



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

# Programa-Me 2013

## Regional de Madrid y Reus

### Problemas

Ejercicios realizados por



Facultad  
de  
Informática  
Universidad Complutense  
de Madrid



I.E.S. Antonio de Nebrija  
(Móstoles)

Realizado en el **I.E.S. El Lago (Madrid)** y en el **I.S. Baix Camp (Reus)**  
14 de marzo de 2013





## Listado de problemas

<b>A Pintando fractales</b>	<b>3</b>
<b>B La pieza perdida</b>	<b>5</b>
<b>C Carrera popular</b>	<b>7</b>
<b>D Triángulos con piedras</b>	<b>9</b>
<b>E Abadías pirenaicas</b>	<b>11</b>
<b>F El pan en las bodas</b>	<b>13</b>
<b>G Partido de squash</b>	<b>15</b>
<b>H Año 2013</b>	<b>17</b>
<b>I Manías tenemos todos</b>	<b>19</b>
<b>J Campo de minas</b>	<b>21</b>

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)

Revisores:

- Ferran Borrell Micola (I.S. Baix Camp - Reus)
- Cristina Gómez Alonso (I.S. Baix Camp - Reus)
- Roger Meix Mañá (I.S. Baix Camp - Reus)
- Catalina Molano Alvarado (I.E.S. Antonio de Nebrija - Móstoles)





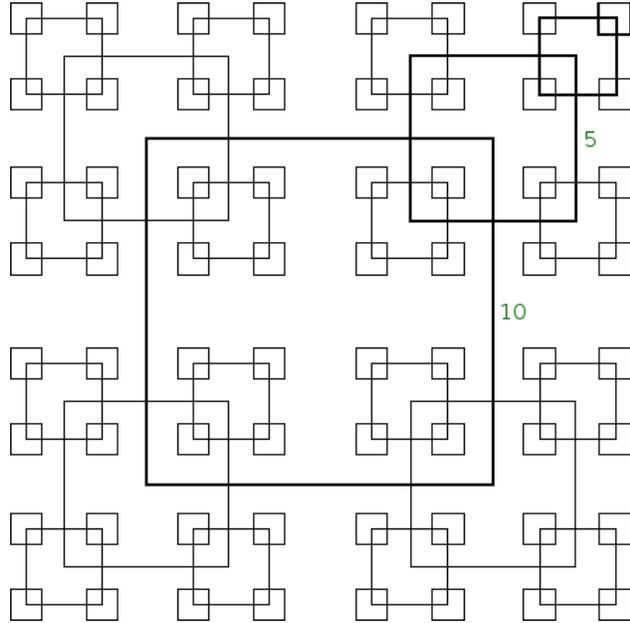
# Pintando fractales

Un *fractal* es un objeto geométrico cuya estructura básica se repite a diferentes escalas. Los fractales, conocidos por los matemáticos desde principios del siglo XX, llegaron al gran público con el auge de los ordenadores pues permiten generar figuras vistosas utilizando fórmulas matemáticas.

La figura que hoy nos planteamos, no obstante, no es de una gran vistosidad. Consiste en un simple cuadrado de longitud  $l$  cuyas cuatro esquinas son el centro de otros tantos cuadrados de longitud  $l/2$ . Las esquinas de cada uno de ellos, a su vez, son el centro de otros cuatro cuadrados con longitud  $l/4$ , y así sucesivamente hasta llegar a cuadrados de longitud 1. La figura se construye de tal forma que las longitudes *son siempre enteras*, por lo que si empezamos con un cuadrado de longitud, pongamos, 5, los cuadrados siguientes serán de longitud 2.

La imagen muestra la figura generada si empezamos con un cuadrado cuyo lado tiene longitud 10. Los cuadrados en sus esquinas tienen longitud 5, los siguientes tienen longitud 2 para terminar con los cuadrados más pequeños con lados de longitud 1.

La pregunta que nos hacemos es la cantidad de tinta que necesitaremos para pintar la figura dada la longitud del cuadrado más grande. O, dicho de otra forma, cuál es la suma de las longitudes de los lados de todos los cuadrados.



