



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

# Programa-Me 2017

## Regional de Madrid

### Problemas

Patrocinado por



Ejercicios realizados por



Universidad Complutense  
de Madrid

Realizado en el CFTIC de Getafe (Madrid)



21 de marzo de 2017

21 de marzo de 2017  
<http://www.programa-me.com>

## Listado de problemas

<b>A Trifelios</b>	<b>3</b>
<b>B Las dimensiones del puzzle</b>	<b>5</b>
<b>C En campos de fútbol</b>	<b>7</b>
<b>D El teléfono de Meucci</b>	<b>9</b>
<b>E Imprimiendo páginas sueltas</b>	<b>11</b>
<b>F Yonge Street</b>	<b>13</b>
<b>G Rebotando en el parchís</b>	<b>15</b>
<b>H La maldición de la lotería</b>	<b>17</b>
<b>I Seleccionando castellers</b>	<b>19</b>
<b>J Dígitos saltarines</b>	<b>21</b>

Autores de los problemas:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)

Revisores:

- Ferran Borrell Micola (I.S. Baix Camp - Reus)
- Cristina Gómez Alonso (I.S. Baix Camp - Reus)

Agradecemos a Roger Ciurana (Cap de Colla de Xiquets de Reus) su ayuda en la elaboración del problema **I**. La imagen de dicho problema es © Fototeca.cat



# ● A Trifelios

Aunque poca gente sabe lo que es un *trifelio*, casi todo el mundo conoce al menos uno: “*monja*”. Como todos los niños saben, si se repite rápidamente esa palabra muchas veces, aparece otra distinta, “*jamón*”.

Hay muchas otras parejas de palabras que tienen esta misma propiedad, como “*copa*” y “*Paco*”, “*mora*” y “*amor*” o “*diosa*” y “*adiós*”.

Otras requieren cambios ortográficos (más allá de las tildes) como “*carro*” y “*roca*”, “*llave*” y “*bella*” o “*labio*” y “*viola*”.

Por último, también existen algunas parejas que son trifelios al escribirlos, pero no al decirlos, ya sea por cambio de fonética o de entonación (o ambas), como “*cogeré*” y “*recoge*” o “ *encuadernaré*” y “*reencuaderna*”. Los llamamos *trifelios ortográficos*.



## Entrada

La entrada comienza con el número de casos de prueba que se deberán procesar. Cada caso ocupará una línea independiente y contendrá dos palabras de no más de 20 letras. Aunque suponga fallos ortográficos, por simplicidad ninguna vocal llevará tilde.

## Salida

Para cada caso de prueba se escribirá “SI” si la pareja de palabras es un trifelio, y “NO” en otro caso.

Dado que fonéticamente la ‘b’ y la ‘v’ son similares, deben considerarse iguales. De nuevo por simplicidad, el resto de idiosincrasias del español deberán ignorarse y considerar por tanto únicamente los trifelios *ortográficos*. Ten en cuenta que una palabra *no* forma un trifelio consigo misma ni, en este problema, con sus posibles variaciones de ‘b’ y ‘v’.

## Entrada de ejemplo

```
5
monja jamon
Paco copa
carro roca
lavese Besela
vota bota
```

## Salida de ejemplo

```
SI
SI
NO
SI
NO
```



## ● B

# Las dimensiones del puzzle

Los puzzles son un pasatiempo que consiste en reconstruir una figura uniendo correctamente las diferentes partes que la componen, normalmente pequeñas piezas planas que encajan unas con otras. La cantidad de piezas de las que se componga el puzzle es una indicación de su complejidad. También lo es la imagen que contiene; las zonas con un color uniforme son mucho más difíciles de montar, al no tener ayuda sobre qué pieza va junto a qué otra.

Cualquiera que haya montado un puzzle alguna vez sabe que lo más sencillo es comenzar buscando las piezas que forman los laterales, para construir el borde externo. Se empieza buscando las piezas que son planas en alguno de sus lados y una vez recopiladas se encajan hasta formar el marco. En este segundo paso, resulta muy útil conocer el número de piezas a lo ancho y a lo alto que tiene el puzzle.



