



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

ProgramaMe 2024

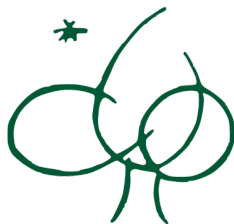
Regional Online Tarde - Torrent (Valencia)

Problemas



En homenaje a Niklaus Wirth, padre del lenguaje Pascal.

Organizado por:



**IES SERRA
PERENXISA**

Listado de problemas

A - Secuencia digna

B - Binario equilibrado

C - Trampeando el feedback

D - 3 en raya FOREVER

E - Código único

F - Identificando componentes

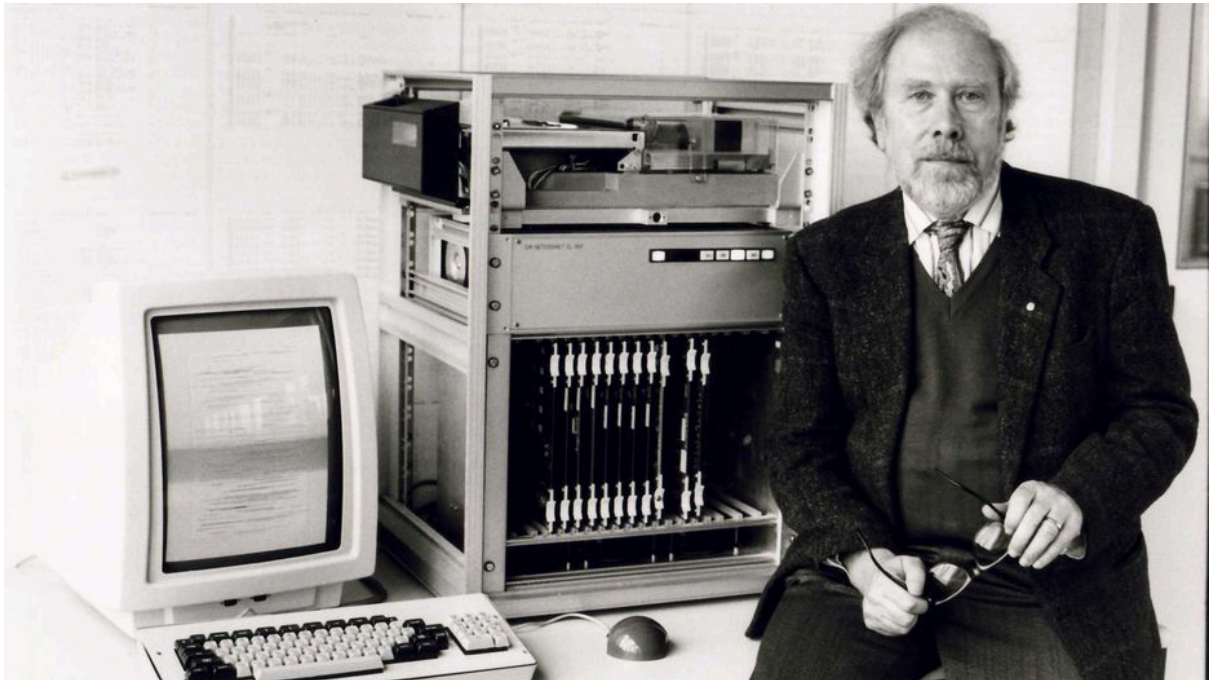
G - Proyectos de software

H - Recuperando tesoros

Problema A: Secuencia digna



ProgramaMe Regional Online Tarde 2023/24
IES Serra Perenxisa (Torrent, Valencia)



Niklaus Wirth fue un gran pionero en el campo de la ciencia de la computación y entre otras cosas director jefe del equipo que diseñó el lenguaje de programación Pascal y cuyo trabajo nos ha dejado un legado impresionante. Entre sus contribuciones, además de técnicas, estuvo la mentalidad de que aprender a programar es parte de la formación matemática básica.

Una de las actividades básicas para aprender programación y matemáticas, era saber trabajar con secuencias numéricas. Nuestro buen amigo Frikencio, quiere comprobar si sus amigos serían dignos estudiantes de Wirth y ha pensado una forma de comprobarlo: les va a proponer como debe ser una secuencia y deberán realizar un programa que compruebe si una secuencia dada cumple o no su propuesta.