## Entrada

La entrada contiene varios casos de prueba. Cada uno aparece en una línea independiente que contiene un número con la longitud de los lados del cuadrado más grande que hay que pintar.

## Salida

Para cada caso de prueba se deberá escribir la suma de las longitudes de los lados de todos los cuadrados que forman la figura. Se garantiza que esa longitud será menor que  $2^{31}$ .

## Entrada de ejemplo

```
1
3
5
```

## Salida de ejemplo

```
4
28
116
```



## ● B

# La pieza perdida

Andrés no da crédito. Se ha encontrado en la calle una bolsa llena de piezas de un puzzle. Lo peculiar del caso es que las piezas *están numeradas* por detrás, lo que implica que su anterior propietario tuvo el puzzle montado y lo desmontó etiquetando pacientemente cada ficha para hacer mucho más fácil el siguiente montaje. Tras un rato se da cuenta de por qué, tras todo ese esfuerzo, decidió deshacerse del puzzle: en la misma bolsa había una nota que decía “*Falta una pieza*”.



Lo que Andrés se pregunta ahora es ¿qué pieza es la que falta?

### Entrada

La entrada estará compuesta de numerosos casos de prueba. Cada uno de ellos contiene la información de *un puzzle* en dos líneas. La primera línea contiene el número de piezas *totales* del puzzle (al menos dos y como mucho 10.000), y la segunda contiene los números de todas las piezas que había en la bolsa (no necesariamente en orden). La numeración de las piezas comienza por el número 1.

La entrada terminará con un puzzle de cero piezas, que no generará salida.

### Salida

Para cada caso de prueba se debe escribir una línea con el número de la pieza que faltaba en la bolsa.

### Entrada de ejemplo

```
10
1 2 3 5 6 7 8 9 10
8
6 4 3 2 1 8 7
0
```

### Salida de ejemplo

```
4
5
```





# Carrera popular

Adolfín Folk participa hoy en una carrera popular benéfica. En la zona de la salida están los listados de los nombres de los participantes, con el número de dorsal al lado. Tiene que buscarse, para saber en qué cola ponerse a esperar para que le den su dorsal.

Mirando la lista (¡¡es una lista enorme!!) se ha dado cuenta de que es fácil identificar a los participantes que son hermanos, dado que los nombres aparecen ordenados por apellido.

## Entrada

La entrada comenzará con una línea conteniendo únicamente un número con la cantidad de casos de prueba. A continuación aparecerán dichos casos de prueba.

Cada uno será una lista de participantes, con un participante por línea con el formato “Apellido, Nombre” (sin las comillas), y ordenados por apellido. El número de participantes es arbitrariamente grande, pero ninguna línea tendrá más de 40 caracteres. Además, el uso de las mayúsculas en el texto es errático, por lo que debe ignorarse.

Cada caso de prueba acaba con la cadena “====” (cuatro símbolos de igual consecutivos, sin las comillas).

## Salida

Para cada caso de prueba se debe indicar, en la misma línea, el número de participantes en la carrera y cuántos de ellos tienen al menos otro hermano inscrito. Ambos números se separarán por un espacio.

## Entrada de ejemplo

```
2
Folk, Adolfin
Zapatilla, Zape
ZAPATILLA, ZIPI
====
Bunny, Bugs
Folk, Adolfin
Mouse, Mickey
====
```

## Salida de ejemplo

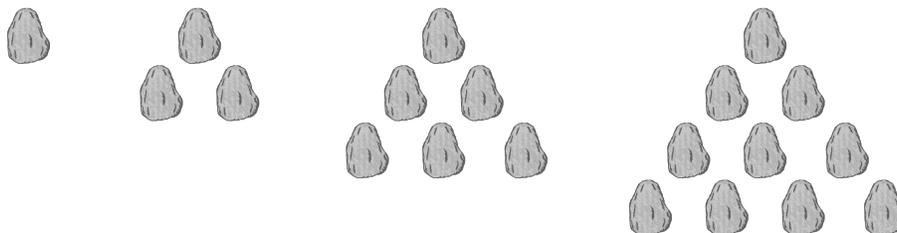
```
3 2
3 0
```





# Triángulos con piedras

En su última clase, Dianthe aprendió lo que son los números triangulares. “Si tenéis un número triangular de piedrecitas — les decía Pitágoras — podréis formar un triángulo con el mismo número de piedras en cada lado”. Cogiendo pequeñas piedras del suelo, les hizo una demostración:



Con una sola piedra, se puede formar un triángulo de lado 1. Con 3, se puede formar un triángulo de lado 2. Serán necesarias 10 piedras para formar un triángulo de lado 4. En cada paso, se suma una piedra más a las que se añadieron en el paso anterior. Es decir, primero se pone una piedra, luego dos más, luego tres más, y así sucesivamente.

