

Programa-Me

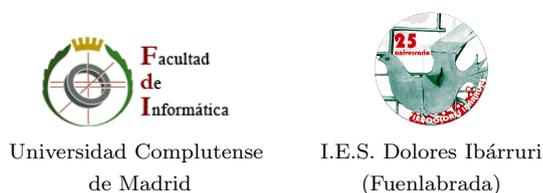
“First, solve the problem. Then, write the code” John Johnson.

Programa-Me 2015 Regional de Madrid Problemas

Patrocinado por



Ejercicios realizados por



Realizado en el I.E.S. Clara del Rey (Madrid) - 18 de marzo de 2015



18 de marzo de 2015
<http://www.programa-me.com>

Listado de problemas

A Números polidivisibles	3
B Repartiendo el botín	5
C Pi. Pi. Pi. Pi. Pi. Piiii	7
D Completa la suma	9
E Me quiere, no me quiere	11
F Erasμός	13
G Asalto al castillo	15
H Reinas atacadas	17
I Los Dalton	19
J Búsqueda de polidivisibles	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. Dolores Ibárruri - Fuenlabrada)

Revisores:

- Ferran Borrell Micola (I.S. Baix Camp - Reus)
- Cristina Gómez Alonso (I.S. Baix Camp - Reus)
- Marc Nicolau Reixach (I.S. Bosc de la Coma - Olot)



Números polidivisibles

El número 381.654.729 tiene una propiedad muy curiosa que no cumple ningún otro número. Si lo miras con cuidado es probable que te des cuenta de que tiene los nueve dígitos entre el 1 y el 9 y que no repite ninguno de ellos. Sin embargo, eso no es lo único especial que tiene (hay muchos otros números así).

Lo que realmente lo hace singular es que, además de lo anterior, es divisible por 9; si se le quita el último dígito, queda un número divisible por 8; si se le vuelve a quitar el último dígito, queda un número divisible por 7; y así continuamente hasta llegar a un número de un único dígito que, naturalmente, es divisible por 1:

$$\begin{aligned}381.654.729 &= 9 \times 42.406.081 \\38.165.472 &= 8 \times 4.770.684 \\3.816.547 &= 7 \times 545.221 \\381.654 &= 6 \times 63.609 \\38.165 &= 5 \times 7.633 \\3.816 &= 4 \times 954 \\381 &= 3 \times 127 \\38 &= 2 \times 19 \\3 &= 1 \times 3\end{aligned}$$

Esta última peculiaridad es lo que en matemáticas se conoce como un número *polidivisible*, que puede definirse de la siguiente forma: un número es *polidivisible* si es divisible por su *longitud* y, además, si se le quita el último dígito queda un número que es a su vez polidivisible.

Existen otros números polidivisibles como el 102 o el 9.876. Pero su cantidad es limitada: hay un total de 20.456 números polidivisibles distintos, el mayor de los cuales tiene 25 dígitos.

Entrada

La entrada estará compuesta por distintos números positivos, cada uno de ellos en una línea. Los números serán siempre mayores que cero y nunca mayores que 10^{18} .

Salida

Por cada número, se escribirá una línea en donde aparecerá "POLIDIVISIBLE" si el número es polidivisible según la definición anterior o "NO POLIDIVISIBLE" si no lo es.

Entrada de ejemplo

```
381654729
102
9876
67
```

Salida de ejemplo

```
POLIDIVISIBLE
POLIDIVISIBLE
POLIDIVISIBLE
NO POLIDIVISIBLE
```




Repartiendo el botín

Al-Colleja y sus secuaces tienen que repartir el botín de su último golpe. No es una tarea fácil, porque todos quieren llevarse lo máximo posible, y todos están armados...



Para no entrar en discusiones que terminen en tragedia, Al-Colleja ha ideado un sencillo método en el que, en lugar de preocuparse de ser justos repartiendo en base a quién ha trabajado más en la consecución del golpe, se lo deja prácticamente todo al azar. Prefiere recibir menos beneficios pero mantener la banda intacta.

