



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

Programa-Me 2019

Calentamiento pre-navideño

Problemas



Ejercicios realizados por

ceedcv

CENTRE ESPECÍFIC
D'EDUCACIÓ A DISTÀNCIA DE
LA COMUNITAT VALENCIANA

Centro Específico de Educación a Distancia
Comunidad Valenciana

Concurso on-line sobre **¡Acepta el reto!**

<https://www.aceptaelreto.com>

19 de diciembre de 2018



**Universidad
Complutense
Madrid**

Listado de problemas

A Tu amigable vecino	3
B Cameos de Stan Lee	5
C Lucha de egos	7
D A la captura del Doctor Muerte	9
E Conjuritis	11
F El hombre sin miedo y sin radar	13

Todos los problemas están ambientados en el “universo Marvel”, en homenaje a **Stan Lee**, padre de infinidad de sus superhéroes, fallecido el 12 de noviembre de 2018. Los personajes e imágenes son propiedad de sus respectivos dueños. Las historias de las ambientaciones de los problemas se inspiran en ellos, pero son inventadas.



Autor de los problemas:

- Sergi García Barea (Centro Específico de Educación a Distancia Comunidad Valenciana, CEEDV)
- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Alfredo Oltra Orengo (Centro Específico de Educación a Distancia Comunidad Valenciana, CEEDV)



Tu amigable vecino

El *Duende Verde*, acérrimo enemigo de nuestro amigable vecino *Spiderman*, ha colocado dos bombas atómicas en pleno Nueva York, en dos rascacielos distintos de la Quinta Avenida. Afortunadamente, *Spiderman* ha capturado al *Duende Verde* y, a base de mamporros y chistes malos, le ha sonsacado donde están. Tras entregar al *Duende Verde* a las autoridades, *Spiderman* se dispone a desactivar las dos bombas, pero quiere hacerlo en el menor tiempo posible, ya que su vida como *Peter Parker* le obliga a visitar a Mary Jane y su tía May.



La comisaría donde está ahora mismo se encuentra también situada en la Quinta Avenida, pero está un poco aturdido por la lucha y no sabe qué bomba desactivar primero para tener más posibilidades de llegar a tiempo para la cena.

Entrada

La entrada comienza con un número indicando cuántos casos de prueba habrá que procesar. Cada uno está compuesto por tres números, $0 \leq S, A, B \leq 10.000$. El primero, S , proporciona la posición, en la Quinta Avenida, de *Spiderman*. Los dos siguientes, A y B , indican las posiciones de las bombas atómicas.

Las posiciones se indican como la distancia, en metros, desde el inicio de la avenida, en el Washington Square Park.

Salida

Por cada caso de prueba, el programa escribirá la distancia mínima, en metros, que tendrá que recorrer *Spiderman* para desactivar las dos bombas colocadas por el *Duende Verde*.

Entrada de ejemplo

```
2
100 150 200
300 100 700
```

Salida de ejemplo

```
100
800
```




Cameos de Stan Lee

Aparte de por haber creado infinidad de superhéroes, Stan Lee es recordado por los *cameos* cinematográficos que hizo en multitud de películas y series.

Quizá por estar más escondidos, son menos conocidos sus *cameos* en los *comics* de los que era autor. Por ejemplo, en una determinada viñeta podía leerse “¡Que me diga dónde **están**, le **exijo!**” Un ojo entrenado, detecta su aparición sin ningún tipo de indicación.

Algunas veces se camuflaba más, partiendo su nombre con letras arbitrarias, como en “¡Eres **tan** lento que te ganaría una oruga!” o incluso en “**Salta**, no le temas”.

Los aficionados a los superhéroes releen sus comics una y otra vez buscando todos esos *cameos*, pero es mucho más práctico automatizar la tarea.



Entrada

La entrada comenzará con una línea con el número de casos de prueba que hay que procesar.

Cada caso de prueba es una línea de texto de hasta 10.000 letras, incluyendo mayúsculas, minúsculas y símbolos de puntuación. Por simplicidad, no aparecerán vocales acentuadas.

Salida

Por cada caso de prueba, el programa escribirá el número de *cameos* de Stan Lee, es decir, el número de veces que aparece su nombre, sin tener en cuenta las mayúsculas ni los espacios, y con la posibilidad de que haya letras, símbolos y espacios separando las letras originales.

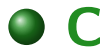
Stan Lee era único, por lo que antes de empezar a buscarle una segunda vez ¡tiene que haber terminado de salir la primera!

