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

ProgramaMe 2024
Regional de Cádiz

Problemas

Ejercicios realizados por

Universidad Complutense
de Madrid

Realizado en el **IES Rafael Alberti**
10 de abril de 2024

10 de abril de 2024
https://www.programame.com

Listado de problemas

Autores de los problemas:

- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)

Revisores:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)

Imágenes y fotografías:

- Problema A, "*El campus de Jussieu*": fotografía de de alchetron.com. Imagen del plano basada en un plano de la Wikipedia.
- Problema H, "*Sam Loyd*": creación propia.
- Problema J, "*Una X marca el lugar*": fotograma de la película "Indiana Jones y la ultima cruzada" en la que se ambienta el enunciado.
- Resto de imágenes: gratis para usos comerciales; no necesitan reconocimiento (licencia Pixabay)

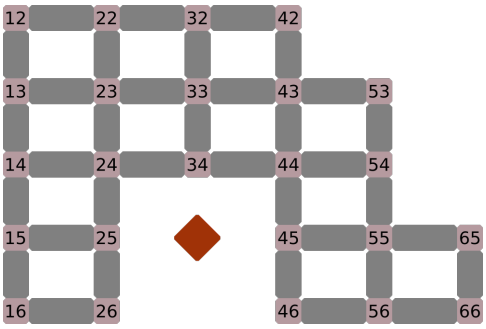
A
El campus de Jussieu

En la orilla izquierda del Sena, en el corazón de París, se encuentra el campus de Jussieu. Lo comparten varias instituciones, entre otras la que fue la Universidad de París VI Pierre et Marie Curie, hoy refundada en la *Universidad de la Sorbona*.

El campus está compuesto por una *rejilla* de edificios de 6 plantas con soportales peatonales. En las intersecciones se encuentran las escaleras y ascensores para acceder a los hasta 4 edificios que conectan.

Para facilitar la orientación, las intersecciones están numeradas de acuerdo a su posición en la rejilla con dos números indicando la "columna" y la "fila" en la que están. Como el número de columnas y filas es menor que 10, los dos números se fusionan en uno solo, de dos dígitos. Así la intersección `23` se refiere a la situada en la "columna" `2`, "fila" `3`. Los edificios, a su vez, se identifican por la pareja de intersecciones que conectan.

Pese a la regularidad de su diseño, no en todos los lugares donde debería haber una intersección hay un acceso con escaleras y ascensor, lo que reduce el número de edificios del complejo.



Entrada

Cada caso de prueba comienza con dos números $2 \leq c, f \leq 100.000$ indicando, respectivamente, el número de columnas y filas de edificios en la *rejilla*. A continuación, en la misma línea, aparece un tercer número $0 \leq \mathrm{h} < \min(250.000,\ c \times f)$ indicando el número de *huecos* en las intersecciones, es decir en cuántas intersecciones falta el acceso.

A continuación aparecen h pares de números indicando cada uno el número de columna (entre 1 y c) y número de fila (entre 1 y f) de una intersección ausente. Como el tamaño de la rejilla puede ser grande, ambos números aparecen separados por espacio, en contra de lo que ocurre en la numeración del campus de Jussieu. Se garantiza que cada intersección ausente aparece únicamente una vez.

La entrada termina con tres ceros.

Salida

Por cada caso de prueba el programa escribirá el número de *edificios* que tiene la rejilla del campus. Para que un edificio se construya deben existir *las dos intersecciones que conecta*. En particular, puede ocurrir que exista una intersección sin edificios adyacentes porque faltan las cuatro intersecciones contiguas.

Entrada de ejemplo

```
2 2 0
2 2 2
2 1 1 2
6 6 12
1 1 2 1 3 1 4 1 5 1 6 1 5 2 6 2 6 3 6 4 3 5 3 6
0 0 0
```

Salida de ejemplo

```
4
0
35
```

B
Checkmult

Para detectar errores de transmisión en redes de datos, es habitual calcular un *checksum* o "suma de verificación" a partir de los datos transmitidos, y enviar también el valor obtenido. En el otro extremo, se recalcula el valor con los datos recibidos y si el resultado no coincide con el esperado se asume que se ha producido un error de transmisión y se toman las medidas oportunas.