El procedimiento es sencillo. Coge todos los billetes conseguidos y los pone en un montón tras barajarlos. Después se coloca toda la banda en círculo y va dando un billete a cada uno, hasta que quedan todos repartidos. Eso sí, el primero que recibe billete es él, de esa forma se asegura de que si los billetes se terminan a mitad de una vuelta, él siempre habrá recibido uno adicional.

El componente de azar aparece porque los billetes están descolocados, así que puede tocar en el reparto desde el mísero billete de 10 hasta el deseado de 500...

Entrada

La entrada consta de una sucesión de casos de prueba o botines a repartir. Para cada botín aparece una primera línea que indica el número de billetes que hay que repartir (como mucho 1.000) seguido del número de participantes en el golpe que deben recibir recompensa (no más de 10). Siempre habrá al menos un billete y al menos un villano. La segunda línea contiene tantos números como billetes, indicando su valor (todos los billetes tienen valor). El primer número expresa la cuantía del primer billete que se reparte.

La entrada termina con una línea con dos ceros que no debe procesarse.

Salida

Para cada caso de prueba se mostrará una línea por cada participante. Cada línea tendrá un primer número indicando la cantidad de dinero recibido seguido del carácter ':' y la lista de billetes que recibe esa persona separados por espacios.

Tras cada caso de prueba aparecerá una línea con tres guiones (---).

Entrada de ejemplo

```
5 2
10 20 50 200 500
3 2
10 50 10
1 2
50
0 0
```

Salida de ejemplo

```
560: 10 50 500
220: 20 200
---
20: 10 10
50: 50
---
50: 50
0:
---
```




Pi. Pi. Pi. Pi. Pi. Piiii

El primer uso de *señales horarias* por parte de la humanidad se pierde en la noche de los tiempos. El canto de un gallo o el tañido de una campana sirvieron como mecanismos primitivos para avisar de los momentos en los que realizar ciertos actos en común.

La medición del tiempo adquirió una importancia capital cuando se comenzó a utilizar para averiguar la longitud de un punto cualquiera de la tierra al comparar la hora local con la de un punto de referencia conocido. Si bien esta solución se hizo viable por primera vez en 1760, no fue hasta la segunda mitad del siglo XIX cuando comenzó a utilizarse ampliamente. En esa época se establece definitivamente el meridiano que pasa por Greenwich como el meridiano cero para todo el mundo. Además, se construye el famoso *Big Ben*, reloj que servía para dar la hora oficial de Londres sincronizando así, entre otras cosas, los ferrocarriles y el tráfico marítimo.



Pero, obviamente, las señales horarias del Big Ben no son audibles a grandes distancias. Para solucionarlo, en la noche de Año Nuevo de 1924, la BBC decidió retransmitir sus campanadas a todo el país. El éxito fue tal, que el 5 de febrero de ese mismo año comenzaron a emitir, a todas las horas en punto, las que hoy se conocen como *Greenwich Time Signal* o GTS (“señales de la hora de Greenwich”), que terminaron extendiéndose por las radios de todo el mundo tras la Segunda Guerra Mundial. Son los 6 famosos pitidos, cinco cortos en los últimos segundos de cada hora, y uno más largo en el primer segundo de la hora siguiente, que podemos oír en prácticamente todas las emisoras de radio.

Cada GTS supone un total de 6 segundos durante los cuales las radios de todo el mundo no emiten nada más para evitar lo que los ingleses llaman “*crashing the pips*”. Con el paso de los años, eso ha ido acumulando más y más tiempo en silencio. ¿Sabes cuánto?

Entrada

El programa recibirá, por la entrada estándar, múltiples casos de prueba. Cada uno consiste en dos números, el primero indicando el número de días durante los cuales se ha emitido el GTS, y el segundo el número de emisoras que lo han hecho.

El último caso de prueba, que no deberá procesarse, tendrá 0 en ambos valores.

Salida

Para cada caso de prueba el programa escribirá, en una línea independiente, el tiempo dedicado por las emisoras a radiar el GTS. El formato será `D:HH:MM:SS`, donde `D` indica número de días, `HH` número de horas, `MM` número de minutos y `SS` número de segundos totales.

