"First, solve the problem. Then, write the code" John Johnson.

Programa-Me 2019
Regional de Madrid y Terrassa
Problemas

Ejercicios realizados por

Universidad Complutense
de Madrid

Realizado en el **IES Virgen de la Paloma (Madrid)**
y en el **INS Nicolau Copèrnic (Terrassa)**

14 de marzo de 2019

14 de marzo de 2019
http://www.programa-me.com

Listado de problemas

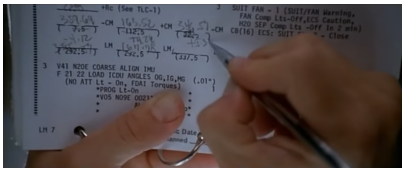
Autores de los problemas:

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

Revisores:

- Iván Cantón Sáez (Institut Nicolau Copèrnic – Terrassa)
- Cristina Gómez Alonso (Institut Sa Palomera – Blanes)
- Fede Grangé Borras (Institut Nicolau Copèrnic – Terrassa)
- Josep Queralt Molina (Institut Nicolau Copèrnic – Terrassa)

A
Houston, necesito una verificación



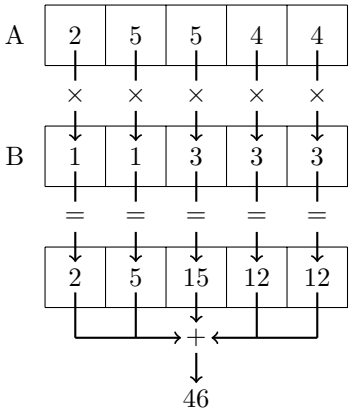
La séptima misión tripulada del programa Apolo de la NASA, Apolo 13, se hizo mundialmente famosa en 1970. Tras dos días alejándose de la tierra, una explosión en un tanque de oxígeno obligó a abortar la misión y regresar sin tocar la Luna. Las dificultades a las que se enfrentaron tanto los ingenieros de tierra como los propios astronautas y el hecho de que finalmente regresaran sanos y salvos hizo que muchos etiquetaran la misión como un *fracaso triunfal*.

En 1995 se estrenó una película con el mismo nombre en la que se relata la aventura. En un momento de la misma James Lovell, el comandante de la misión encarnado por Tom Hanks, hace unos cálculos de conversión de ángulos y pide a control de tierra que verifiquen los resultados. Acto seguido, dicta los datos y se ve como varios ingenieros en Houston comprueban que, efectivamente, los cálculos son correctos.

En lugar de conversiones de ángulos, el cálculo bien podría haber sido el producto escalar de dos vectores. Esta operación consiste en multiplicar los componentes de cada vector uno a uno y sumar sus resultados.

Por ejemplo, si tenemos dos vectores A y B con 5 componentes: A=(2,5,5,4,4) y B=(1,1,3,3,3), el producto escalar de ambos es:

$$A \cdot B = 2 \times 1 + 5 \times 1 + 5 \times 3 + 4 \times 3 + 4 \times 3 = 2 + 5 + 15 + 12 + 12 = 46$$



Eso sí, como en la situación de emergencia en la que estaban el tiempo apremiaba, en lugar de dictar el contenido de cada vector componente a componente, el astronauta podría "comprimirla" en bloques de valores consecutivos iguales. Así, la descripción del vector A anterior sería algo como "un dos, dos cincos y dos cuatros". En este caso no se ahorra nada, pero si el vector tiene diez millones de números en como mucho 30.000 grupos, la compresión puede significar que termine de dictarlos antes de la reentrada.

Entrada

La entrada estará compuesta por varios casos de prueba, cada uno ocupando tres líneas.

La primera línea contiene el número de grupos de números de los vectores A y B respectivamente. Ambos tendrán como mínimo un grupo, y no más de 30.000.