Dianthe se pregunta de qué tamaño será el triángulo más grande que puede formar así, si tiene 1000 piedras (aunque es posible que la sobren algunas). ¿Puedes ayudarla?

## Entrada

La entrada estará compuesta por múltiples casos de prueba. Cada uno contendrá un único número en una línea, indicando el número de piedras que tiene Dianthe (hasta 250.000.000).

La entrada terminará cuando el valor sea 0, que no deberá procesarse.

## Salida

Para cada caso de prueba se debe indicar el tamaño de los lados del triángulo más grande que se puede formar con las piedras disponibles, así como el número de piedras que sobrarán.

## Entrada de ejemplo

```
1
6
13
0
```

## Salida de ejemplo

```
1 0
3 0
4 3
```



# ● E

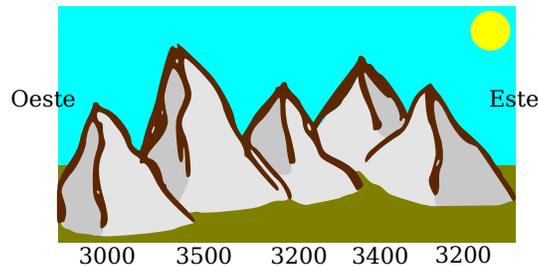
## Abadías pirenaicas



Es bien sabido que los monjes, en las abadías, se levantan muy temprano para rezar durante *maitines*, observando la salida del sol.

La localización de las abadías y monasterios es por tanto de vital importancia para que se pueda observar el sol aparecer en el horizonte. Los monjes prefieren las localizaciones que no tengan montañas altas hacia el Este, que les impedirían ver salir el sol.

Esto es un problema especial en los Pirineos, la cordillera montañosa que va de Oeste a Este y separa la Península Ibérica del resto del continente europeo. Los Pirineos están llenos de lugares tranquilos y mágicos, por lo que son emblemáticos para la meditación y el rezo. Pero al estar orientados de Oeste a Este, tienen una limitación en el número de localizaciones candidatas a construir abadías, es decir las que reciben el sol en los primeros momentos de la mañana. Este problema lo sufren también otras cordilleras en todo el mundo.



### Entrada

La entrada estará compuesta de múltiples casos de prueba.

Cada caso de prueba comenzará con un número indicando el número de montañas de una determinada cordillera montañosa que avanza de Oeste a Este. A continuación se indicarán las alturas (en metros, sin decimales) de cada una de esas montañas, comenzando con la situada más al Oeste. Ninguna cordillera tendrá más de 20.000 montañas.

La entrada terminará con una cordillera sin montañas, que no debe procesarse.

### Salida

Para cada caso de prueba se debe indicar el número máximo de abadías que se pueden construir en esa cordillera montañosa. Sólo se pueden construir abadías en las cimas que *sean más altas* que todas las montañas que tengan al Este.

### Entrada de ejemplo

```
5 3000 3500 3200 3400 3200
4 4000 3500 3500 3200
0
```

### Salida de ejemplo

```
3
3
```

### Fuente

Basado en el problema 4.11 de “*Algoritmos correctos y eficientes*”, Narciso Martí Oliet, Clara Segura, y Alberto Verdejo, Garceta, 2012.