La secuencia que ha pensado Frikencio se genera de la siguiente manera: comienza con un número positivo o negativo. La secuencia debe alternar números positivos y negativos. Esta alternancia solo se realiza si el siguiente número tiene un valor absoluto mayor que el anterior. Si no es así, se mantiene el tipo de valor (positivo o negativo).

Frikencio no has preguntado si somos dignos en el arte de programar y si nos atrevemos a comprobar si una secuencia de números cumple su propuesta o no.

Nuestra tarea consiste realizar un programa que, dada un conjunto de secuencia de Frikencio, nos diga si cada secuencia es correcta o no. ¿Demostraremos ser dignos programadores de los cuales estarían orgullosos Frikencio y Wirth?

Entrada

La entrada consta de una primera línea con un número entero T tal que $(1 \leq T \leq 100000)$, el cual representa el número de casos de prueba a procesar.

Cada caso de prueba estará compuesto de dos líneas, donde:

- La primera línea contendrá un número entero N tal que $(1 \leq N \leq 100000)$, el cual representa cuantos elementos tendrá la secuencia.
- La segunda línea contendrá la secuencia de números enteros, separados por espacios, siendo los números comprendidos con valores entre -100000 y 100000 , con excepción del cero, que no aparecerá en los casos de prueba.

Salida

El programa indicará una única línea por cada caso de prueba, indicando "CORRECTA" si la secuencia cumple las condiciones indicadas, o "INCORRECTA" en caso contrario.

Ejemplo de entrada

```
4
5
1 -2 3 -4 5
5
1 -2 3 -4 3
5
1 -2 2 4 -3
3
1 -2 -2
```

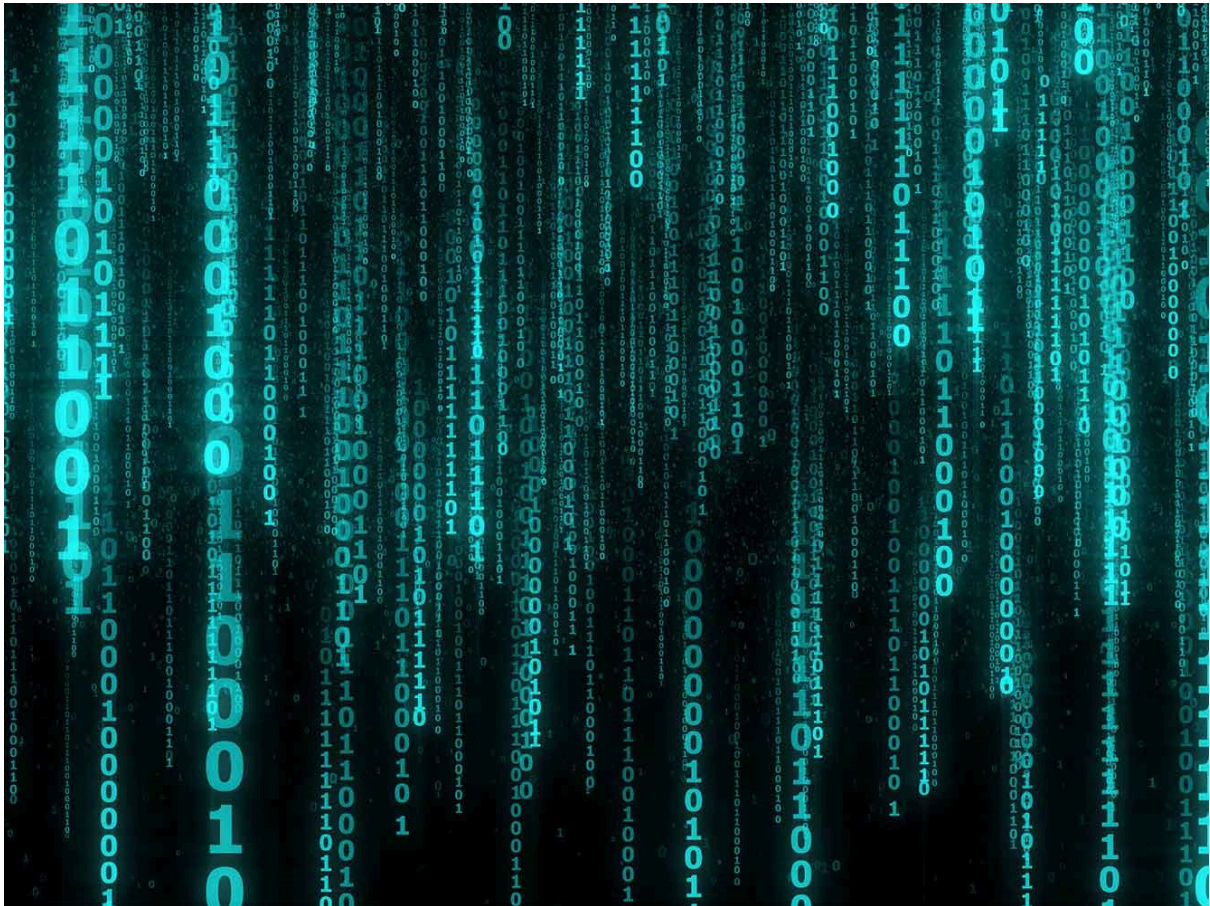
Ejemplo de salida

```
CORRECTA
INCORRECTA
INCORRECTA
CORRECTA
```


