



“First, solve the problem. Then, write the code” John Johnson.

Programa-Me 2013

Regional on line

Problemas

6 de marzo de 2013



Universidad Complutense
de Madrid



I.E.S. Antonio de Nebrija
(Móstoles)

<http://www.programa-me.com>

Listado de problemas

A Nochevieja	3
B San Fermín	5
C ¿A dibujar hexágonos!	7
D ¿Es matriz identidad?	9
E Va de modas...	11
F Reloj a través del espejo	13
G ¿Cuál es la siguiente matrícula?	15
H Perímetro de un rectángulo	17
I Suma de dígitos	19
J Ascensor	21

Autores de los problemas:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Patricia Díaz García (I.E.S. Antonio de Nebrija - Móstoles)

A

Nochevieja

Ramón se pasa el día de Nochevieja contando los minutos que faltan para que den las uvas. ¿Puedes ayudarlo?

Entrada

La entrada consiste en una serie de horas, cada una en una línea. Cada hora está formada por las horas y los minutos separados por : y utilizando siempre dos dígitos. Se utiliza una representación en formato 24 horas (es decir, desde 00:00 a 23:59).

La entrada termina cuando la hora es la medianoche (00:00), que no debe procesarse.

Salida

Para cada caso de prueba se mostrará una línea con el número de minutos que faltan para medianoche.

Entrada de ejemplo

```
23:45
21:30
00:01
00:00
```

Salida de ejemplo

```
15
150
1439
```




San Fermines

Un buen corredor de San Fermines intenta mantener la emoción para fomentar el espectáculo. Este tipo de corredores está muy en forma y son capaces de correr mucho más deprisa que los toros. Sin embargo, para mantener la emoción, no lo hacen. Se limitan a mantener una velocidad adecuada para que los toros se mantenga siempre cerca, pero no lleguen a pillarles.

El problema consiste en, dadas las velocidades de todos los toros que participan en los San Fermines (y que se suponen constantes), ¿a qué velocidad deben ir estos corredores expertos para mantener el espectáculo conservando su integridad?

Entrada

La entrada consta de una serie de casos de prueba, cada uno en una línea. El primer número de la misma indica el número de toros que intervienen en la carrera ($n \geq 1$). A continuación aparece un entero por cada uno de los toros, indicando la velocidad a la que correrá ese toro (recuerda, es velocidad constante). La velocidad es siempre positiva y no excede 10^9 .

Salida

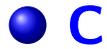
Para cada caso de prueba se mostrará una línea en la que aparecerá la velocidad a la que deberán ir los corredores expertos para mantener el espectáculo.

Entrada de ejemplo

```
7 1 9 8 7 10 3 12
1 10
```

Salida de ejemplo

```
12
10
```

¡A dibujar hexágonos!

Implementa un programa que, dado un entero n y un carácter c , dibuje el hexágono regular de n caracteres c de lado.

Entrada

La entrada consta de una serie de casos de prueba terminados con un 0 0. Cada caso de prueba es una línea con un entero mayor o igual que 0, seguido del carácter con el que se formará el hexágono.

Salida

Para cada caso de prueba se mostrará el hexágono regular que tiene como longitud de lado el número de caracteres especificado. Dicho hexágono estará dibujado con el carácter proporcionado en la entrada.

Entrada de ejemplo

```
3 *
4 p
0 0
```

Salida de ejemplo

```
***
*****
*****
*****
***
  pppp
 pppppp
pppppppp
pppppppppp
 pppppppp
  pppppp
   pppp
```




¿Es matriz identidad?

Se dice que una matriz es *identidad* cuando todos sus elementos son cero a excepción de la diagonal principal, que se encuentra rellena de unos:

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Para que una matriz sea identidad debe de ser *cuadrada*, es decir, tener el mismo número de filas que de columnas.

Entrada

La entrada consta de una serie de casos de prueba. Cada uno comienza con un número que representa el número de filas, como máximo 1000, de una matriz cuadrada. A continuación se dan los elementos que forman la matriz.

La entrada terminará con una matriz de 0 filas.

Salida

Para cada caso de prueba se indicará SI si la matriz es identidad y NO en caso contrario.