Por fin vas a empezar a montar el puzzle que te regalaron hace varios años y quieres saber sus dimensiones. Conoces el número total de piezas, dado que lo indica la caja. Suponiendo que el ancho y el alto son lo más parecido posible, ¿eres capaz de averiguar las dimensiones del puzzle sabiendo que no sobra ni falta ninguna pieza?

### Entrada

El programa leerá de la entrada estándar múltiples casos de prueba, uno por línea. Cada caso de prueba consta de un número positivo menor o igual que  $10^9$  indicando el número total de piezas del puzzle.

La entrada termina con un puzzle sin piezas, que no deberá procesarse.

### Salida

El programa escribirá en la salida estándar, para cada caso de prueba, las dimensiones del puzzle correspondiente, indicando primero el ancho y luego el alto separados por un espacio.

Recuerda que si hay varias posibles respuestas, se dará aquella en la que la distancia entre el ancho y el alto sea mínima. Además, se asumirá que el ancho no será nunca menor que el alto.

### Entrada de ejemplo

```
1000
2000
500
0
```

### Salida de ejemplo

```
40 25
50 40
25 20
```







# En campos de fútbol

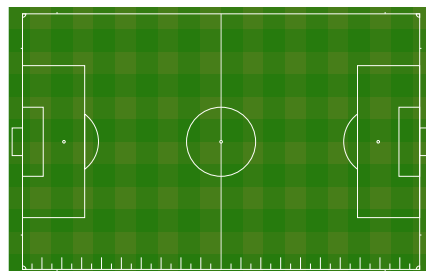
Según el Sistema Internacional de Unidades, para medir superficies debe utilizarse el *metro cuadrado* (abreviado  $m^2$ ), que es el área en el interior de un cuadrado cuyos lados miden exactamente un metro.

Un metro es la longitud del trayecto recorrido por la luz en el vacío en un intervalo de tiempo de  $1/299.792.458$  segundos.

Un segundo es la duración de  $9.192.631.770$  periodos de la radiación correspondiente a la transición entre los dos niveles hiperfinos del estado fundamental del átomo de cesio 133.

El Sistema Internacional de Unidades se jacta de que sus unidades se basan en fenómenos físicos fundamentales. Pero cuando los periodistas tienen que hablar sobre la superficie quemada en un incendio forestal, el espacio arrasado por unas inundaciones, o la cantidad de cultivo echado a perder por un inoportuno granizo, no suelen llevar en el bolsillo cesio 133 para empezar a medir.

La solución, aceptada por el Comité Internacional de Periodistas, es medir la superficie en campos de fútbol, que es algo mucho más fácil de hacer a ojo. Especialmente porque el tamaño de un campo de fútbol es algo impreciso; el largo admitido puede estar entre 90 y 120 metros, y el ancho entre 45 y 90. Con tanto margen, los periodistas pueden tener las espaldas cubiertas por muy desentrenado que tengan el ojo; se dé la cifra que se dé, es muy posible que haya un campo de fútbol con el que encaje la estimación.



## Entrada

El programa leerá, de la entrada estándar, un primer número con la cantidad de casos de prueba que deberá procesar. Cada caso de prueba es una superficie medida en el antiguo sistema de metros cuadrados, seguido de la estimación, en campos de fútbol, de un periodista. Ambos números serán mayores que 0 y no excederán el 100.000.

## Salida

Por cada caso de prueba, el programa escribirá "SI" si existe un campo de fútbol, de dimensiones legales, que hace que la estimación sea correcta, y "NO" en caso contrario.

## Entrada de ejemplo

```
5
10 1
10000 1
10000 2
10000 3
11000 1
```

## Salida de ejemplo

```
NO
SI
SI
NO
NO
```





# El teléfono de Meucci

En 1876, Alexander Graham Bell patentó el teléfono, lo que le sirvió para ser considerado su inventor durante más de un siglo.