Hay muchas formas de calcular el *checksum*, que se conocen de forma conjunta como *funciones de redundancia*. La idea es tan útil que se utiliza en muchos otros lugares, como detección de errores en dispositivos de almacenamiento, o incluso criptografía y firmas digitales.

Inspirado por el término inglés *checksum*, Asier Rorde Tecta ha inventado su propia función de redundancia y la ha llamado *checkmult*. Para calcular el valor asociado a un número, coge todos sus dígitos que no sean cero y los multiplica entre sí, repitiendo el proceso hasta terminar con un solo dígito. Además, cuenta el número de ceros que se encuentra durante el proceso, incluídos los del número original. Al final, termina con dos valores, el dígito resultado de las sucesivas multiplicaciones y el número de ceros vistos. Pone ambos juntos y ese es el valor del *checkmult*.

Entrada

El programa deberá procesar múltiples casos de prueba, cada uno un número $1 \leq n \leq 10^9$.

La entrada termina con un 0, que no debe procesarse.

Salida

Por cada caso de prueba se escribirá el *checkmult* del número del caso de prueba. Se garantiza que el resultado siempre tendrá dos dígitos, pues el número de ceros encontrados durante el proceso nunca será mayor que 9.

Entrada de ejemplo

```
1
20
506
0
```

Salida de ejemplo

```
10
21
32
```

Notas

El último ejemplo, `506` tiene un primer cero. La multiplicación de sus dígitos que no son cero, `5` y `6` da `30`, que añaden un segundo cero. Al quitarlo, se queda un único dígito, el `3`. El *checkmult* es la concatenación de este dígito final y del número total de ceros encontrados, de ahí el `32`.

C
Coetáneos

Para entender mejor la historia, la historia del arte y sus relaciones, es importante conocer el contexto de los personajes históricos y de los artistas, para comprender sus influencias y la situación social, científica y técnica que vivieron.

Por ejemplo, la música de Mozart (1756–1791) tuvo mucha influencia en la de Beethoven (1770–1827) y se cree que ambos autores llegaron a conocerse en Viena en 1787. La música de Beethoven también se vio muy influenciada por la época turbulenta que vivía Europa. Su sinfonía número 3, "La Heroica", estuvo originalmente dedicada a Napoleón (1769–1821), aunque cuando éste se autoproclamó emperador, Beethoven tachó su nombre de la partitura.

Solo unas décadas antes, la situación en Norteamérica también había sido convulsa, con la Guerra de Independencia y la creación de Estados Unidos, país del que Benjamín Franklin (1706–1790) es considerado uno de los padres fundadores. Político, científico e inventor, fue un *polímata*, igual que su coetáneo Sir Isaac Newton (1642–1727).

Entrada

La entrada comienza con un número indicando cuántos casos deberán ser procesados. Cada caso de prueba son dos parejas de números, cada una indicando el año de nacimiento y muerte de una persona. Se garantiza que, en los datos de entrada, ninguna persona nace antes del año 1 o después de 1900, ni la resta entre su año de muerte y nacimiento es mayor que 100.

Salida

Por cada caso de prueba se indicará el número de años en las que ambas personas coincidieron en el tiempo. Se asume que la fecha de nacimiento es a principios de año y la de muerte a finales, de modo que si una de las personas muere el mismo año que la otra nace se considera que coincidieron un año.

Entrada de ejemplo

```
4
1756 1791 1770 1827
1756 1791 1706 1790
1642 1727 1769 1821
1706 1790 1642 1727
```

Salida de ejemplo

```
22
35
0
22
```

D
Eligiendo vagones

Como viaje de fin de curso, se han juntado varias clases y se han ido de viaje por Europa aprovechando los descuentos del Interrail. Cada vez que tienen que coger billetes para un trayecto ocurre lo mismo: quieren ir todos juntos, pero, siendo tantos, resulta misión imposible.

Como solución de compromiso, se conforman con estar todos en vagones contiguos, siempre que sea la menor cantidad de vagones posible, para que si alguien en un extremo quiere hablar con otra persona que está sentada en el extremo contrario tenga que andar poco por los pasillos del tren.

Después de haber visitado la Torre Eiffel y los Campos Eliseos están comprando los billetes para ir a Berlín. Saben cuántos son, y cuántos huecos libres hay en cada vagón, y tienen que decidir qué asientos coger.