Entrada de ejemplo

```
3
1 0 0
0 1 0
0 0 1
2
0 1
1 0
5
1 0 0 0 0
0 5 0 0 0
0 0 1 0 0
0 0 0 1 0
0 0 0 0 1
0
```

Salida de ejemplo

```
SI
NO
NO
```




Va de modas...

Dado un conjunto, o distribución, de valores se define la *moda* como el valor (o valores) que más se repite en dicho conjunto.

En este problema te pedimos que calcules la moda de un conjunto de números.

Entrada

La entrada consta de una serie de casos de prueba.

Cada caso comienza con un número que representa el número de valores que tiene el conjunto, que nunca será mayor de 25.000. En la siguiente línea se proporcionan, separados por espacio, los valores de todos los elementos del conjunto. Todos ellos serán números enteros.

La entrada terminará con una serie de 0 valores.

Salida

Para cada caso de prueba se mostrará la moda de la distribución, es decir el número que más se repite.

Se garantiza que ningún caso de prueba contendrá más de una moda; es decir no habrá dos valores que aparezcan el mismo número de veces.

Entrada de ejemplo

```
11
1 2 2 3 3 3 4 4 4 4 5
17
1 8 9 6 3 2 1 5 4 7 9 6 3 2 1 4 7
0
```

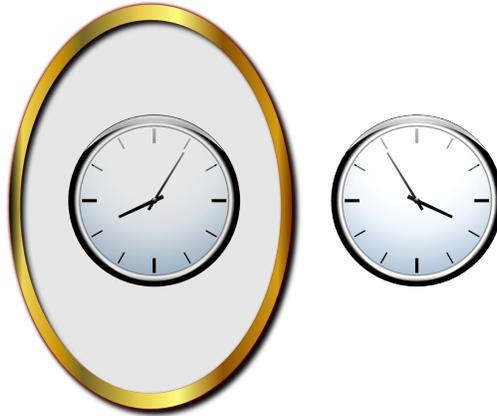
Salida de ejemplo

```
4
1
```




Reloj a través del espejo

Hoy te has despertado y al mirar, con legañas en los ojos, el reloj de agujas de tu habitación te has llevado un susto de muerte. ¡Las 8:05! ¡Ya llegas tarde!! Afortunadamente enseguida te has dado cuenta de que estabas viendo el reloj a través del espejo. Al mirar al reloj real, has visto que eran solo las 3:55. No sólo no llegas tarde, sino que todavía hay tiempo para otro buen sueñecito...



¿Eres capaz de hacer un programa que, dada una hora en el espejo, devuelva la hora real?

Entrada

La primera línea de la entrada contiene un entero que indica el número de horas que siguen. Cada una de las líneas siguientes especifica una hora vista en el espejo. La hora tendrá el formato HH:MM, y estará comprendida entre 01:00 y 12:59.

Salida

Para cada hora leída, el programa debe escribir, en una línea independiente, la hora real con el formato HH:MM, y también comprendida entre 01:00 y 12:59.

Entrada de ejemplo

```
3
08:05
12:00
10:09
```

Salida de ejemplo

```
07:45
12:00
01:51
```




¿Cual es la siguiente matrícula?

Las matrículas de los automóviles en España están compuestas por dos partes. La primera consiste en un número decimal de 4 dígitos que se va incrementando de uno en uno. Cuando se llega al último valor (9999), se reinicia la cuenta (en 0000) y se incrementa la segunda parte, que contiene letras.

Esta segunda parte consiste en tres letras consecutivas, excluyendo las vocales y la letra Ñ. Cuando, tras recorrer todos los números, se incrementa la letra, se pasa a la siguiente en el abecedario en la letra situada más a la derecha, saltándose las vocales y la Ñ. Si para ella se acaba el abecedario, se vuelve al principio (B) y se incrementa la anterior con este mismo procedimiento.

¿Eres capaz de pasar de una matrícula a la siguiente?

Entrada

La entrada consta de una serie de casos de prueba. Cada uno aparece en una línea, conteniendo una combinación de números y letras (en mayúscula) correspondiente a una matrícula. Consulta el ejemplo para más información.

El último caso de prueba, que no deberá procesarse, contendrá la última matrícula posible, 9999 ZZZ.

Salida

Para cada caso de prueba se mostrará una línea en la que aparecerá la matrícula que va a continuación de la dada, con las letras en mayúscula.