La realidad, sin embargo, fue bien distinta, porque el invento había sido creado por Antonio Meucci, un inmigrante italiano afincado en Estados Unidos que en 1860 había mostrado al mundo el llamado *teletrófono*. Las penurias económicas que estaba pasando Meucci no le permitieron pagar la patente de su invento y Graham Bell se aprovechó de ello. Afortunadamente, el año 2002 puso a cada uno en su lugar cuando el congreso de los Estados Unidos aprobó una resolución en la que se reconocía que, efectivamente, el inventor había sido Meucci y la oficina de patentes había fallado a la hora de hacer bien su trabajo.



Hoy nos planteamos qué habría pasado si hubiera sido él quién hubiera explotado el invento desde el principio. Es posible que, al ser italiano, los números para marcar en lugar de venir expresados con los dígitos arábigos podrían haber venido en números romanos. Así, el número 555-55-55 habría sido VVV-VV-VV.

La primera implicación de ese cambio es que no habríamos tenido el cero entre los dígitos y quizá habría sido sustituido por la X que representa el 10. El número 555-50-05 sería entonces VVV-VX-XV.

La segunda implicación es mucho más dramática, porque la forma que tenemos de escribir un número de teléfono no serviría. La razón está en la forma de escribir cada dígito en notación romana:

Dígito	0	1	2	3	4	5	6	7	8	9
Símbolo	X	I	II	III	IV	V	VI	VII	VIII	IX

Con esta codificación, un número de teléfono como IVII podría representar, entre otros, I-V-I-I (1511), I-V-II (152), IV-II (42) o I-VI-I (161).

Ante semejante locura, dado un número de teléfono expresado en números romanos, ¿a cuántos números de teléfono distintos puede representar?

## Entrada

La entrada estará formada por distintos números de teléfono, cada uno en una línea.

Cada número está expresado en números romanos según lo explicado más arriba. Habrá como mucho 20 caracteres, que serán únicamente I, V y X.

## Salida

Por cada caso de prueba se escribirá, en una línea independiente, el número de teléfonos distintos a los que puede representar, sabiendo que los únicos “dígitos romanos” válidos son los indicados en la tabla. Se garantiza que el resultado será siempre menor que  $2^{31}$ .

## Entrada de ejemplo

XX
III
IVII

## Salida de ejemplo

1
4
6





# Imprimiendo páginas sueltas

En virtualmente todos los cuadros de diálogo que muestran las aplicaciones a la hora de imprimir, el usuario puede elegir qué páginas quiere conseguir en papel, para evitar que se imprima todo el documento.

Si se quieren varias páginas, se pueden indicar sus números separados por comas. Si esas páginas son consecutivas, por ejemplo todas las páginas entre la 10 y la 20, en lugar de escribir una a una, se puede especificar el rango con los números de los extremos separados con un guión, "10-20".

Sabiendo las páginas que quieres imprimir, ¿cuál es la forma más corta de indicarlo en el cuadro de diálogo?



## Entrada

El programa tendrá que procesar varios casos de prueba. Cada uno está compuesto por los números de las páginas que se quieren imprimir, de menor a mayor separadas por espacio. Ningún documento tiene más de  $10^9$  páginas y nunca querremos imprimir la misma página más de una vez. La secuencia acaba con un 0.

La entrada termina con un caso sin páginas (una línea con un único 0), que no deberá procesarse.

## Salida

Para cada caso de prueba se debe escribir la especificación más corta posible que hay que indicar en el cuadro de diálogo para imprimir las páginas deseadas. Si hay varias opciones, se preferirá siempre especificar las páginas con un rango.

Entre los números y los separadores (coma o guión) no se deben introducir espacios.

## Entrada de ejemplo

```
1 2 3 4 0
1 2 5 6 0
100 122 123 125 0
0
```

## Salida de ejemplo

```
1-4
1-2,5-6
100,122-123,125
```



# ● F Yonge Street



Aunque no es su capital, Toronto es la ciudad más grande de Canadá y su centro financiero. Además, está dentro del Libro Guinness de los Récords por tener la calle más larga del mundo, llamada Yonge Street, con una longitud total de 86 kilómetros.

