obstáculos que se colocan dentro de los pasillos.



Todos los ratones han sido muy bien entrenados y, cuando son colocados en una celda arbitraria del laberinto, siguen un camino que los lleva a la celda de salida en un tiempo mínimo.

Vamos a realizar el siguiente experimento: se coloca un ratón en cada celda del laberinto (excepto en la celda de salida) y se inicia un temporizador de cuenta atrás. Cuando el cronómetro se detiene, contamos la cantidad de ratones que han salido del laberinto.

¿Puedes escribir un programa que, dada la descripción del laberinto y el límite de tiempo, prediga la cantidad de ratones que saldrán del laberinto? Puedes suponer que no hay cuellos de botella en el laberinto, es decir, que todas las celdas tienen espacio para un número arbitrario de ratones.

## Entrada

La entrada está compuesta por diversos casos de prueba. La primera línea de cada caso contiene 4 números: el número  $1 \leq N \leq 10.000$  de celdas del laberinto (numeradas de 1 a  $N$ ), el número  $S$  de la celda donde se encuentra la salida, el número  $T$  de segundos con el que se inicia el cronómetro para la cuenta atrás, y el número  $1 \leq P \leq 100.000$  de pasadizos. Las siguientes  $P$  líneas describen cada una de ellas un pasadizo, dando 3 números: dos números de celda distintos  $A$  y  $B$  y los segundos que tarda un ratón en llegar de  $A$  a  $B$  (un número entre 1 y 10.000).

Obsérvese que cada conexión es unidireccional, es decir, los ratones no pueden viajar de  $B$  a  $A$  a menos que haya otra línea que especifique ese pasadizo. Además, el tiempo requerido para viajar en cada dirección puede ser diferente.

## Salida

Para cada caso de prueba el programa debe escribir una línea con el número de ratones que alcanzarán la celda de salida  $S$  en como mucho  $T$  segundos.

## Entrada de ejemplo

```
5 5 20 5
1 2 5
1 4 10
2 4 7
3 4 15
4 5 10
3 1 10 2
2 3 5
3 2 6
```

## Salida de ejemplo

3 0
--------

## ● H

# ¿Cuándo seré rico?

Jaime ha cumplido 12 años y quiere tener una paga para poder gastársela en lo que quiera. Su padre es un importante empresario que se resiste a darle dinero: prefiere que Jaime le pida lo que quiere y si lo considera oportuno se lo compra. Pero cada vez que Jaime le pide una consola de videojuegos de última generación, su padre se la niega sistemáticamente.

Jaime quiere conseguir el dinero para la consola, así que ha ideado una estrategia para ir consiguiendo su dinero de manera paulatina. Justo antes de entrar a una reunión de alto nivel, le propone a su padre una forma de darle la paga que este acepta sin pensarlo mucho debido a las prisas: el primer día le dará un céntimo de euro, el segundo otro céntimo de euro y a partir del tercero siempre le dará el doble de lo que le dio dos días antes más lo que le dio justo el día antes. Jaime calcula cuántos días tardará en tener el dinero suficiente, pero... a veces se pregunta cuánto tardaría en hacerse millonario.



### Entrada

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

Cada caso de prueba representa una cantidad de céntimos (entre 1 y  $10^9$ ) que es el objetivo a conseguir por parte de Jaime.

### Salida

Para cada caso de prueba se escribirá en una línea el número de días que Jaime tarda en conseguir el dinero suficiente, suponiendo que es ahorrador y no se gasta nada hasta conseguir esa cantidad.

### Entrada de ejemplo

```
3
10
1000
10000
```

### Salida de ejemplo

```
4
11
14
```





# El carpintero Ebanisto

El carpintero *Ebanisto* ha recibido el encargo de cortar un tablón en varios trozos que han sido previamente marcados sobre la madera. El esfuerzo de cortar un tablón de madera en dos es el doble de su longitud.

Ebanisto se ha dado cuenta de que el orden en el que realice los cortes en el tablón influye en el esfuerzo empleado. Por ejemplo, supongamos que un tablón de 10 metros de longitud tiene que cortarse a 3, 6 y 8 metros de uno de los extremos. Una posibilidad sería cortar primero por la marca de los 3 metros, luego por la marca de los 6 metros y finalmente por la de 8 metros, lo que le costaría a Ebanisto un esfuerzo total de  $2 \times 10 + 2 \times 7 + 2 \times 4 = 42$ . Sin embargo, si corta primero por la marca del 6, después por la del 3 y finalmente por la del 8, entonces le costaría un esfuerzo de  $2 \times 10 + 2 \times 6 + 2 \times 4 = 40$ .



¿Puedes ayudar a Ebanisto a averiguar en qué orden cortar el tablón por las marcas para minimizar el esfuerzo realizado?

## Entrada

La entrada constará de varios casos de prueba. La primera línea de cada caso contendrá dos números positivos:  $L$  (entre 10 y 1.000.000), que representa la longitud del tablón que debemos cortar; y  $N$  (entre 1 y 500), que indica el número de cortes que se deben realizar. La siguiente línea contendrá  $N$  números positivos  $c_i$  ( $0 < c_i < L$ ), que determinan los puntos en los que se deben realizar los cortes, dados en orden creciente.

La entrada termina con 0 0.

## Salida

Para cada caso de prueba se debe escribir el mínimo esfuerzo que debe realizar Ebanisto para realizar todos los cortes.

## Entrada de ejemplo

```
10 3
3 6 8
20 4
8 10 15 17
0 0
```

## Salida de ejemplo

```
40
88
```