Problema B: Binario equilibrado



ProgramaMe Regional Online Tarde 2023/24
IES Serra Perenxisa (Torrent, Valencia)



Frikencio desde pequeño siempre ha alucinado al ver largas cadenas de números binarios, admirando su belleza y quedándose embobado.

Mientras disfruta de este espectáculo visual, que genera un equilibrio en su espíritu, pero pese a ello, su cabeza no puede parar de pensar. Una de las cosas que se pregunta cuando ve una cadena interminable de unos y ceros, es ¿Estará equilibrada?

Para Frikencio, se considera que una cadena binaria está equilibrada si tiene el mismo número de ceros que de unos y además los dos primeros dígitos son iguales a los dos últimos.

Por ejemplo, las cadenas "0101" y "00111100" son equilibradas y las cadenas "000111", "0001" y "10001" no lo son.

Frikencio nos ha propuesto realizar un programa que resuelva esta tarea. ¿Podremos con el reto?

Entrada

La entrada consta de una primera línea con un número entero T tal que $(1 \leq T \leq 100000)$, el cual representa el número de casos de prueba a procesar.

Cada caso de prueba estará compuesto de una línea formada exclusivamente por valores "0" o "1", siendo la línea de un tamaño entre 4 y 100000 caracteres

Salida

El programa indicará una única línea por cada caso de prueba, indicando "EQUILIBRADA" si la cadena está equilibrada, "ERROR" si no está equilibrada.

Ejemplo de entrada

5

0101

00111100

000111

0001

10001

Ejemplo de salida

EQUILIBRADA

EQUILIBRADA

ERROR

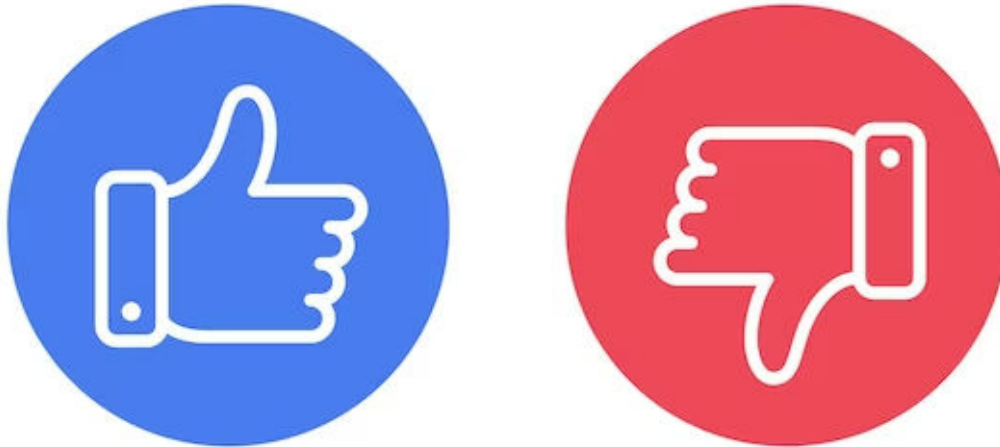
ERROR

ERROR

Problema C: Trampeando el feedback



ProgramaMe Regional Online Tarde 2023/24
IES Serra Perenxisa (Torrent, Valencia)



Frikencio ha utilizado sus conocimientos técnicos para crear una nueva aplicación de software libre y ha decidido lanzarla al mundo. Frikencio, además de lanzarla, ha creado un sitio web sencillo para recopilar comentarios de los usuarios, con dos botones para presionar: "Me gusta" y "No me gusta".

Sin embargo, el sitio no es tan simple internamente. Aunque el usuario lo ve como un único servidor, internamente hay dos servidores distintos, el servidor "A" y el servidor "B", cada uno con sus propios conteos para los "Me gusta" y los "No me gusta".