Todavía está por incorporar a ese mismo libro el nombre de la primera persona que consiga recorrerla entera sin que le pille ningún semáforo en rojo. Y, con semejante tamaño, es poco probable que alguien lo consiga antes de que se invente el teletransporte.

Aun así, es espectacular ver, desde un extremo, como todos los semáforos cambian de color hasta donde alcanza la vista; y como todos los coches pisan el freno, llenando 86 kilómetros de luces rojas.

Dadas las posiciones de todos los vehículos y de los semáforos en el momento en el que éstos se ponen en rojo, ¿eres capaz de saber dónde terminará parado cada uno haciendo cola hasta que los semáforos vuelvan a estar verdes?



## Entrada

Cada caso de prueba comienza con una línea con hasta 200.000 caracteres representando la situación de Yonge Street en el momento en el que todos sus semáforos se cierran.

Un carácter 'l' indica la posición de un semáforo, un carácter '.' indica una posición libre, y las letras del alfabeto inglés representan vehículos de diferente tipo (automóvil, furgoneta, camión, tractor... amigo que vas en burro).

Tras esa línea aparece otra, con consultas sobre el estado final de la calle. El primer número,  $n$ , indica cuántas consultas vienen a continuación, en esa misma línea, separadas por espacio. Cada una consiste en el número de posición por la que se pregunta. La primera posición de la calle, más a la izquierda en la entrada, se corresponde con el número 1.

## Salida

Para cada consulta de los casos de prueba se escribirá el contenido de esa posición una vez que todos los vehículos han parado en sus posiciones a la espera de que se abran los semáforos. Las respuestas de cada caso de prueba se darán todas en la misma línea, separadas por espacio. No debe aparecer un espacio tras la última respuesta.

Los vehículos se mueven de izquierda a derecha hasta llegar a un semáforo, momento en el que se detendrán, delante de él, formando una fila. Canadá es un país tranquilo, y nadie se adelanta mientras conduce.

## Entrada de ejemplo

```
A.|.B
5 1 2 3 4 5
AB...|
6 6 5 4 3 2 1
.A|.
1 2
```

## Salida de ejemplo

```
. A | . .
| B A . . .
A
```





## ● G

# Rebotando en el parchís

En el parchís y muchos otros juegos de mesa en los que los jugadores compiten por ser los primeros en alcanzar una meta, a menudo es necesario llegar al destino de manera exacta. Tras la tirada de dados, si el valor obtenido es más del necesario para llegar al destino, la ficha debe *rebotar*, volviendo hacia atrás una vez alcanzada la meta.



A veces, conseguir caer exactamente en la posición final es lo más tedioso, y los jugadores ven cómo sus fichas rebotan continuamente en ella sin ser capaces de que el azar les deje, por fin, terminar la partida.

### Entrada

Cada caso de prueba está compuesto por tres números, indicando la cantidad de casillas  $c$  del tablero de la variante del parchís a la que se está jugando, la posición  $p$  en la que está la ficha del jugador, y el número  $t$  obtenido en la última tirada de dados. La primera posición del tablero es la 1, la última es la meta, y se garantiza que  $1 \leq p, t < c \leq 10.000$ .

Un caso con todos los números a 0 marca el final.

### Salida

Para cada caso de prueba, el programa escribirá la posición final de la ficha en el tablero tras la tirada.

### Entrada de ejemplo

```
100 99 1
100 99 2
900 800 300
0 0 0
```

### Salida de ejemplo

```
100
99
700
```





# La maldición de la lotería

Cuando nacieron sus nietos, Sue Erth Uda cambió los números a los que jugaba en la bonoloto de su país. Y fue una buena decisión. A los 90 años, le tocó una cantidad ingente de dinero que la hizo entrar, directamente, en la lista Forbes de las personas más ricas del mundo.