Entrada

Cada caso de prueba comienza con una primera línea con dos números. El primero, $1 \leq p \leq 10^8$, indica la cantidad de personas que viajan juntas en el grupo. El segundo, $1 \leq n \leq 500.000$, indica el número de vagones del tren.

La segunda línea contiene n números separados por espacio con la cantidad de asientos libres de cada vagón, empezando por el vagón más cercano a la locomotora y terminando por el más alejado. Ningún número es mayor que 10.000 y la suma total de huecos libres no supera 2×10^9.

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

Salida

Por cada caso de prueba el programa escribirá el mínimo número de vagones adyacentes necesarios para que todos los estudiantes del grupo puedan viajar, junto con la posición en el tren del primer vagón que ocuparán. Los vagones se numeran empezando por el 1, el situado junto a la locomotora, hasta el n.

Si hay más de una posible solución, se elegirá aquella más cercana a la locomotora. Si no hay solución se escribirá `NO ENTRAN`.

Entrada de ejemplo

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

Salida de ejemplo

```
2 3
3 2
NO ENTRAN
```

E
Guijarros en la balanza

En casa de los abuelos del pequeño Justo Enmedio hay una balanza antigua que ha permanecido en la familia durante varias generaciones. Aunque hoy es un mero objeto decorativo, durante muchos años fue una herramienta importante en la pequeña tienda de ultamarinos que regentaban.

A Justo le entretiene jugar con los guijarros de un río cercano que colecciona, intentando repartirlos en los dos platillos de la balanza de tal forma que ambos lados pesen lo mismo. Pero le cuesta mucho.

Entrada

Cada caso de prueba comienza con un número $2 \leq n \leq 12$ indicando el número de guijarros que Justo se ha llevado a casa de los abuelos. A continuación, en otra línea, aparecen los pesos de las n piedrecillas, todos positivos. Se garantiza que la suma de todos ellos es par y no excede 10^9.

La entrada termina con un `0` que no debe procesarse. Nunca habrá más de 10.000 casos en la entrada.

Salida

Por cada caso de prueba, el programa escribirá `SI` si es posible repartir los n guijarros en dos bloques de modo que ambos pesen lo mismo, y `NO` si resulta imposible.

Entrada de ejemplo

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

Salida de ejemplo

```
SI
NO
SI
SI
```

F
Palíndromos ocultos

A Ana Neuquén le encantan los palíndromos, esas palabras que se leen igual de izquierda a derecha que de derecha a izquierda. *Lamentablemente*, el español es un *rollo* porque se *conocen* muy pocos. Pero hay palabras que, *reordenando* sus letras, pueden convertise en palíndromos.

Cuando otros hablan, siempre se oye un *susurro* porque Ana *murmura* palabras incomprensibles intentando crear esos palíndromos. Para *alegrarle* su futuro cumpleaños y que se pueda *entretener*, has decidido *regalarle* un programa que, dada una palabra, *intente* reordenar sus letras y las escriba de modo que formen un palíndromo, aunque sea impronunciable.

Entrada

Cada caso de prueba es una palabra de no más de 20 letras minúsculas del alfabeto inglés.

Salida

Por cada caso de prueba se escribirá una palabra con las mismas letras que la original pero reordenadas de modo que el resultado sea un palíndromo, aunque no signifique nada o, incluso, no pueda leerse. Si hay varias opciones se escribirá la que sea menor lexicográficamente.

Si no se puede formar ningún palíndromo se escribirá `NO HAY`.

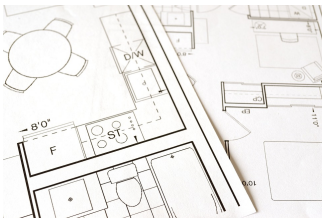
Entrada de ejemplo

```
lamentablemente
rollo
conocen
reordenando
ala
palindromo
```

Salida de ejemplo

```
aeelmntbtnmleea
lorol
cnoeonc
denoraroned
ala
NO HAY
```

G
Pelotazo urbanístico



Moisés Eña está dispuesto a hacerse rico gracias al ladrillo. Ha comprado un gran terreno en un pueblecito bien comunicado y planea hacer una urbanización de lujo. Hace algún tiempo intentó forrarse haciendo pisos baratos para gente con recursos modestos y el tema no acabó muy bien.