Las dos líneas siguientes contienen la descripción de los vectores A y B. Cada una contiene una pareja de números por cada grupo de valores consecutivos; el primero indica el número de repeticiones (al menos 1) y el segundo el valor que se repite (un número entre 0 y 50). Se garantiza que el número de componentes de ambos vectores coincide y que no excederá 10.000.000.

Salida

Por cada caso de prueba se escribirá una única línea con el producto escalar de A y B.

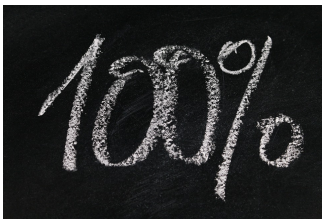
Entrada de ejemplo

```
3 2
1 2 2 5 2 4
2 1 3 3
1 1
1000000 40
1000000 40
```

Salida de ejemplo

```
46
1600000000
```

B
Döner sospechoso



Tras varias horas de turismo por una ciudad que prefiero no nombrar, decidí hacer una pausa rápida para comerme un *Döner* (también conocidos, incorrectamente, como *Döner Kebab*). Entré en un establecimiento y me sorprendió que uno de los que ofrecían era especialmente barato, aunque no ponía de qué carne estaba hecho.

Cuando le pregunté al encargado me respondió con dificultad. "Carne buena, buena, aquí, yo mezclo — dijo señalándose para darme la idea de que la carne la compraban y procesaban ellos mismos —. Conejo y caballo, 50%. Por cada conejo, un caballo, 50%".

Le di las gracias y me marché en busca de alguna otra cosa que llevarme al gaznate. Si esa era su forma de calcular porcentajes, prefería no saber nada más.

Entrada

La entrada comienza con un número que indica cuántos casos de prueba habrá que procesar.

Un caso de prueba está compuesto por dos números enteros entre 0 y 100 con el número de kilos de conejo y de caballo respectivamente que se han mezclado para preparar la carne que da vueltas en el asador. Se garantiza que la suma será mayor que cero.

Salida

Por cada caso de prueba, el programa escribirá el porcentaje de conejo que hay en la mezcla. El resultado se escribirá sin decimales, redondeado hacia abajo.

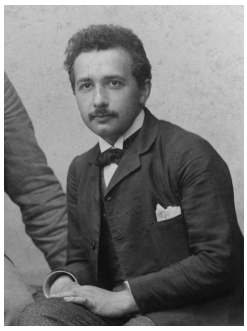
Entrada de ejemplo

```
3
53 47
3 5
86 97
```

Salida de ejemplo

```
53
37
46
```

C
Empleado del año



Cuando trabajaba en la oficina de patentes de Berna, en Suiza, Albert Einstein tuvo varios enfrentamientos con el director, quien terminó incluso jactándose de que había tenido que enseñarle a "expresarse correctamente".

Uno de esos enfrentamientos se debió al ridículo modo que el director usaba para contabilizar las patentes que cada empleado tramitaba al día. En lugar de registrar, sencillamente, el número, el director quería saber *la media* de patentes diarias que cada uno tenía desde el principio de año. Aseguraba que eso le daba una idea de qué empleado era el más efectivo a largo plazo.

El problema llegó cuando, para incentivar la productividad de los empleados, el director decidió dar un premio al empleado que más patentes hubiera registrado en un mismo día durante el último año. Como lo que se mantenía en el registro eran las medias, averiguar el valor diario suponía bastante trabajo.

Pero Albert se ofreció a calcularlo, porque estaba seguro de que ganaría el premio. Durante 1905 había dedicado bastantes días, en secreto, a escribir artículos científicos sobre física. Esos días no había tramitado patentes y, para ocultarlo, había tenido que trabajar a destajo a final de año para subir su media, tramitando una cantidad inusualmente alta de patentes. Además, se sentía capacitado para calcular lo que se le pedía rápidamente.

El premio al empleado del año fue lo primero que ganó gracias a esos artículos. Al año siguiente conseguiría por ellos el grado de Doctor en Zúrich, y en 1921 el Premio Nobel de Física. Además, 1905 pasó a conocerse como el "*annus mirabilis*" (año milagroso) por ellos.