Los usuarios ingresan al sitio web, uno por uno. Cada usuario es uno de los siguientes tipos:

- **Tipo 1:** un usuario ha usado la aplicación y le gusta, por lo que presionan el botón de "Me gusta".
- **Tipo 2:** un usuario ha usado la aplicación y no le gusta, por lo que presionan el botón de "No me gusta".
- **Tipo 3:** un usuario que no ha usado la aplicación, por lo que mira el número actual de "me gusta" y "no me gusta" de la aplicación en el servidor en el que se encuentra y decide qué botón presionar. Si hay más "No me gusta" que "Me gusta", entonces el usuario presiona "No me gusta". De lo contrario, presionan "Me gusta".

Cada usuario vota en la aplicación exactamente una vez.

Dado que tienes dos servidores, puedes manipular los votos para que tu aplicación reciba tantos "me gusta" como sea posible. Cuando un usuario ingresa al sitio, conoces su tipo y puedes enviarlo ya sea al primer servidor o al segundo.

Frikencio tiene algo de inseguridad sobre la aplicación, y quiere maximizar el número total de "Me gusta" a su aplicación. Haz una aplicación que maximice el número de "Me gusta" totales, sumando los de ambos servidores.

Entrada

La entrada consta de una primera línea con un número entero T tal que $(1 \leq T \leq 100000)$, el cual representa el número de casos de prueba a procesar. Cada caso de prueba consta de:

- La primera línea de cada caso de prueba contiene cinco enteros N, A, B, X, Y donde:
 - N tal que $(1 \leq N \leq 500)$ indicando el número de usuarios.
 - A tal que $(1 \leq A \leq 5000)$ indicando el número de "Me gusta" inicial del servidor A.
 - B tal que $(1 \leq B \leq 5000)$ indicando el número de "No me gusta" inicial del servidor A.
 - X tal que $(1 \leq X \leq 5000)$ indicando el número de "Me gusta" inicial del servidor B.
 - Y tal que $(1 \leq Y \leq 5000)$ indicando el número de "No me gusta" inicial del servidor B.
- La segunda línea de cada caso de prueba contiene N enteros, separados por espacios, con los tipos de los usuarios en el mismo orden en que ingresan al sitio.

Salida

Para cada caso de prueba, imprime un solo entero: el número máximo total de "Me gusta" que puedes obtener (sumando los "Me gusta" de ambos servidores) enviando de forma óptima a cada tipo de usuario al servidor adecuado.

Ejemplo de entrada

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


Ejemplo de salida

19

18

5

Problema D: 3 en raya FOREVER



ProgramaMe Regional Online Tarde 2023/24 IES Serra Perenxisa (Torrent, Valencia)



Frikencio se preguntaba cuál fue el primer juego de 3 en raya de la historia. Investigando, se dio cuenta de que en 1952, Alexander S. Douglas creó el primer juego de computadora en usar una pantalla gráfica digital, llamado "OXO". Este juego es una versión del tres en raya para la computadora EDSAC en la Universidad de Cambridge. El juego poseía una rudimentaria pantalla digital y utilizaba un marcador giratorio de teléfono para el control del juego.

Frikencia, se ha propuesto crear una nueva versión de 3 en raya llamado "3 en raya FOREVER". Esta versión del 3 en raya tiene dos particularidades:

- Puede comenzar cualquiera de los dos jugadores (CRUZ o CIRCULO). El jugador que comienza debe obligatoriamente situar su jugada inicial en el centro del tablero.
- Aunque alguien resulte ganador, se continúa jugando hasta completar el tablero.

Periódicamente, Frikencio realiza garabatos al azar en un tablero de 3x3 y se pregunta si esos garabatos al azar podrían haber sido un estado válido de una partida de "3 en raya FOREVER".

Estos datos los representan en 3 líneas, donde un X significa una cruz, la letra O mayúscula significa un círculo y el carácter - significa una casilla en blanco.

Nos han pedido ayuda a nosotros como programadores. Nuestra tarea será realizar un programa que calcule si una partida de "3 en raya FOREVER" es correcta o no. En caso de ser correcta, nos indicará qué jugador (CIRCULO O CRUZ) comenzó la partida.

Entrada

En la primera línea, un entero C indicando el número de casos de pruebas deben procesarse.