No fue consciente de la magnitud de su premio, por lo que cuando lo cobró pidió que se lo ingresaran en su cuenta bancaria, la de toda la vida, la misma que había compartido con su difunto marido. Esa cuenta tenía algunas restricciones, poco importantes en condiciones normales, pero que resultaron desastrosas para sus herederos. En concreto, la cuenta tenía un límite en la cantidad de efectivo del que se podía disponer. Como mucho, se podía extraer al día 10 doblones, a la semana 30, y al mes 90. Eso significaba que, con esos números, si la misma semana sacaba el máximo durante tres días, el resto de días no podría sacar nada de dinero. Del mismo modo, si tres semanas del mes sacaba el máximo, la cuarta no podría sacar nada.

El premio ascendió a la astronómica cifra de 10.000.000 doblones, por lo que, cuando falleció, lo hizo siendo rica, pero sólo sobre el papel.

## Entrada

La entrada consiste en un primer número con la cantidad de casos de prueba que deberán procesarse. Por cada uno, se proporcionarán cuatro números,  $p$ ,  $d$ ,  $s$  y  $m$  con la cantidad de dinero del premio, y el límite por día, semana y mes. Se garantiza que las restricciones serán coherentes, es decir  $d \leq s \leq 7 \times d$  y  $s \leq m \leq 4 \times s$ . Todos los números serán mayores que 0 y menores que  $10^9$

## Salida

Para cada caso de prueba se indicará el tiempo mínimo necesario para sacar todo el dinero del premio del banco. Se deberá indicar el número de meses *completos* necesarios, seguido del número de semanas *completas* del último mes, seguido, finalmente, del número de días de la última semana en la que se ha tenido que sacar dinero.

Por simplicidad, se asumirá que todos los meses tienen exactamente 4 semanas y, obviamente, todas las semanas tienen 7 días.

## Entrada de ejemplo

```
5
100 100 200 300
100 50 100 300
100 50 90 300
28 1 7 28
10000000 10 30 90
```

## Salida de ejemplo

```
0 0 1
0 0 2
0 1 1
0 3 7
111111 0 1
```





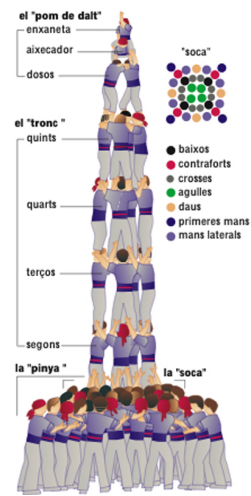
# Seleccionando castellers

La *colla castellera* “Xiquets de Reus” ensaya duramente para mejorar sus *castells*, lo que le lleva a conseguir constantes récords en la elaboración de formaciones cada vez más complejas.

Un *castell* es una torre humana donde toda su estructura se mantiene en pie exclusivamente gracias a la fuerza de sus integrantes, llamados *castellers*.

Los *castells* se clasifican en función de dos factores. El primero es el número de personas que forman cada nivel (también llamados plantas o pisos); el segundo es su altura, es decir el número de plantas que alcanza. Se utiliza la notación  $cdn$ , donde  $c$  indica el número de *castellers* en cada nivel, y  $n$  el número de niveles. Cuando los pisos están compuestos por un único *casteller*, la construcción es conocida con el nombre de *pilar*, y en ese caso en lugar de poner un 1 en la  $c$ , se pone directamente una  $p$  (por ejemplo,  $pd8$ ).

La figura de la derecha muestra un *castell* 4d8. Observa que en la parte superior aparecen 3 pisos que no están compuestos por 4 personas. Debido a su dificultad e importancia, los tres niveles superiores son especiales y reciben nombres particulares:



- *Enxaneta*: es el *casteller* que se sitúa en el nivel más alto, y da validez a la estructura.
- *Acotxador*: también llamado *aixecador*, es el *casteller* que se sitúa horizontalmente en el penúltimo nivel, dando el apoyo necesario al *enxaneta*.
- *Dosos*: son los *dos* integrantes del antepenúltimo nivel, justo debajo del *acotxador*. En los pilares, este piso también consta de un único integrante.