Como es lógico, `HH` no deberá ser mayor que 23 y `MM` y `SS` no podrán ser mayores de 59. Además deberán escribirse siempre con dos dígitos. Para el número de días (`D`), que no será nunca mayor de 1.000, no se deben escribir dígitos adicionales.

Entrada de ejemplo

```
1 1
3 9
300 2
0 0
```

Salida de ejemplo

```
0:00:02:24
0:01:04:48
1:00:00:00
```


● D

Completa la suma

Son habituales los acertijos numéricos en los que los dígitos son sustituidos por letras y hay que descubrir esa sustitución. El siguiente es uno de los más famosos, en el que se pide que se obtenga la relación entre letras y dígitos sabiendo que no hay dos letras asignadas al mismo dígito:

$$\begin{array}{r} \text{SEND} \\ + \text{MORE} \\ \hline \text{MONEY} \end{array}$$

Otros, mucho más sencillos, se limitan a poner una operación (por ejemplo una suma) y quitan algunos dígitos que son los que hay que completar. Por ejemplo:

$$\begin{array}{r} 87- \\ + 2-1 \\ \hline 1-63 \end{array}$$

Es fácil ver que la suma buscada es $872 + 291 = 1163$. ¿Eres capaz de hacer un programa que resuelva este tipo de acertijos?

Entrada

El programa deberá leer de la entrada estándar una serie de casos de prueba, cada uno en una línea.

Cada caso de prueba consiste en tres números separados por un espacio. Los dos primeros corresponden a los sumandos, y el tercero al resultado. Cada uno de los tres números tendrá siempre un único valor desconocido, representado por el carácter “-”. Ningún número tendrá más de 9 dígitos ni ceros supérfluos a la izquierda.

Salida

Para cada caso de prueba se escribirá la suma buscada, con el mismo formato que en la entrada, pero sin valores desconocidos. Para que la suma sea considerada correcta, todos los números deben tener la misma cantidad de dígitos que tenían en el planteamiento del acertijo, y no deben tener ceros supérfluos a la izquierda.

Si hay varias soluciones posibles se escribirá “VARIAS”. Si no hay ninguna, se escribirá “IMPOSIBLE”.

Entrada de ejemplo

```
87- 2-1 1-63
87- 29- 1-63
87- 2-1 -63
- - 2-
1- -1 -11
```

Salida de ejemplo

```
872 291 1163
VARIAS
IMPOSIBLE
IMPOSIBLE
IMPOSIBLE
```




Me quiere, no me quiere

El hermano pequeño de Diana está descubriendo lo que es el amor. Anda siempre despistado de un lado para otro, sin prestar atención a los demás y con la cabeza llena de pájaros.

Alguien le enseñó el juego de “me quiere, no me quiere” con margaritas, pero él prefiere, sin lugar a dudas, los tréboles. Y es que, quitando una por una sus tres hojas, su enamorada ¡siempre le quiere! El resultado es que cada vez que en el jardín encuentra una zona en la que han crecido varios tréboles, se dedica a deshojarlos uno por uno repitiendo, para sí, la rítmica cancioncilla.

Diana ha dejado ya de creer en esas cosas de niños. Pero todavía sigue pensando que encontrar un trébol de cuatro hojas le traería suerte. Cada vez que descubre en el jardín el resultado de las locuras de amor de su hermano entra en cólera porque sólo quedan las hojas de los desafortunados tréboles desperdigadas por el suelo, y no puede buscar su suerte allí.

¿O sí?



Entrada

El programa recibirá, por la entrada estándar, un primer número indicando cuántos casos de prueba vendrán a continuación. Cada uno estará compuesto por un único número, en una línea, mayor que 0 y menor que 20.000, indicando el número de hojas de trébol que Diana ha encontrado en el jardín.

Salida

Para cada caso de prueba el programa escribirá, en una línea independiente, el *menor* número posible de tréboles de cuatro hojas que ha encontrado, y deshojado, su hermano. Si es imposible que se encuentre con ese número de hojas, se escribirá **IMPOSIBLE**. Se debe asumir que los tréboles tendrán siempre 3 ó 4 hojas.