Entrada

La entrada está compuesta por distintos casos de prueba, cada uno con los datos de un empleado.

Cada caso comienza con una línea con el número de días que ese empleado ha trabajado ese año ($1 \leq N \leq 365$). La línea siguiente contiene N números separados por espacios con la evaluación de la media de patentes por día desde principio de año. Se garantiza que esa media siempre es un número entero que no excederá 1.000.000.

Salida

Por cada caso de prueba se escribirá una línea con las patentes obtenidas por el empleado cada uno de los días en los que trabajó ese año.

Entrada de ejemplo

```
3
3 4 5
3
3 4 3
```

Salida de ejemplo

```
3 5 7
3 5 1
```

D

El mejor dato del paro

Cada vez que el Instituto Nacional de Estadística publica los datos del paro del último mes, comienzan las distintas interpretaciones de las tasas. Si el paro sube, los políticos en el gobierno buscarán la forma de maquillar el mal dato comparándolo, por ejemplo, con la subida producida el mismo mes del año anterior. Si el paro baja, los grupos de la oposición buscarán malos datos en tasas secundarias como la calidad de los nuevos contratos.

Dejando a un margen ese tipo de estrategias, una forma fácil de entender si el dato es bueno o no es la comparación con la llamada "serie histórica": ¿cómo de buena es la tasa comparándola con los últimos meses? Es innegable que si la tasa es la más baja de los últimos 15 meses, la economía no está tan mal.

Entrada

La entrada estará compuesta de distintos casos de prueba, cada uno representando la serie histórica de la tasa del paro de un país.

Para cada serie, aparecerán dos líneas. La primera tiene el número $1 \leq n \leq 300.000$ de meses a considerar y la segunda la tasa del paro de cada uno de los meses, separadas por espacios. Todas las tasas estarán entre 0 y 10^7.

La entrada termina con una serie histórica de 0 meses, que no deberá procesarse.

Salida

Por cada caso de prueba se escribirá una única línea con n números, uno por mes, separados por un espacio. El valor asociado al mes m indicará cuántos meses consecutivos llevaba ininterrumpidamente la tasa del paro con un valor peor (por encima) que el alcanzado ese mes.

El dato del mes m nunca podrá ser mayor que m - 1, pues ese es el número de valores anteriores. En ese caso la tasa es la mejor de toda la serie histórica hasta ese momento.

Entrada de ejemplo

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

Salida de ejemplo

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

E
Poniendo la mesa

Los invitados están al llegar y la mesa sigue sin poner. En la mesa de la cocina, tus padres han preparado ya todo lo que hay que trasladar: pilas de platos, cubiertos y copas. Por delante, un montón de paseos de la cocina al salón y muy poco tiempo.

Como las prisas en este caso no son buenas (que haya que pararlo todo para recoger del pasillo los pedazos de una copa rota puede ser desastroso) la única solución es *paralelizar* el trabajo. Tu hermano pequeño es el candidato perfecto para ayudarte.

Empezaréis por llevar todas las copas. Como son delicadas, tu hermano las llevará de una en una. Y para que se sienta "mayor" le has dicho que tú harás lo mismo: las llevarás también de una en una a no ser que el número de copas que queden en la cocina sea par. En ese caso en lugar de una, llevarás dos.

Si el primer paseo lo da tu hermano y os vais alternando los viajes, ¿cuántos necesitaréis para llevar todas las copas?

Entrada

La entrada está compuesta de distintos casos de prueba, cada uno en una línea. En cada una de ellas aparecerá el número de copas que hay que trasladar (hasta 10.000).

La entrada termina con una línea con un cero, que no deberá procesarse.

Salida

Por cada caso de prueba se indicará el número de paseos que hay que dar en total, teniendo en cuenta que el primer paseo lo da el hermano pequeño.