Entrada de ejemplo

```
5
Que me diga donde estan, le exijo!
Eres tan lento que te ganaria una oruga. Calla y salta, no le temas.
Esto es canela, verdad que si?
Stan Stan Lee Lee
Stan Loo
```

Salida de ejemplo

```
1
2
1
1
0
```

Lucha de egos

Se veía venir. La fila de superhéroes entrando en el despacho de Stan Lee a quejarse se divisaba desde Asgard. Y es que todos querían ser más poderosos que sus compañeros de trabajo. Que si Luke Cage decía que era más fuerte que Hulk, que si Iron Man quería ser más seductor que Daredevil, que si Jessica Jones quería ser más impertinente que la Viuda negra...

Stan Lee propuso que esta lucha de egos se zanjara mediante batallas justas, pactadas y equilibradas. Equilibrarlas supuso un quebradero de cabeza para todos.

Al final, el sindicato THANOS (*Trabajadores Heroicos Aunque Necesitados de Organización Sindical*) tuvo que intervenir, y acordó con Stan Lee crear un sistema que emparejara a los superhéroes en función de sus preferencias.

El propio Stan Lee asignó a cada superhéroe un nivel de destreza entre 1 y 5. Por su parte, los superhéroes indicaron sus preferencias de la destreza que debía tener su oponente:



- Si querían enfrentarse con superhéroes con un nivel concreto, indicaban “=N”, con N entre 1 y 5.
- Si, por el contrario, estaban dispuestos a enfrentarse con superhéroes que tuvieran un nivel mayor, decían “>N”, con N entre 0 y 4.
- Los más cobardes podían indicar “<N” para pedir superhéroes de nivel inferior a N, entre 2 y 6.

La lucha se ha convertido en tradición y se hace todos los años. Tienes la valoración de cada superhéroe y sus preferencias y debes emparejarlos. Cada superhéroe luchará con el primero que vaya después en la lista y que cumpla las restricciones de ambos, siempre que no haya sido emparejado previamente. Con este mecanismo de emparejamiento, es posible que algunos superhéroes se queden sin emparejar.

Entrada

El programa deberá procesar múltiples casos de prueba leídos de la entrada estándar.

Cada uno comienza con el número N de superhéroes que competirán en el *torneo de egos* (como mucho 100).

A continuación aparece una línea de N números entre 1 y 5 con la destreza de cada uno. Después, seguirá una segunda línea con sus preferencias en el mismo orden, y de la forma <N, =N y >N.

La entrada termina con un caso sin superhéroes, que no debe procesarse.

Salida

El programa escribirá, por cada caso de prueba, los emparejamientos realizados en líneas independientes. Cada superhéroe se identifica por su posición en la lista de la entrada (valores entre 1 y N).

La asignación de emparejamientos se realiza buscando desde el primero hasta el último el primer superhéroe que vaya después en la lista que permita que ambos estén conformes con el enfrentamiento. Cada superhéroe lucha solo una vez, por lo que si un superhéroe es asignado en un duelo, no se le busca otro.

Si el proceso termina sin ningún emparejamiento se escribirá NO HAY.

Después de cada caso de prueba se escribirá “---”.

Entrada de ejemplo

```
3
3 1 4
<3 >3 =4
6
3 1 4 4 5 3
=3 >0 <5 =4 <3 >2
0
```

Salida de ejemplo

```
NO HAY
---
1 6
2 3
---
```



A la captura del Doctor Muerte

Los 4 Fantásticos han dado con el escondrijo del *Doctor Muerte* en la lejana Latveria y van a intentar capturarlo para que pague por sus numerosos crímenes. El *Doctor Muerte*, con el fin de protegerse, se ha rodeado de sus sirvientes a modo de escudos humanos.



Reed Richards, líder de *Los 4 Fantásticos*, posee muchas armas (de un solo uso) para enfrentarse al *Doctor Muerte*, pero quiere utilizarlas con sabiduría: cada arma ocasiona un daño distinto a inocentes y a villanos.

Ante la situación actual, Reed Richards seleccionará siempre primero las armas que causen un menor daño a inocentes y, en caso de empate, aquellas que causen mayor daño a villanos. Si hay dos armas iguales, se usará primero la que vaya antes en la lista.

Sabiendo el daño que hay que infligir al *Doctor Muerte* para vencerle, la forma en la que se elegirán las armas y recordando que solo pueden usarse una vez, es el momento de elegir las armas a utilizar.

Entrada

La entrada estará compuesta de múltiples casos de prueba, cada uno ocupando varias líneas.

Un caso de prueba comienza con dos números $1 \leq V \leq 10.000$ con la vitalidad del *Doctor Muerte* y $1 \leq N \leq 1.000$ con el número de armas disponibles.

A continuación aparecerán N líneas. Cada una indica las características de un arma con dos números (entre 0 y 10.000) con el daño que hace a inocentes y a villanos, respectivamente.

La entrada termina con un cero.

Salida

Por cada caso de prueba se escribirá una línea indicando la sucesión de armas escogidas hasta la derrota del *Doctor Muerte*. Las armas se identifican por su posición en la lista de la entrada (números entre 1 y N), y aparecerán separadas por un espacio. Ten en cuenta que el modo de elegir las armas podría no ser el que menos daño total haga a inocentes!