Ahora ha puesto su punto de mira en gente más acaudalada y cree que la mejor forma de llamar su atención es haciendo viviendas unifamiliares en parcelas lo más grandes posible. Por eso quiere dividir su terreno en *el menor* número de parcelas posible, para que sean cuadradas y tan grandes como se pueda, siempre que todas sean iguales y cubran el terreno completo sin dejar espacio sin utilizar.

Entrada

La entrada comienza con un número indicando cuántos casos de prueba deberán procesarse (como mucho 50.000). Cada uno está compuesto por dos números $1 \leq w, h \leq 10^9$ indicando las dimensiones del terreno comprado por Moisés.

Salida

Por cada caso de prueba se indicará el *mínimo* número de parcelas en las que puede dividirse el terreno de Moisés de modo que todas sean iguales y cuadradas (mismo ancho que alto) y no quede ningún espacio del terreno libre.

Entrada de ejemplo

```
4
10 10
5 10
8 6
46349 46351
```

Salida de ejemplo

```
1
2
12
2148322499
```

H
Sam Loyd

Sam Loyd es considerado uno de los creadores de puzzles más importantes de Estados Unidos, si no el más importante, tal y como lo nombró Martín Gardner en 1957. Pero, al mismo tiempo, fue mentiroso y oportunista, asignándose la autoría de puzzles que no inventó.

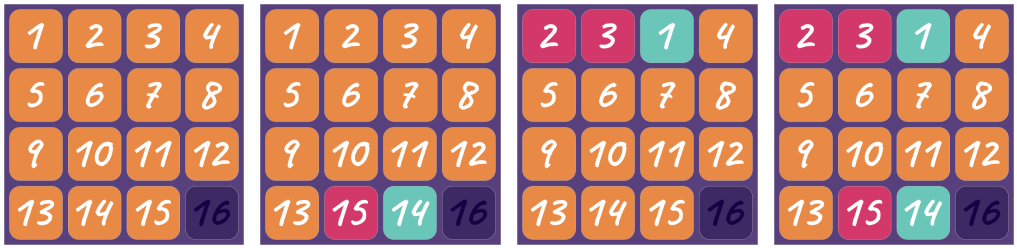
Uno de ellos fue el *juego de las 15 baldosas*. Consiste en un "tablero" con 4×4 compartimentos en el que hay colocadas 15 "baldosas" que se pueden deslizar unas sobre otras aprovechando el hueco que dejan. Las baldosas comienzan en una determinada configuración y el objetivo es conseguir recolocarlas en otra.

La primera versión del juego fue creada por Palmer Chapman durante la década de 1870. En 1880 su popularidad estalló en Estados Unidos, convirtiéndose en una moda maniática que lo hizo aparecer en infinidad de lugares, como ocurriría con el cubo de Rubik un siglo después. Aun así, no fue hasta 1891 cuando Sam Loyd empezó a asegurar que había sido invención suya, cosa que afirmó hasta su muerte en 1911.

Al menos sí colaboró al resurgimiento del juego cuando ofreció un premio de 1.000 dólares (una pequeña fortuna para el momento) a quien resolviera el que llamó *puzzle 15–14*. Consistía en una distribución inicial de las piezas en la que la número 14 y la número 15 estaban invertidas y se pedía que se colocaran todas en orden.

También en esto fue un tramposo. En 1879 se había demostrado matemáticamente que *era imposible* resolverlo. Las configuraciones de las baldosas forman dos conjuntos independientes. Es posible desplazar las baldosas de cualquier configuración a cualquier otra *del mismo conjunto*, pero es imposible pasar de una configuración de un conjunto a otra del contrario.

Una configuración incial puede llevarse a una configuración final si *la paridad del número de inversiones es la misma*. El *número de inversiones* se calcula colocando todas las baldosas seguidas por filas, poniendo un 16 en el lugar que ocupa el hueco, y contando la cantidad de parejas de baldosas en las que una baldosa con un número más grande aparece antes que una con un número más pequeño. Por ejemplo, la configuración inicial con todas las fichas colocadas en secuencia (1, 2, . . . , 16, figura a) no tiene inversiones y por tanto tiene paridad par. En la configuración tramposa de Sam Loyd (1, 2, . . . , 13, **15**, **14**, 16, figura b) la pareja de baldosas 15–14 supone una inversión (la única en este caso) y por tanto la configuración tiene paridad impar. Al tener paridades contrarias, una configuración no es alcanzable desde la otra.