Entrada de ejemplo

```
3
1
3
7
```

Salida de ejemplo

```
IMPOSIBLE
0
1
```


● F Erasmús

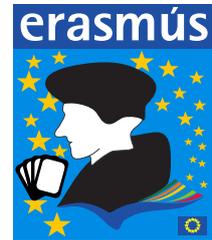


Es conocida la afición existente entre los universitarios españoles al *mus*, un juego de cartas para cuatro personas con más de 200 años de historia que también es muy jugado en algunos países de Hispanoamérica e incluso en algunas regiones del sur de Francia.

Cuando esos españoles consiguen una beca *erasmus* para estudiar durante unos meses en otro país de Europa, es fácil entender, pues, que no falte una baraja de cartas entre su equipaje, por lo que pueda ocurrir. En su afán de *evangelización* muchos de estos estudiantes de erasmus intentan que el juego se extienda, y se empeñan en explicar las reglas a sus compañeros de otros países.

Como a buen seguro sabrás, al *mus* se juega por parejas, de forma que una pareja de jugadores se enfrenta a otra. Desgraciadamente, a la hora de hacer esas parejas la mayoría de las veces los españoles “se buscan” entre sí, de forma que las partidas terminan siendo competiciones entre países (juegan dos españoles contra dos italianos, por ejemplo).

En esta ocasión, sin embargo, un grupo de estudiantes de erasmus se ha propuesto formar parejas heterogéneas, por lo que no van a aceptar parejas cuyos dos componentes sean de la misma nacionalidad. A la vista de cuántos estudiantes hay de cada país, ¿cuántas parejas distintas pueden formarse?



Entrada

La entrada estará compuesta por varios casos de prueba terminados con una línea con un 0.

Cada caso de prueba tendrá dos líneas. En la primera aparecerá un único entero que indica el número de países totales representados (un número entre 1 y 100.000). La segunda línea contendrá un número por cada país, representando la cantidad de estudiantes de ese país que quieren jugar al *mus*. El número de estudiantes de cada país no excederá nunca 10^9 .

Salida

Por cada caso de prueba, el programa escribirá el número de parejas distintas que pueden formarse sin que una pareja tenga a sus dos integrantes de la misma nacionalidad. Se garantiza que la respuesta no será nunca superior a 10^{18} .

Entrada de ejemplo

```
2
1 1
2
2 2
3
2 2 1
2
1000000 1000000
0
```

Salida de ejemplo

```
1
4
8
1000000000000
```




Asalto al castillo

Jaime Lannister, más conocido por el *Matarreyes*, acaba de llegar a Aguasdulces. Venía con la intención de entrar en el castillo de forma pacífica pero Brynden Tully, el *Pez Negro*, se ha negado a aceptar sus condiciones, así que tendrá que tomarlo a punta de espada. No le gusta romper sus promesas pero al fin y al cabo todos saben lo que vale la palabra del *Matarreyes*; una traición más o menos no extrañará a nadie. A pesar de todo, Jaime nota un regusto amargo cuando piensa en el tiempo que pasó encerrado en aquel castillo y en como Catelyn Stark le hizo prometer que nunca levantaría una espada contra sus habitantes antes de liberarle.



En fin, los hechos son los hechos y, por poco que le guste, ha llegado el momento de acabar con la rebeldía del *Pez Negro* como le ha ordenado su querida hermana.

En estos momentos, Jaime cuenta con 25.000 hombres de diversas casas afines a los Lannister y está pensando cual sería la mejor distribución en batallones de cara al ataque de los muros. Quiere repartirlos de manera que los batallones tengan el mismo número de hombres y se ha dado cuenta de que hay múltiples maneras de realizar el reparto; por ejemplo con esos 25.000 hombres podría hacer 100 batallones de 250 hombres, 250 batallones de 100 hombres y otras configuraciones hasta un total de 24 formas distintas. Ahora le gustaría poder saber para cada una de sus batallas pasadas y futuras, cómo se podría repartir a sus hombres siguiendo el mismo procedimiento. ¿Podrías ayudarle?