Si es imposible derrotar al *Doctor Muerte* se escribirá "MUERTE ESCAPA".

Entrada de ejemplo

```
100
4
9 100
8 90
2 11
8 89
100
1
10 99
0
```

Salida de ejemplo

```
3 2
MUERTE ESCAPA
```


● E Conjunitis



El *Doctor Extraño* es un poderoso hechicero entrenado por *El Anciano*, y sucesor del título de “Hechicero supremo” a su muerte. A pesar de ser un superhéroe muy poderoso, ahora está en un apuro.

El Barón Mordo ha sometido al *Doctor Extraño* a un hechizo llamado “*conjunitis*” que le impide pensar en conjuros con claridad y en su mente se mezclan conjuros reales y conjuros falsos.

Tras invocar una sucesión de conjuros, el *Doctor Extraño* está convencido de que dos le han fallado. Sabe cuánto daño inflinge, a priori, cada uno de sus conjuros, y sabe cuánto daño real ha causado en total en el villano destino de sus hechizos. Ahora necesita saber qué dos han fallado por estar siendo afectados por su *conjunitis* para no usarlos otra vez.



Entrada

El programa deberá procesar, de la entrada estándar, múltiples casos de prueba.

Cada caso de prueba comienza con un número $3 \leq N \leq 200.000$ que indica la cantidad de hechizos invocados. A continuación sigue una línea con N números con el daño que, a priori, inflinge cada uno. La lista está ordenada de menor a mayor, y no hay dos hechizos iguales. El hechizo menos poderoso posible sólo tiene poder 1, y el más fuerte alcanza 500.000.

Por último, aparece una tercera línea indicando el daño total causado sobre el villano que ha recibido todos los hechizos.

La entrada termina con un caso sin hechizos, que no deberá procesarse.

Salida

Por cada caso de prueba, el programa escribirá, en la salida estándar, el poder a priori de los dos hechizos que no han funcionado, de menor a mayor. Se garantiza que habrá solución y que será única.

Entrada de ejemplo

```
3
1 2 3
3
4
1 3 5 9
10
5
1 3 5 6 7
18
0
```

Salida de ejemplo

```
1 2
3 5
1 3
```


● F

El hombre sin miedo y sin radar

Daredevil, también conocido como “*El hombre sin miedo*”, es un superhéroe que quedó ciego de niño por culpa de una sustancia radiactiva. Dicha sustancia le hizo perder la vista pero, al mismo tiempo, agudizó el resto de sus sentidos.



Así, *Daredevil* desarrolló un sistema de radar (similar a la *ecolocalización*) que le permite saber en qué lugar están los villanos con los que va a combatir.

Sin embargo, su archienemigo *Kingpin* le ha tendido una trampa y le ha encerrado en una habitación cuya resonancia anula su sistema de radar. Además, ha convocado a un montón de villanos que le esperan con rencor para darle una paliza.

Daredevil todavía no sabe la que se le viene encima pero, afortunadamente, tiene un *gadget* programable que quizá pueda ayudarle.

Entrada

El programa deberá procesar múltiples casos de prueba. Cada uno comienza con la descripción de la sala a la que llega *Daredevil*. Una sala es un espacio rectangular organizado en celdas, cada una de las cuales puede estar libre u ocupada por un villano.

La primera línea del caso de prueba contiene dos números, $1 \leq F, C \leq 100$ indicando la cantidad de filas y columnas que tiene la sala. A continuación vienen F líneas de C letras cada una. Las letras pueden ser “X” para indicar que la celda está ocupada por un villano, o “.” para indicar que está vacía.

Tras el mapa del caso de prueba aparece un número $1 \leq N \leq 100$ con la cantidad de consultas que se realizarán sobre ese mapa. Cada consulta es una posición donde podría estar *Daredevil* y la dirección hacia la que apunta con su *gadget*. La posición se indica proporcionando primero el número de fila, y luego el número de columna. La celda (1,1) se encuentra en la esquina superior izquierda del mapa. La dirección será IZQUIERDA, DERECHA, ARRIBA o ABAJO. Se garantiza que no se realizará ninguna consulta en una posición ocupada por un villano.

La entrada termina con dos ceros.

Salida

Por cada consulta se escribirá, en una línea independiente, la distancia, en número de celdas, al villano más cercano desde la posición de la consulta en la dirección indicada. Si no hay ningún villano, se escribirá “NINGUNO”.

Al final de cada caso de prueba se escribirá “---”.

Entrada de ejemplo

```
3 8
.X...X.
...X...
....X...
3
3 2 ARRIBA
2 1 DERECHA
1 1 ABAJO
0 0
```

Salida de ejemplo

```
2
3
NINGUNO
---
```