Entrada de ejemplo

```
1
3
0
```

Salida de ejemplo

```
1
2
```

F
El antepasado meticuloso

Al hacer limpieza en la vieja casa del pueblo, han aparecido unos papeles amarillentos dentro de un carcomido baúl. Debieron ser escritos muchos años atrás por alguien de la familia a quien ya nadie recuerda. Pero, fuera quien fuese, debió ser una persona adelantada a su tiempo, porque estaban escritos a máquina en una época en la que la mayoría de la gente del pueblo apenas era capaz de escribir su nombre, y con dificultad.

Lo más increíble de esos papeles era la delicadeza de la maquetación: salvo la última, todas las líneas de cada párrafo tenían la misma longitud. Dado que las letras y el espacio de la máquina de escribir tenían todos el mismo ancho, eso significaba que todas las líneas de un mismo párrafo tenían el mismo número de elementos. Además, quien escribiera estos viejos folios, ni siquiera se permitió poner dos espacios consecutivos. Por ejemplo, en uno de los folios estaba escrita la primera frase del Quijote ocupando tres líneas, las dos primeras con una longitud exacta de 80 letras:

```
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo
que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y
galgo corredor.
```

Hoy estamos acostumbrados a la *justificación completa*, pero conseguirla artesanalmente es toda una demostración de paciencia. Especialmente en esas viejas máquinas, donde no se cuenta con la *elasticidad de los espacios* para ajustar el ancho.

Entrada

La entrada comienza con un número indicando la cantidad de casos de prueba que habrá que procesar.

Cada caso de prueba comienza con dos números, el máximo número de letras y espacios que entra en una línea, $1 \leq L \leq 100$, y el número de palabras que tenemos que añadir en el párrafo, $1 \leq N \leq 5.000$.

A continuación aparecen, en otra línea, N números con la longitud, en letras, de cada palabra (como mucho 50).

Salida

Por cada caso de prueba el programa escribirá la mayor longitud de línea que podemos usar, menor o igual que L, de modo que todas las líneas tengan la misma longitud, salvo la última, que puede ser distinta, siempre que no supere la longitud de las demás.

Ninguna palabra puede cortarse en dos líneas. Además, entre cada par de palabras consecutivas se deberá contar el hueco ocupado por un espacio.

Entrada de ejemplo

```
3
10 4
3 4 4 3
10 4
3 4 4 4
85 33
2 2 5 2 2 7 2 4 6 2 6 10 2 2 5 6 3 5 2 7 2 3 2 5 2 10 6 8 5 5 1 5 9
```

Salida de ejemplo

```
8
IMPOSIBLE
80
```

G
Superagente 86

En 1965 los teléfonos eran muy diferentes a los de hoy: el tamaño, la calidad del sonido ¡y hasta la forma de marcar! En lugar de la marcación *por tonos* se usaba la marcación *por pulsos*. Por cada dígito se hacía girar un disco una cantidad que dependía del propio dígito. En función de la longitud del giro, se emitían por la línea uno o varios pulsos, que la centralita era capaz de contar para reconocer el número. El dígito 1 emitía un pulso, el 2 enviada dos, y así sucesivamente, con la peculiaridad de que el 0 emitía 10. Entre dígito y dígito se producía una pausa hasta que el usuario giraba de nuevo el disco.

Maxwell Smart, el Superagente 86, ha interceptado desde su zapatófono una llamada entre agentes de KAOS. Por desgracia, lo único que tiene es la longitud total de la marcación en número de pulsos. Sabe que entre dígito y dígito transcurre el tiempo asociado a un pulso, pero no sabe más. ¡Ni siquiera cuántos dígitos tenía el número!

Maxwell no se distingue por su audacia, y tiene la tonta idea de que solo con la longitud de la marcación podrá saber el número que ha marcado el agente de KAOS. La Agente 99 insiste en que eso es imposible, pero no consigue convencerle.