Para cada caso de prueba, habrá 3 líneas con 3 caracteres cada una. Estos caracteres solo podrán ser X (letra X mayúscula), O (letra O mayúscula) y -. Cada tablero tendrá al menos una jugada. $1 \leq C \leq 100$

Salida

Para cada caso de prueba, escribir una línea indicando quién comenzó (CRUZ o CIRCULO)

o si no es válida la palabra IMPOSIBLE.

Ejemplo de entrada

5

--X

OXO

OOO

XXX

XOX

-OO

--X

-X-

--O

Ejemplo de salida

IMPOSIBLE

CIRCULO

IMPOSIBLE

CIRCULO

CRUZ

Problema E: Código único



ProgramaMe Regional Online Tarde 2023/24
IES Serra Perenxisa (Torrent, Valencia)



Frikencio ha querido organizar un concurso con sus amigos, llamado "El Desafío del Código Único". La idea es simple: los participantes deben elegir un número entero como su "código secreto". El ganador del concurso es el participante cuyo código es único (es decir, nadie más eligió este número, excepto él) y en el caso de que haya varios valores únicos, el menor es el ganador.

Frikencio te ha pedido ayudar para que realices un programa donde determines el nombre del ganador del concurso, o indiques "-1" si no hay ganador.

Entrada

La entrada consta de una primera línea indicando el número total de participantes P tal que $(1 \leq P \leq 10000)$.

Tras ello, tendremos P líneas, donde cada línea contendrá un entero, indicando el número seleccionado, y tras ello, un espacio y una cadena de texto (sin espacios) indicando el nombre del amigo que eligió ese número.

Salida

Se debe imprimir el nombre del ganador, o "-1" si no lo hay.

Ejemplo de Entrada

6

1 Alba

2 PEPE

11 JUAN

3 LUIS

1 Elena

12 MariaJo

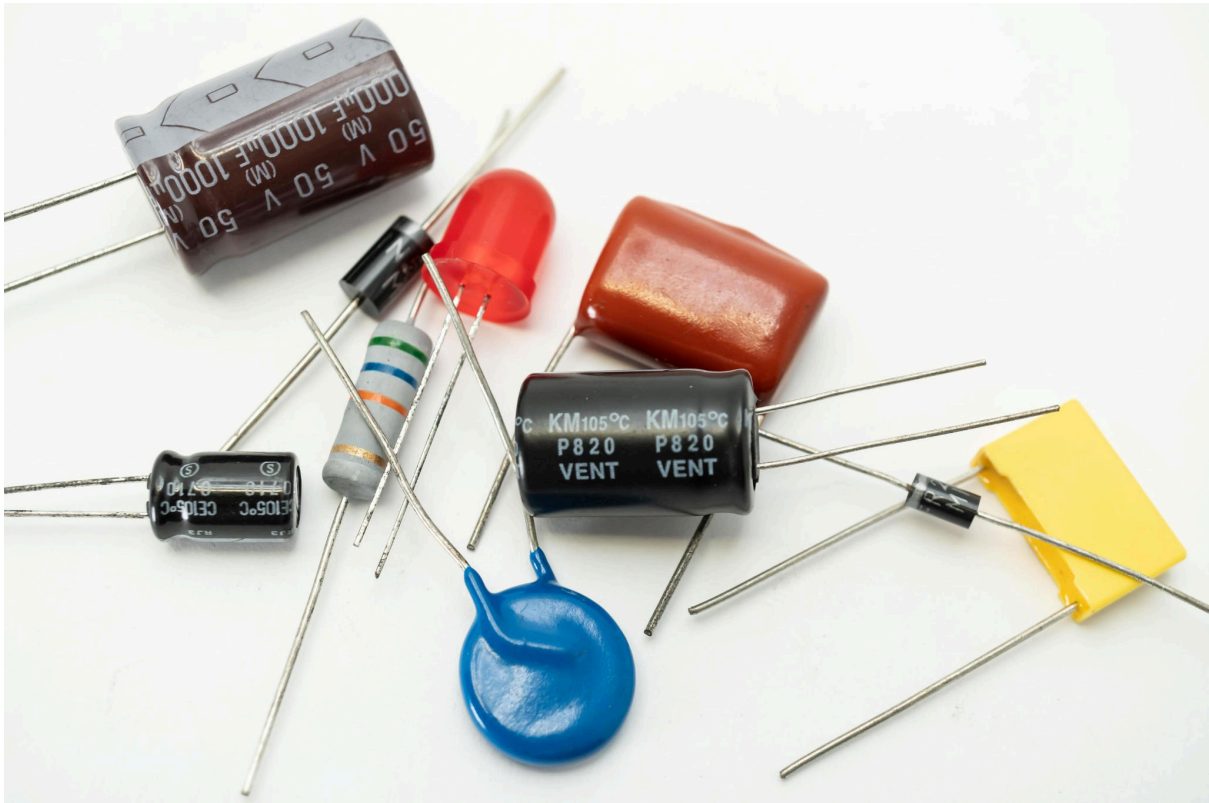
Ejemplo de Salida

PEPE

Problema F: Identificando componentes



ProgramaMe Regional Online Tarde 2023/24
IES Serra Perenxisa (Torrent, Valencia)