Entrada

La entrada consta de una serie de casos de prueba. Cada uno es una línea con un número positivo entre 1 y 500.000 que indica el número de hombres de los que dispone Jaime. La entrada termina con un caso de prueba sin soldados, que no deberá procesarse.

Salida

Para cada caso de prueba se mostrará una línea en la que aparecerá el número de configuraciones posibles de batallones del mismo número de hombres que puede formar Jaime con los soldados de los que dispone.

Entrada de ejemplo

```
25000
500
1000
0
```

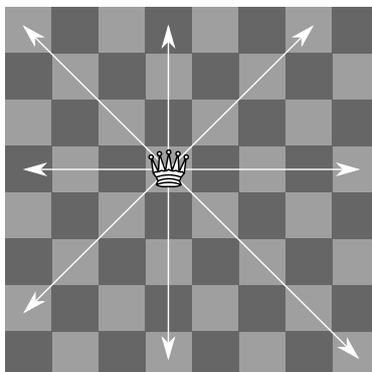
Salida de ejemplo

```
24
12
16
```



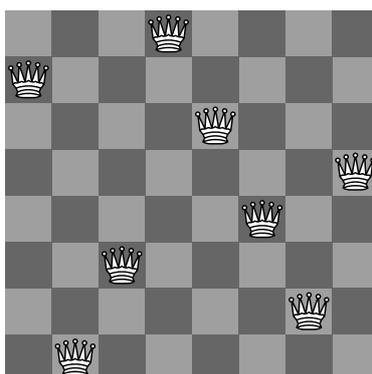

Reinas atacadas

En el ajedrez, la reina es la pieza más poderosa, al poderse mover cualquier número de escaques en vertical, horizontal, o diagonal.



Movimientos de la reina

En 1848, el alemán Max Bezzel planteó el *puzzle de las 8 reinas*, en el que retó a colocar 8 reinas sobre un tablero sin que se atacaran entre sí. Dos años después, se dieron algunas de las 92 soluciones.



Una de las soluciones posibles

Desde entonces, matemáticos y aficionados de todo el mundo han estudiado el problema, generalizándolo a tamaños de tableros de ajedrez de $N \times N$. En 1972, Dijkstra, en plena *crisis del software*, usó el problema para demostrar el poder de la programación estructurada, y desde entonces es un ejemplo clásico de algoritmo *de vuelta atrás*.

Para poder colocar las reinas, el primer paso es saber cuándo un grupo de reinas sobre un tablero de ajedrez se atacan entre sí, es decir cuándo hay al menos una reina que podría *comer* a otra siguiendo las reglas del movimiento del juego.

Entrada

La entrada consta de un conjunto de casos de prueba. Cada uno comienza con una línea con dos números. El primero indica el ancho y alto del tablero de ajedrez (siempre será cuadrado de como mucho 2.000×2.000). El segundo indica el número de reinas colocadas sobre él (entre 1 y 100).

A continuación vendrá una línea con la posición de todas las reinas. Para cada una, se indicará primero la coordenada X y luego la Y, separadas por espacio. Las posiciones de cada reina también se separarán por un único espacio. Todas las posiciones serán válidas (cada coordenada estará entre 1 y el tamaño del tablero) y se garantiza que no habrá dos reinas en la misma posición.

La entrada termina con un caso de prueba con un tablero de tamaño 0×0 y sin reinas que no debe procesarse.

Salida

Para cada caso de prueba, el programa escribirá, en la salida estándar, una línea con el texto “SI” si hay reinas atacadas en la configuración dada, y “NO” en otro caso (sin las comillas).

Entrada de ejemplo

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

Salida de ejemplo

```
NO
SI
NO
```



Los Dalton

Los Dalton (o también *hermanos Dalton*) son personajes secundarios de la serie de cómics de Lucky Luke creada por Maurice de Bévère (Morris). Pocos saben que en realidad estos personajes están inspirados en unos ladrones estadounidenses de finales del siglo XIX.