Entrada

La entrada comienza con un número indicando cuántos casos de prueba habrá que procesar. Cada caso de prueba es un número $1 \leq n \leq 10.000$ con la longitud, en pulsos, de la marcación.

Cada dígito i necesita i pulsos para ser marcado, salvo el `0`, que necesita 10. Además, el tiempo entre dígito y dígito cuenta como un pulso adicional.

Salida

Por cada caso de prueba el programa escribirá cuántos números de teléfono distintos existen que requieren ese número de pulsos para ser marcados. Como el número puede ser muy alto, se calculará módulo 1.000.000.007.

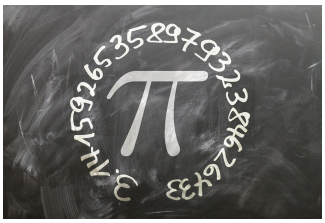
Entrada de ejemplo

```
4
1
2
3
45
```

Salida de ejemplo

```
1
1
2
42010781
```

H
Asamblea General de Indiana



La educación es una herramienta a menudo usada por los gobiernos para moldear, a medio plazo, a la ciudadanía. La negación del Holocausto o de la teoría de evolución son solo dos ejemplos.

Uno de los intentos más famosos de establecer una verdad científica por decreto ocurrió en 1897, cuando la Asamblea General de Indiana estuvo a punto de promulgar un proyecto de ley por el que se enseñaría a los alumnos un método de *cuadrar el círculo*. En aquel momento ya se había demostrado que era imposible, pero un aficionado a las matemáticas convenció a la asamblea de que había conseguido hacerlo y se lo ofreció "como una contribución a la educación que solo podrá ser utilizada por el Estado de Indiana en forma gratuita sin necesidad de pagar ningún tipo de royalties".

Ante semejante obsequio, el decreto pasó los primeros trámites de la asamblea a toda velocidad. Todo parecía presagiar que los pequeños indianeses estaban condenados a aprender que el valor de π era 3.2 y no el interminable 3.14... Si no hubiera sido por el profesor Waldo que paró a tiempo aquella locura, Indiana habría sido el único sitio del mundo donde el *día Pi* no se habría celebrado el 14 de marzo.

Entrada

Cada caso de prueba es el valor de π propuesto por algún matemático advenedizo. Tendrá siempre dos decimales detrás del punto.

La entrada termina con un `0.00` que no debe procesarse.

Salida

Por cada caso de prueba, el programa escribirá "`SI`" si, con ese supuesto valor de π, existiría un *día Pi*, y escribirá "`NO`" en caso contrario.

El *día Pi* existirá si la parte entera de la constante propuesta es un mes válido y la parte decimal un número de día válido para ese mes. Como queremos celebrar este significativo día todos los años, no se considerará válido el 29 de febrero.

Entrada de ejemplo

```
3.14
3.20
3.00
0.00
```

Salida de ejemplo

```
SI
SI
NO
```

Desbloqueo retro

Tu amigo Amadeo L. D. Zings vive en el pasado. Añora los viejos tiempos de *disquettes*, chirriantes *módems* a 56 Kbps y monitores CRT. ¡Qué tiempos aquellos en los que para ver el siguiente capítulo de una serie había que esperar una larga semana de incertidumbre!

Fiel a sus principios, sigue utilizando su viejo móvil; dice que la sensación de pulsar botones físicos es preferible a la frialdad de una pantalla, por muchos colores y aplicaciones que tenga. Pero reconoce que le gustaría que su móvil no pudiera usarlo cualquiera, y tener un patrón de desbloqueo como ha visto en alguno de esos teléfonos mucho más modernos que el suyo. Está convencido de que la pantalla del trasto al que llama teléfono podría mostrar, con letras y símbolos, la "matriz" sobre la que pintar el patrón. Luego, simularía el desplazamiento utilizando el teclado numérico a modo de *flechas del cursor*, con el `1` indicando que quiere desplazarse hacia arriba y a la izquierda, por ejemplo.