Al margen de la estructura central, existen un conjunto de *castellers* que se posicionan en la base formando la *piña*, que respalda toda la estructura conforme va creciendo. Opcional y adicionalmente a ella, también se puede situar un pequeño grupo de *castellers* por encima de la piña, para afianzar el segundo nivel cuando la altura del *castell* y su complejidad así lo requieren. Esta estructura se llama *folre*.

Para que un *castell* alcance altura y estabilidad, son muchos los factores a tener en cuenta durante su construcción: la fuerza de sus integrantes, su altura, su peso, experiencia, valentía, soltura... Por ejemplo, para evitar desequilibrios, es importante que los *castellers* de un mismo piso tengan una diferencia de altura de, como máximo, 15 centímetros. Las comisiones técnicas de las *collas castelleras* deben decidir quién integrará cada piso en función de las alturas de los *castellers* de los que disponen debido a esta restricción.

## Entrada

La primera línea de la entrada indicará el número de casos de prueba que deberá procesar el programa.

Para cada caso de prueba se indicará en una primera línea el número de *castellers* por nivel de la construcción que se quiere intentar en la próxima salida (entre 1 y 9), junto a la cantidad de integrantes de la *colla* entre los que elegirlos (entre 1 y 100).

La siguiente línea del caso de prueba contendrá las alturas de todos ellos, expresadas en centímetros, sin ningún orden particular. Las alturas fluctúan entre 100 y 220 centímetros.

## Salida

Para cada caso de prueba, el programa escribirá el número máximo de niveles del tamaño indicado que se pueden formar con los *castellers* de la *colla*, manteniendo la restricción de que la máxima diferencia de alturas entre los integrantes de un piso no puede ser mayor de 15 centímetros.

Por su importancia, la comisión técnica ha elegido ya quienes serán el *enxaneta*, el *actoxador* y los *dosos*, por lo que sus alturas no se listan en la entrada, y no deben tenerse en cuenta para la salida.

Tampoco hay que preocuparse de la *piña* ni el *folre* que, aunque tienen una importancia clave en la construcción, son más tolerantes con las diferencias de alturas.

### Entrada de ejemplo

```
4
3 9
152 159 167 173 180 188 194 201 209
3 9
152 160 168 176 184 192 200 208 216
3 10
170 105 130 175 180 192 195 172 195 165
2 10
170 105 130 175 180 192 195 172 195 165
```

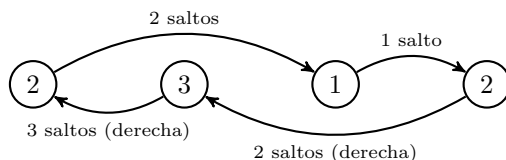
### Salida de ejemplo

```
3
0
2
4
```



# Dígitos saltarines

Un número tiene sus dígitos saltarines si, empezando por el primero, se puede ir saltando de uno al siguiente tantos puestos como indiquen, y al final se llega de nuevo al principio habiendo pasado por todos ellos. Por ejemplo, el número 2.312 tiene a sus dígitos saltarines. Empezando por el primero, el primer 2, avanzamos dos dígitos y llegamos al 1. Desde él, avanzamos un dígito más, y llegamos al segundo 2. Desde él, avanzamos otros dos dígitos (dando la vuelta y empezando de nuevo por la izquierda), y llegamos al 3. Y desde éste, avanzamos tres veces, y llegamos otra vez al principio.



Los números con dígitos saltarines son una rareza de la naturaleza y son difíciles de encontrar.

## Entrada

El programa deberá leer un primer número  $n$  indicando cuántos casos de prueba deberá procesar. A continuación vendrán  $n$  líneas, cada una con un número  $0 < i < 10^9$ .

Ningún número tendrá ceros superfluos a la izquierda.

## Salida

Para cada caso de prueba, el programa escribirá “SALTARINES” si el número tiene a sus dígitos saltarines, y “NORMALES” en otro caso.

## Entrada de ejemplo

```
4
2312
3
12
30
```

## Salida de ejemplo

```
SALTARINES
SALTARINES
NORMALES
NORMALES
```