(a) Cero inversiones (b) Una inversión (c) Dos inversiones (d) Tres inversiones

La figura c muestra una configuración con dos inversiones (la formada por las parejas 2–1 y 3–1), lo que la categoriza como paridad par y por tanto es alcanzable desde la configuración inicial. Por último, la figura d tiene tres inversiones, las dos de la figura c y la formada por la pareja 15–14. Tiene por tanto paridad impar y no es alcanzable desde la configuración inicial, pero sí desde la configuración tramposa de Sam Loyd.

Entrada

Cada caso de prueba está compuesto por una permutación de los números del 1 al 16 indicando una configuración del puzzle de las 15 baldosas.

Salida

Por cada caso de prueba, el programa escribirá `SI` si la configuración es alcanzable desde la posición inicial del tablero con todas las baldosas en orden. Si no es alcanzable se escribirá `NO`.

Entrada de ejemplo

```
1 2 3 4 5 6 7 8 9 10 11 12 13 15 14 16
2 3 1 4 5 6 7 8 9 10 11 12 13 14 15 16
2 3 1 4 5 6 7 8 9 10 11 12 13 15 14 16
```

Salida de ejemplo

```
NO
SI
NO
```

Toques de balón

Tu hermana pequeña está todo el día con un balón de fútbol al lado. Le encanta pasar las horas dando toques, del pie a la cabeza, de la cabeza a la rodilla, de la rodilla parada en la nuca, y otra vez. Puede dar cientos de toques al balón antes de que éste toque el suelo.

Una pregunta que todo el mundo la hace es cuál es su record de toques, pero no lo sabe. Aunque muchas veces se ha propuesto contarlos, la realidad es que pronto se aburre y pierde la cuenta.

Para ayudarla, has montado un montón de sensores en el patio y la has pedido que esté un rato con el balón. Ahora tienes una larga cadena de caracteres, con una 'T' indicando un toque de tu hermana con alguna parte del cuerpo al balón (pie, rodilla, cabeza, ...) o una 'S' cuando el balón toca el suelo. Lo siguiente es averiguar la longitud de la secuencia más larga compuesta únicamente de letras 'T'.

Entrada

La entrada comienza con un número indicando cuántos casos de prueba se deben procesar. Cada uno está compuesto por una cadena de no más de 1.000 caracteres, todos ellos 'T' o 'S'.

Salida

Por cada caso de prueba el programa escribirá la secuencia consecutiva más larga de 'T' del caso de prueba.

Entrada de ejemplo

```
3
TTTST
STSTTSSS
TTTTTT
```

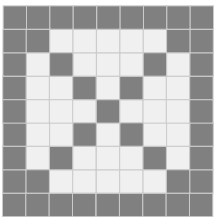
Salida de ejemplo

```
3
2
6
```

J
Una X marca el lugar

En la película *Indiana Jones y la última cruzada*, Indy encuentra la entrada a la tumba del caballero gracias a una gran X marcada en el suelo de losetas de una biblioteca veneciana.

En la película original, las losetas eran de marmol antiguo, bien pulido y trabajado. Jorge Lucas está haciendo un corto sobre dos hermanos, Ind y Ana Youns, y quiere replicar la escena, pero no tiene presupuesto para un suelo tan caro. Ha decidido que en la sala cuadrada donde tiene que estar la gran `X` pondrá losetas blancas y negras. Las losetas junto a la pared serán negras, las diagonales también, y el resto serán blancas.



¿Cuántas losetas necesita de cada color?

Entrada

La entrada comienza con un número indicando cuántos casos de prueba deben procesarse. Cada caso de prueba es un número impar entre 5 y 39.999 indicando el número de losetas a lo ancho (y a lo alto) que entran en la sala cuadrada donde se grabará la escena.

Salida

Por cada caso de prueba se escribirá el número de losetas negras y blancas, separadas por un espacio, que se deben comprar para poder crear la `X` en el suelo.

Entrada de ejemplo

```
3
7
9
11
```

Salida de ejemplo

```
33 16
45 36
57 64
```