## ● F

# El pan en las bodas

A Jack Dauson le cuesta mucho aprenderse las normas de etiqueta cuando se sienta en una mesa. ¿Cuál es el cuchillo de la carne? ¿Y el tenedor del pescado? Pero lo que más le incomoda es no saber cuál es el pan que le pertenece. Cuando se sienta a la mesa se encuentra que tanto a la izquierda como a la derecha hay un platito con una barrita de pan. Uno de los dos es el suyo, y el otro es el de la persona que se sienta a su lado pero... ¿a qué lado? Si el suyo es el de la izquierda, el de la derecha será para el comensal que se sienta a la derecha. Si la barrita que hay a la derecha es la suya, será justo al contrario.



Normalmente lo que hace es esperar a que algún otro comensal coma algo de pan para deducir cuál es el suyo y así no confundirse (o al menos poder echar la culpa a otro sobre su confusión). Y es que es importante no confundirse de lado porque es posible que eso provoque que haya alguien que se quede sin comer pan.

Hoy Jack está de invitado en una boda donde hay mesas redondas. Cuando se acerca a la suya se da cuenta de que ya hay varias personas que han empezado a comer barritas. ¿Podrán comer pan todos, o hay ya alguien que no podrá porque sus dos vecinos se comerán las dos barritas que él tenía a sus dos lados?

## Entrada

La entrada contiene distintos casos de prueba. Cada uno de ellos contiene, en una línea, la descripción de la configuración (válida) de una mesa. La línea comienza con un número positivo que indica el número de asientos (como mucho 1000 sillas). Tras un espacio, aparece un carácter por cada una de las sillas que hay en la mesa. El carácter I indica que en esa silla el comensal ha empezado la barrita de su izquierda; una D indica que ha comido de la barrita de la derecha. Por último un . indica que en esa silla aún no se ha sentado nadie (o que, quién lo ha hecho, aún no ha probado el pan). Los invitados a la boda son personas decentes y no se roban el pan los unos a los otros, por lo que si un comensal ya ha empezado una barrita, la persona de al lado no la probará.

Ten en cuenta que la mesa es circular, por lo tanto se considera que el último carácter tiene a su derecha el primero.

El último caso de prueba, que no deberá ser procesado, contiene una mesa sin sillas.

## Salida

Para cada caso de prueba se mostrará una única línea indicando si, una vez llena la mesa, todos los comensales podrán comer pan (TODOS COMEN) o hay al menos uno que no podrá (ALGUNO NO COME).

## Entrada de ejemplo

```
6 .I....
6 .D.I..
4 ....
0
```

## Salida de ejemplo

```
TODOS COMEN
ALGUNO NO COME
TODOS COMEN
```





# Partido de squash

Es curioso esto de los marcadores en los deportes. En deportes de equipo, como el fútbol o el baloncesto, el que marca el último gol o la última canasta no tiene por qué ser el ganador. Para saber quién gana hay que seguir el sencillo procedimiento de sumar todos los tantos y el equipo que más haya conseguido es el vencedor.



Sin embargo, en muchos deportes individuales como el tenis, frontón o squash es fácil saber quién gana. Basta con ver el último tanto del partido y el que lo gane es el que se llevó la victoria final. Sin embargo recomponer el marcador es más difícil.

Pongamos el caso del squash. Aunque hay distintas formas de puntuar, una de las más utilizadas por los jugadores *amateur* es jugar al mejor de cinco *sets*. El jugador que antes gane tres sets gana el partido (y en ese momento, obviamente, éste termina). Los sets se juegan a 9 puntos con tres matizaciones:

- Es necesario ganar con una ventaja de al menos dos puntos, por lo tanto, si se llega a un empate a 8, ganará el que llegue a 10. Si al intentarlo se empata a 9, se jugará a 11, etc.
- Los puntos sólo suben al marcador si el que ganó el tanto *tenía el saque*. Es decir, se utiliza lo que se conoce como “recuperación del saque”. Si el jugador A saca y gana el punto, éste sube a su marcador. Pero si el tanto lo gana B, dado que no fue él el que puso la bola en juego, recupera el saque pero no puntúa.
- El jugador que gana un *set* comienza sacando en el siguiente.

Nos piden averiguar el marcador final de un partido de squash conociendo los ganadores de todos los puntos. Es posible que el partido no termine debido a que el alquiler de la pista llega a su fin. En ese caso se presentará el resultado parcial.

