



“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 ¿Cuántos días faltan?	3
B Los saltos de Mario	5
C Viendo pasar el tiempo	7
D Matrices Triangulares	9
E Cálculo de la Mediana	11
F Tableros de Ajedrez	13
G Súmale uno	15
H Área de un rectángulo	17
I Número hyperpar	19
J Zapping	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)



¿Cuántos días faltan?

Dado un día del año, ¿sabrías decir cuántos días faltan para Nochevieja? Asumiremos que nos encontramos en un año *no* bisiesto.

Entrada

La entrada comenzará con un entero para especificar el número de casos de prueba que vendrá a continuación. Para cada caso de prueba se mostrará una línea en la que aparecerán dos enteros, el primero de ellos será correspondiente al día y el segundo correspondiente al mes de una fecha válida.

Salida

Para cada fecha de la entrada, se mostrarán el número de días completos que faltan hasta el día de Nochevieja.

Entrada de ejemplo

```
3
28 12
30 12
1 1
```

Salida de ejemplo

```
3
1
364
```




Los saltos de Mario

Mario se encuentra ante el castillo final. Puede verlo desde lo alto del muro en el que se encuentra. En breve podrá entrar en la Cámara de Koopa, enfrentarse (y vencer) al monstruo final y salvar a la princesa.

Sin embargo, tiene ante sí una serie de muros que tendrá que sobrepasar. Para eso, saltará desde el primero de ellos, donde se encuentra, al siguiente, y desde él al siguiente, y así sucesivamente hasta llegar al último.

La pregunta que nos hacemos es, ¿cuántos de estos saltos serán hacia arriba y cuántos hacia abajo? Mario realiza un salto hacia arriba cuando tiene que alcanzar un muro que está por encima de él, y hacia abajo cuando tiene que alcanzar un muro que está por debajo.

Entrada

El primer valor de la entrada es un número que indica la cantidad de casos de prueba a evaluar. Cada caso de prueba comienza con un entero mayor que cero y no mayor que 10^9 que indica el número de muros del escenario (recuerda que Mario se encuentra situado en la parte de arriba del primero). A continuación se proporciona la serie de enteros que indican la altura cada uno de ellos.

Salida

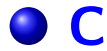
Para cada caso de prueba se mostrará una línea en la que aparecerán dos enteros, uno con los saltos hacia arriba y otro con los altos hacia abajo, separados por un espacio.

Entrada de ejemplo

```
3
8
1 4 2 2 3 5 3 4
2
9 9
5
1 2 3 4 5
```

Salida de ejemplo

```
4 2
0 0
4 0
```

Viendo pasar el tiempo

En un reloj digital de 24 horas que muestra dos dígitos para las horas y dos dígitos para los minutos, ¿durante cuántos minutos al día se puede ver algún 0? ¿Y algún 2? ¿Y entre las 00:30 y las 13:41?

Entrada

Cada caso de prueba consiste una línea en la que aparecen dos horas y un número de un sólo dígito. Las horas aparecerán expresadas de la misma forma que se muestran en un reloj digital configurado con el formato de 24 horas: dos dígitos para las horas y dos dígitos para los minutos separados por `:`. Tras las dos horas (que aparecen separadas por espacios) aparece un número $0 \leq n \leq 9$ que representa el dígito por el que nos preguntan.

Las horas estarán en el intervalo 00:00 y 23:59 y la segunda hora nunca será anterior a la primera.

La entrada termina con una línea con 24:00 24:00, que no debe procesarse.

Salida

Para cada caso de prueba se mostrará, en una línea independiente, el número de minutos en los que se puede ver el dígito n entre las dos horas indicadas.

Entrada de ejemplo

```
00:00 00:02 0
00:00 00:02 1
00:00 00:02 2
12:59 13:01 0
24:00 24:00 0
```

Salida de ejemplo

```
3
1
1
2
```




Matrices Triangulares

Se dice que una matriz cuadrada, es decir que tiene el mismo número de filas que de columnas, es *triangular* cuando *todos* los valores que están por encima o por debajo de la diagonal principal son cero. También son triangulares aquellas matrices que cumplen estas dos condiciones a la vez.

$$I_3 = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{bmatrix}$$

Realiza un programa que diga si una matriz cuadrada dada es o no triangular.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con un número que representa el número de filas, mayor que cero y menor que 1000, de la matriz cuadrada. A continuación se dan los elementos que forman la matriz.

La entrada terminará con una matriz de 0 filas.

Salida

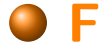
Para cada caso de prueba se indicará SI si la matriz es triangular y NO en caso contrario.

Entrada de ejemplo

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

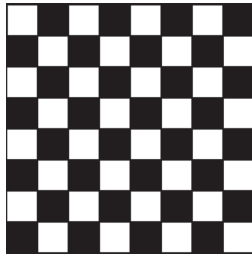
Salida de ejemplo

```
SI
SI
NO
```

Tableros de Ajedrez

Un *escaque* es cada una de las casillas cuadradas e iguales en las que se divide un tablero de ajedrez. Estos tableros están formado por 64 escaques, de los cuales 32 son claros y los otros 32 son oscuros (a menudo blancos y negros), alternándose en su disposición.



¿Eres capaz de pintar un tablero así?

Entrada

La entrada consta de una serie de casos de prueba terminados en 0. Cada caso de prueba es línea que nos indica el tamaño de cada escaque y el caracter por el que están formados los escaques negros. Los escaques blancos siempre estarán formados por espacios.

Salida

Para cada caso de prueba se mostrará el tablero creado siguiendo la configuración dada en la entrada.