Lo que sí es ampliamente conocido es su aspecto y la forma de colocarse en las viñetas. Por un lado, todos ellos tienen la misma apariencia, siendo la altura su única diferencia física. Por otro lado, en las viñetas siempre se colocaban por orden de altura, formando una graciosa estampa reconocible al instante. Aunque en los cómics finalmente hay cuatro hermanos Dalton, existe la creencia de que Morris hizo bocetos en los que aparecían muchos más hermanos, todos exactamente iguales salvo por su altura. Hoy coleccionistas de todo el mundo buscan incansablemente esos bocetos.

Actualmente estamos trabajando en un software que reconozca si una viñeta puede o no ser una foto de los Dalton. Tras una serie de pasos de extracción de siluetas, hemos conseguido las alturas de todos los personajes que aparecen en el dibujo y nos toca decidir si pueden o no ser ellos.

Entrada

La entrada estará compuesta por la descripción de varias viñetas. Cada una de ellas aparece en dos líneas. La primera tiene el número N de personas que hay en la viñeta (como mínimo dos). La segunda tiene las N alturas de cada uno, empezando por el personaje de la izquierda y terminando por el de más a la derecha. El sistema de extracción de siluetas nos da las alturas en números enteros entre 1 y 10^{18} . Ten en cuenta que no estamos seguros de cuántos hermanos son (pueden ser bastantes más de cuatro, o incluso menos de cuatro).

La entrada termina con una viñeta sin personajes, que no deberá procesarse.

Salida

Se escribirá una línea por cada viñeta, indicando si *todos* los personajes que hay en ella pueden ser los Dalton (DALTON) o no (DESCONOCIDOS).

Entrada de ejemplo

```
4
1 2 3 4
5
10 1 2 3 4
4
1 1 2 2
0
```

Salida de ejemplo

```
DALTON
DESCONOCIDOS
DESCONOCIDOS
```




Búsqueda de polidivisibles

Los números *polidivisibles* son aquellos números que:

- Son mayores que cero.
- El número formado por su primer dígito es múltiplo de 1 (esto lo cumplen todos los números).
- El número formado por sus dos primeros dígitos es múltiplo de 2.
- El número formado por sus tres primeros dígitos es múltiplo de 3.
- El número formado por sus cuatro primeros dígitos...
- En general, el número formado por sus K primeros dígitos es múltiplo de K . Se debe asumir que los números se escriben en base 10 y sin ceros a la izquierda.

Por ejemplo, el número 2.016 es polidivisible, pues es mayor que cero y 2 es divisible por 1, 20 lo es por 2, 201 por 3 y, por último, el propio 2.016 es divisible por 4.

Sorprendentemente la cantidad de números polidivisibles no es infinito. De hecho hay únicamente 20.456 números polidivisibles, el mayor de ellos de 25 dígitos.

Dado que el mundo de las matemáticas no nos tiene muy acostumbrados a series finitas de números, para corroborar que efectivamente el conjunto total no es infinito queremos empezar por ser capaces de generar los números polidivisibles. Dado un número polidivisible N y una cantidad máxima de dígitos D , queremos obtener todos los números polidivisibles que comiencen por N y tengan como mucho D dígitos.

Entrada

La entrada estará compuesta por distintos casos de prueba. Cada uno de ellos se compone de una línea que contiene dos números, N ($0 < N < 10^{18}$) y D ($0 < D \leq 18$), que indican el comienzo (prefijo) del número y la cantidad máxima de dígitos de los polidivisibles a generar. Se garantiza que N es un número polidivisible y que D será siempre mayor o igual que el número de dígitos del propio N .

Salida

Para cada caso de prueba se escribirán todos los números polidivisibles que comiencen con N y tengan como mucho D dígitos. La lista se escribirá en *orden lexicográfico* (“alfabético”) y tras ella aparecerá una línea con tres guiones ---.

Entrada de ejemplo

```
2012 4
2016 6
```

Salida de ejemplo

```
2012
---
2016
20160
201600
201606
20165
201654
---
```