## Entrada

La entrada contendrá una sucesión de partidos de squash. Cada partido aparecerá en una línea distinta y será una sucesión de A's y B's que indicarán que el jugador A o el jugador B ganó ese tanto. Todos los partidos terminarán con el carácter F que indica que, o bien el partido terminó con un ganador, o bien dejaron de jugar debido a que terminó el alquiler de la pista.

Asume que en todos los partidos empieza sacando el jugador A.

La entrada termina con un partido sin tantos (es decir únicamente con una F), partido que no deberá ser procesado.

## Salida

Para cada partido muestra su resultado con el formato que aparece en el ejemplo. Para cada set empezado aparecerá los puntos conseguidos por el jugador A y tras un - los puntos conseguidos por el jugador B. Separa el resultado de cada set con un espacio. *No* añadas un espacio tras el último set.

## Entrada de ejemplo

```

AAAAAAAAAF
BBBBBBBBBF
AAAAAAAAABBBBBBBBBBAABBBABBF
BF
F

```

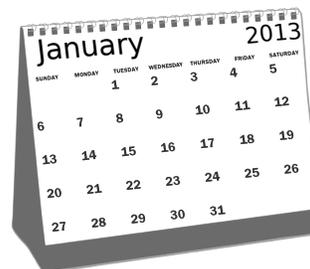
## Salida de ejemplo

```
9-0  
0-9  
9-11 0-1  
0-0
```

# ● H Año 2013

Mary lleva muchos años redactando cartas para sus jefes. Cuando comenzó el año 2013, observó con fastidio que al poner la fecha se veía obligada a utilizar más teclas diferentes que otras veces. Empezó a hacer memoria y se dio cuenta de que, tras 25 años, el año 2013 era el primero que tenía sus cuatro dígitos diferentes. Desde 1988, todos los años habían tenido al menos un dígito repetido.

La *serie* de años *sin* dígitos repetidos que comienza con 2013 terminará en 2019. En 2020 comenzará una serie nueva de números *con* dígitos repetidos, que se mantendrá hasta el 2030, incluido.



## Entrada

La entrada comenzará con una única línea indicando el número de casos de prueba. Cada uno consistirá en un número positivo (año) entre 1.000 y 1.000.000.

## Salida

Para cada caso de prueba se debe indicar el primer año y la longitud de *la serie* de números con o sin dígitos repetidos a la que pertenece.

## Entrada de ejemplo

```
3
1990
2015
2025
```

## Salida de ejemplo

```
1988 25
2013 7
2020 11
```





# Manías tenemos todos

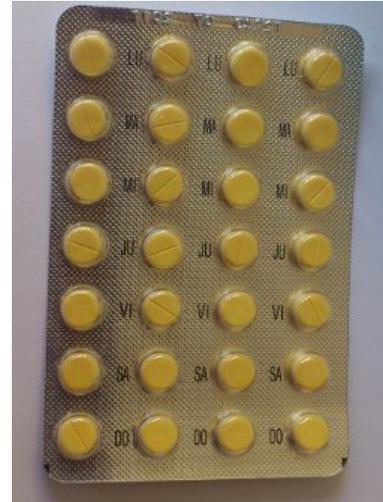
Todos dicen que Mayte es una maniática... y ella no entiende por qué. Al fin y al cabo todos tenemos nuestras rarezas...

Una de las de Mayte tiene que ver con las pastillas. Cuando el médico le pone un tratamiento, ella pide en la farmacia que sean de un laboratorio que las envase con el día de la semana impreso. De esta manera puede controlar si algún día se le ha olvidado tomársela.

Hasta aquí su familia y amigos consideran que todo es relativamente normal. Lo que ya no lo es tanto es que si un día se le olvida tomar su medicación, se “salte” la pastilla de ese día y no se la tome. Dice que es incapaz de tomar una pastilla en un día de la semana distinto del que indica la caja. Cuando se salta una toma, sencillamente abandona indefinidamente la pastilla; ni siquiera la utiliza el mismo día de la semana siguiente. Eso hace que a día de hoy tenga una buena colección de pastillas dispersas en distintas cajas, sólo porque se le olvidó tomarlas y su manía le impide aprovecharlas.