Entrada de ejemplo

```
1 o
2 #
0
```

Salida de ejemplo

```
|-----|
| o o o o |
| o o o o |
| o o o o |
| o o o o |
| o o o o |
| o o o o |
| o o o o |
| o o o o |
|-----|
```

```
|-----|
|  ##  ##  ##  ## |
|  ##  ##  ##  ## |
|##  ##  ##  ## |
|##  ##  ##  ## |
|  ##  ##  ##  ## |
|  ##  ##  ##  ## |
|##  ##  ##  ## |
|##  ##  ##  ## |
|  ##  ##  ##  ## |
|  ##  ##  ##  ## |
|##  ##  ##  ## |
|##  ##  ##  ## |
|  ##  ##  ##  ## |
|  ##  ##  ##  ## |
|##  ##  ##  ## |
|##  ##  ##  ## |
|-----|
```




Súmale uno

Cuando escribimos un número en decimal, usamos como dígitos los valores del 0 al 9. La posición de cada dígito indica el *peso* que tiene en el valor total del número.

Los números escritos en hexadecimal, por su parte, utilizan dígitos del 0 al 15. Para representar los que superan el 9, necesitamos *símbolos* nuevos. Para eso, se utilizan las letras de la A a la F, de manera que, en hexadecimal, el número 9 es seguido por el A, o el BF por C0.

¿Eres capaz de dar el siguiente número hexadecimal a uno dado?

Entrada

La entrada estará compuesta de múltiples casos de prueba. Cada uno consistirá en un número escrito en notación hexadecimal, hasta un máximo de 100 dígitos. Ninguno de los casos de prueba contendrá ceros supérfluos a la izquierda, y usarán, en su caso, letras mayúsculas.

Un caso de prueba con la cadena FIN marca el final.

Salida

Para cada caso de prueba se mostrará una línea en la que aparecerá el número siguiente al dado, escrito en notación hexadecimal sin ceros supérfluos a la izquierda y con las letras en mayúscula.

Entrada de ejemplo

```
9
12
FF
FIN
```

Salida de ejemplo

```
A
13
100
```

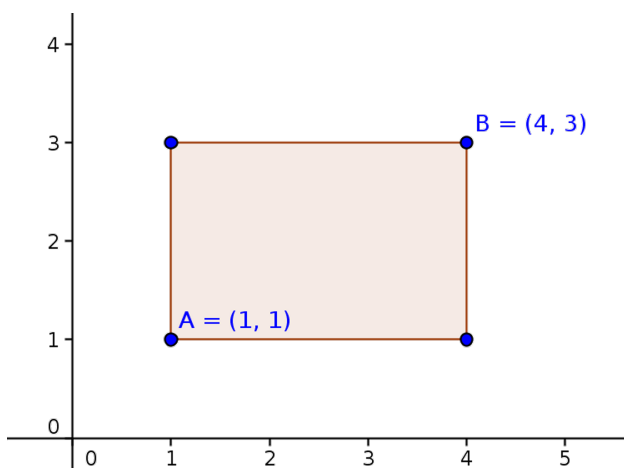



Área de un rectángulo

El área de un polígono es la cantidad de superficie que encierran sus lados. El área de un rectángulo es la multiplicación de la base por la altura. ¿Sabes calcularla?

Entrada

La entrada consta de una serie de casos de prueba. Cada uno contiene la definición de un rectángulo en una sola línea con cuatro números enteros no negativos menores o iguales que 10.000. Los dos primeros definen la posición del vértice inferior izquierdo del rectángulo y los dos siguientes la del vértice superior derecho.



El último caso de prueba, que no debe procesarse, proporciona un rectángulo en el que el segundo vértice está debajo o a la izquierda del primero.

Salida

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

Entrada de ejemplo

```
1 1 4 3
0 0 1 1
9 7 3 6
```

Salida de ejemplo

```
6
1
```




Número hyperpar

Se dice que un número es *hyperpar* cuando todos sus dígitos son pares. ¿Sabes identificarlos?

Entrada

La entrada consta de una serie de casos de prueba. Cada uno está compuesto de una única línea con un número no negativo ni mayor que 10^9 .

Los casos de prueba terminan con un número negativo que no habrá que procesar.

Salida

Para cada caso de prueba se escribirá, en una línea independiente, SI si el número es hyperpar y NO si no lo es.

Entrada de ejemplo

```
2460
1234
2
-1
```

Salida de ejemplo

```
SI
NO
SI
```


J Zapping

Mi tele tiene 99 canales correlativos que van desde el canal número 1 al 99. Hace algún tiempo se me estropeó el mando a distancia, y no me funcionan los números, por lo que no puedo acceder a mi canal favorito con ellos. Al menos, todavía funcionan los botones de pasar al siguiente canal y al anterior. No quiero que también estos botones se me rompan, de modo que siempre intento pulsarlos el menor número de veces.

Sabiendo que al pasar al siguiente canal desde el 99 la tele me salta al 1, y viceversa, ¿cuántas veces tengo que avanzar o retroceder de canal para pasar de un canal A a un canal B de forma óptima?

Entrada

La entrada consta de una serie de casos de prueba consistentes en dos valores entre 1 y 99, el primero correspondiente al canal en el que me encuentro y el segundo al canal al que quiero llegar. La entrada terminará cuando los dos números sean 0.

Salida

Para cada caso de prueba se escribirá una línea en la que aparecerá el número de veces que tengo que saltar de canal para conseguir mi objetivo de forma óptima.

Entrada de ejemplo

```
2 5
2 99
5 63
63 5
0 0
```

Salida de ejemplo

```
3
2
41
41
```