Entrada de ejemplo

```
1234 BBB
9999 BBZ
9999 BBD
9999 ZZZ
```

Salida de ejemplo

```
1235 BBB
0000 BCB
0000 BBF
```



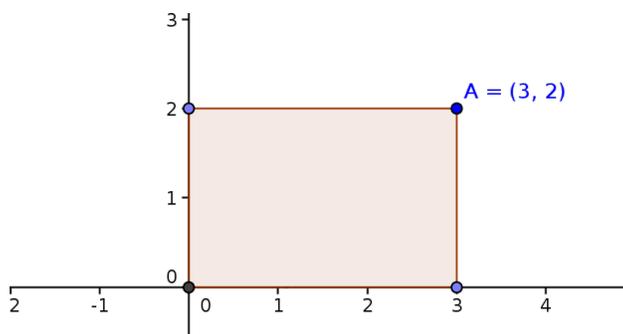

Perímetro de un rectángulo

El *perímetro* de un polígono es la suma de la longitud de todos sus lados. ¿Eres capaz calcular el perímetro de un rectángulo?

Entrada

La entrada consta de una serie de casos de prueba. Cada uno contiene, en una sola línea, la definición de un rectángulo.

Un rectángulo se especifica proporcionando las coordenadas de la esquina superior derecha. Se debe considerar que la esquina inferior izquierda está siempre en el origen de coordenadas, es decir en $(0, 0)$. Se garantiza que la longitud de un lado no superará 10^8 .



El último caso de prueba, que no debe procesarse, contendrá un valor negativo en alguno de los ejes.

Salida

Para cada caso de prueba se escribirá, en una línea independiente, el parámetro del rectángulo.

Entrada de ejemplo

```
3 2
5 4
-9 3
```

Salida de ejemplo

```
10
18
```




Suma de dígitos

¿Cuál es la suma de los dígitos de un número entero no negativo dado? Por ejemplo, la suma de los dígitos del 3433 es 13.

Para darle un poco más de emoción, el programa no se limitará a escribir el resultado de la suma, sino que también escribirá todos los sumandos utilizados: $3 + 4 + 3 + 3 = 13$.

Entrada

La entrada consta de una serie de casos de prueba terminados con un número negativo. Cada caso de prueba es una simple línea con un número no negativo no mayor que 10^9 del que habrá que sumar todos sus dígitos. Ningún número contendrá ceros supérfluos a la izquierda.

Salida

Para cada caso de prueba se mostrará una línea en la que aparecerán cada uno de los dígitos separados por el signo +, tras lo cual aparecerá el símbolo = y la suma de todos los dígitos.

Ten en cuenta que antes y después de cada símbolo (+ y =) hay un espacio.

Entrada de ejemplo

```
3433
7
23
-1
```

Salida de ejemplo

```
3 + 4 + 3 + 3 = 13
7 = 7
2 + 3 = 5
```




Ascensor

Las máquinas son incansables; una vez puestas en marcha, estarán funcionando sin quejarse por muchas veces que les pidas hacer lo mismo.

Pensemos en un ascensor, por ejemplo. Empieza el día en el bajo; el del primero le llama y le hace subir al quinto. Justo después le llaman en el segundo y le toca bajar a la planta baja. Luego el del tercero quiere ir a ver al del noveno... Al final del día ha hecho un largo recorrido de arriba a abajo, y de abajo a arriba. ¿Cuál es la longitud de ese recorrido?

Entrada

La entrada contiene distintos casos de prueba, cada uno en una línea. En cada una de ellas hay una secuencia de enteros. El primero marca el piso inicial en el que empieza el ascensor (el 0 marca la planta baja; no hay plantas por debajo de ella). A continuación aparecen parejas de enteros, cada uno de ellos representando el uso del ascensor por parte de un vecino, con el piso desde el que llama al ascensor y con el piso destino. La lista termina con un -1.

La entrada acaba con una línea en la que el ascensor comienza por debajo de la planta baja.

Salida

Para cada caso de prueba se mostrará una línea en la que aparecerá la longitud (en número de pisos) del recorrido completo del ascensor a lo largo del día.

Entrada de ejemplo

```
0 1 5 2 0 3 9 -1
5 5 4 -1
1 2 3 4 5 -1
-1
```

Salida de ejemplo

```
19
1
4
```