Pero eso está a punto de cambiar. Recientemente ha decidido que, si finalmente aplican la medida del llamado *euro por receta*, recuperará todas las pastillas descartadas para ahorrarse algo de dinero. Eso sí, no está dispuesta a romper su hábito, por lo que seguirá tomándose únicamente las pastillas que estén etiquetadas con el día de la semana que le corresponda.

Suponiendo que comienza a *reciclar* pastillas el próximo lunes, ¿puedes ayudar a Mayte a saber que día tendrá que volver a la farmacia a comprar pastillas?



Blister de pastillas de Mayte

## Entrada

La primera línea de la entrada contendrá el número de casos de prueba que vienen a continuación.

Cada caso de prueba comienza con un número que indica la cantidad de *semanas* que Mayte lleva guardando pastillas. A continuación, se muestra, en líneas independientes, el estado de las pastillas correspondientes a cada semana. Comenzando por la del lunes, para cada una de las 7 pastillas se mostrará un \* si Mayte *no* se tomó esa pastilla, o un - si lo hizo. Por ejemplo, si Mayte se ha tomado las pastillas del martes, miércoles, viernes y sábado la entrada será “\*--\*--\*”.

## Salida

Para cada caso de prueba se debe indicar el día de la semana en el que Mayte deberá empezar una caja no usada de pastillas, y en qué semana se está (empezando a contar por 1). Recuerda que Mayte comienza en lunes la medida de reciclar pastillas, y que nunca se toma una pastilla un día de la semana diferente a aquél con el que está etiquetada.

En la salida, los días de la semana se representarán con una única letra, escribiendo L para lunes, M para martes, X para miércoles, J para jueves, V para viernes, S para sábado y D para domingo.

## Entrada de ejemplo

```
2
4
*****
*--*--*
*-*-*-*
--*---*
3
-----
*****-*
*****
```

## Salida de ejemplo

```
M 2
S 2
```



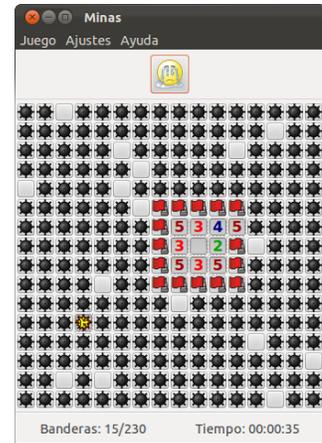
# Campo de minas

Aunque existieron antecedentes previos, el juego *Buscaminas* comenzó a ser famoso cuando se incluyó en la versión 3.1 de Windows, en el lejano 1992. Corre el rumor de que fue incluido para que los usuarios se entretuvieran mientras, sin saberlo, cogían práctica utilizando un dispositivo por aquel tiempo apenas conocido: el ratón.

Hoy el juego es archiconocido (y el dispositivo también). El número de variantes e implementaciones es inmenso, y sigue estando incluido en la gran mayoría de los sistemas de escritorio.

El juego consiste en un tablero rectangular con celdas, algunas de las cuales ocultan *minas*. El usuario debe ir destapando las celdas con cuidado para no seleccionar ninguna mina. Como ayuda, cada vez que destapa una celda libre, se le muestra cuántas minas tiene alrededor.

En este problema os daremos la configuración de un tablero de *Buscaminas* totalmente descubierto, y os pedimos que nos digáis cuántas celdas vacías tienen al menos 6 minas a su alrededor.



## Entrada

La entrada estará compuesta de múltiples casos de prueba. Cada uno comienza con una línea conteniendo dos números enteros positivos, menores que 1.000, que indican, respectivamente, el ancho y el alto del tablero. A continuación vendrá una línea por cada fila del tablero. Cada celda se representa con un \* indicando que en esa celda hay una mina, o con un - indicando que está libre.

La entrada termina con un tablero de ancho o alto 0.

## Salida

Para cada caso de prueba se debe indicar, en una única línea, el número de celdas vacías con al menos 6 minas alrededor.

## Entrada de ejemplo

```
5 4
*--**
---*-
---**
***--
4 5
**-*
****
*--*
**-*
-***
0 0
```

## Salida de ejemplo

```
0
3
```