Como le ves un poco desesperado, has decidido buscar información sobre como se desarrollaba para esos viejos aparatos para darle una sorpresa en su próximo cumpleaños.

Entrada

Cada caso de prueba comienza con dos números, $1 \leq F, C \leq 30$ indicando, respectivamente, el número de filas y de columnas que tiene la matriz de desbloqueo que quieres pintar en la pantalla. A continuación vienen dos números, $1 \leq f \leq F$ y $1 \leq c \leq C$, indicando la fila y la columna en la que comienza el patrón. La última línea del caso de prueba es una sucesión de dígitos indicando el desplazamiento en la matriz. Cada dígito indica una dirección de desplazamiento de acuerdo a su posición en el teclado numérico de un teléfono, de modo, por ejemplo, que los números del 1 al 3 indican un desplazamiento hacia arriba, y, además, el 1 desplaza hacia la izquierda y el 3 hacia la derecha. La secuencia de dígitos, de no más de 1.000, termina con un `5` que marca el final. Se garantiza que la posición actual se mantiene siempre dentro de los límites.

Amadeo es un pelín paranoico con la seguridad, y es posible que su patrón pase varias veces por los mismos sitios.

La entrada termina con dos ceros, que no deben procesarse.

Salida

Por cada caso de prueba, el programa escribirá el estado final del patrón.

El patrón estará encerrado en un marco de guiones y líneas verticales, con signos de suma en las esquinas. Cada posición "pulsable" está separada de las demás por un hueco (espacio o línea en blanco). Las posiciones no tocadas se pintarán con un punto ("`.`") y las tocadas con la letra o mayúscula ("`O`").

Las posiciones tocadas se unirán con líneas ("`-`", "`|`", "`\`", "`/`" o "`X`") en función del desplazamiento realizado.

Entrada de ejemplo

```
3 3
1 1
687615
2 4
1 1
9392735
0 0
```

Salida de ejemplo

```
+-----+
|O-O .|
|  |  |
|O O .|
| X   |
|O-O .|
+-----+
+-------+
|O . O O|
| \ / X||
|. O O O|
+-------+
```

J

¿Dónde está la bolita?

Cuando hace años estuve en Nueva York vi en varias de sus calles más concurridas a trileros y sus ganchos buscando incautos para sisarles unos dólares. Algunos usaban cartas, y otros los famosos tres cubiletes. Debajo de uno de ellos meten una bolita y luego los intercambian rápidamente entre sí. Al terminar, aceptan apuestas sobre debajo de qué cubilete está la bolita. Por muy atento que se esté, ¡siempre consiguen ganar sus apuestas!

En breve volveré a Nueva York, y me estoy preparando para darles una lección. Llevo meses haciendo una aplicación para móvil que analiza las imágenes de la cámara y hace el seguimiento de las posiciones de los cubiletes, y funciona bastante bien. Cada vez que se realiza un intercambio, me dice las posiciones de los dos cubiletes que se han movido. Ya solo me falta que, sabiendo la posición inicial de la bolita, me diga debajo de qué cubilete ha quedado al final.

Entrada

Cada caso de prueba comienza con dos números, el primero $2 \leq N \leq 100.000$ indicando cuántos cubiletes está usando el trilero, y el segundo $1 \leq I \leq N$ con la posición que ocupa el cubilete debajo del que comienza la bolita. A continuación aparece una segunda línea con una sucesión de parejas de números, todos entre 1 y N. Cada pareja indica la posición de los dos cubiletes que se intercambian, y serán siempre distintos. La línea termina con dos ceros, que indican que el trilero ha dejado de mover los cubiletes.

La entrada termina con dos ceros.

Salida

Por cada caso de prueba, el programa escribirá la posición del cubilete debajo del que ha terminado la bola tras todos los movimientos del trilero.

Entrada de ejemplo

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

Salida de ejemplo

```
1
2
```