Frikencio se ha puesto a investigar como funcionan las herramientas para la creación de circuitos impresos (PCBs) y ha visto que son herramientas muy complejas y entre otras cosas, cuentan con algoritmos avanzados que permiten analizar y verificar los diseños antes de su producción.

Uno de los desafíos en el proceso de analizar y verificar diseños de circuitos impresos, es la identificación y conteo de componentes electrónicos, como resistencias y condensadores, representados en el diseño del circuito por símbolos específicos. En este problema, trabajaremos con una representación simplificada de un diseño de circuito en forma de matriz, donde los componentes se identifican con caracteres únicos: las resistencias con “R” y los condensadores con “C”. Los espacios vacíos en el diseño se representan con “.”. En esta matriz, es posible que varios símbolos del mismo tipo estén en contacto entre sí de forma horizontal o vertical. Si hay dos o más símbolos del mismo tipo en contacto, se consideran parte del mismo componente.

Entrada

El programa leerá en la primera línea dos números enteros con las dimensiones de la matriz que representa el diseño del circuito, formado por F filas y C columnas, tal que ($1 \leq F, C \leq 1000$).

Tras ello, se mostrará la descripción de la matriz. La descripción se realizará en las siguientes F líneas, donde cada línea tendrá C caracteres,

Salida

El programa deberá imprimir dos números separados por un espacio: el número de resistencias únicas y el número de condensadores únicos en el diseño del circuito.

Ejemplo de Entrada

5 5

.R.C.

...C.

R..C.

.R.C.

.R.R.

Ejemplo de Salida

4 1

Problema G: Implementación de proyectos de software



ProgramaMe Regional Online Tarde 2023/24
IES Serra Perenxisa (Torrent, Valencia)



Tras el lanzamiento exitoso de una nueva versión de un sistema operativo de código abierto, la comunidad de desarrollo está lista para celebrar este logro mediante la organización de una serie de eventos virtuales y proyectos colaborativos.

Dado un conjunto de proyectos de software y las habilidades de programación requeridas para cada uno, tu tarea es determinar si cada proyecto puede llevarse a cabo con las habilidades de los desarrolladores disponibles en la comunidad.

Cada proyecto se describe mediante una línea que contiene el nombre del proyecto y la lista de habilidades de programación que se requieren. Las habilidades posibles son: "FRONTEND", "BACKEND", "FULLSTACK", "UI/UX", "DATABASE", y "SECURITY". Algunos proyectos necesitan habilidades específicas, y solo los desarrolladores que posean esas habilidades podrán contribuir.

Entrada

La primera línea indica la cantidad total de desarrolladores en la comunidad, representado por un entero D , tal que $(1 \leq D \leq 10000)$.

A continuación, en las siguientes D líneas, se proporciona una lista de desarrolladores, cada uno representado primero por su nombre y tras ello por sus habilidades de programación, separadas por espacios.

Tras ello, se lee un segundo número indicando la cantidad total de proyectos de software planificados, representados por P tal que $(1 \leq P \leq 10000)$.

En las siguientes P líneas, para cada proyecto, se proporciona una línea que contiene el nombre del proyecto y la lista de habilidades de programación requeridas para llevarlo a cabo.

Salida

Para cada proyecto, el programa debe imprimir "POSIBLE" si el proyecto puede ser implementado con las habilidades de los desarrolladores disponibles, o "IMPOSIBLE" en caso contrario.

Ejemplo de Entrada

5

Alice FRONTEND BACKEND

Bob DATABASE SECURITY

Charlie FULLSTACK

Diana FULLSTACK DATABASE

Eva SECURITY

3

SistemaGestion FRONTEND BACKEND DATABASE

WebSegura SECURITY DATABASE

PlataformaEducativa FULLSTACK UI/UX

Ejemplo de Salida

POSIBLE

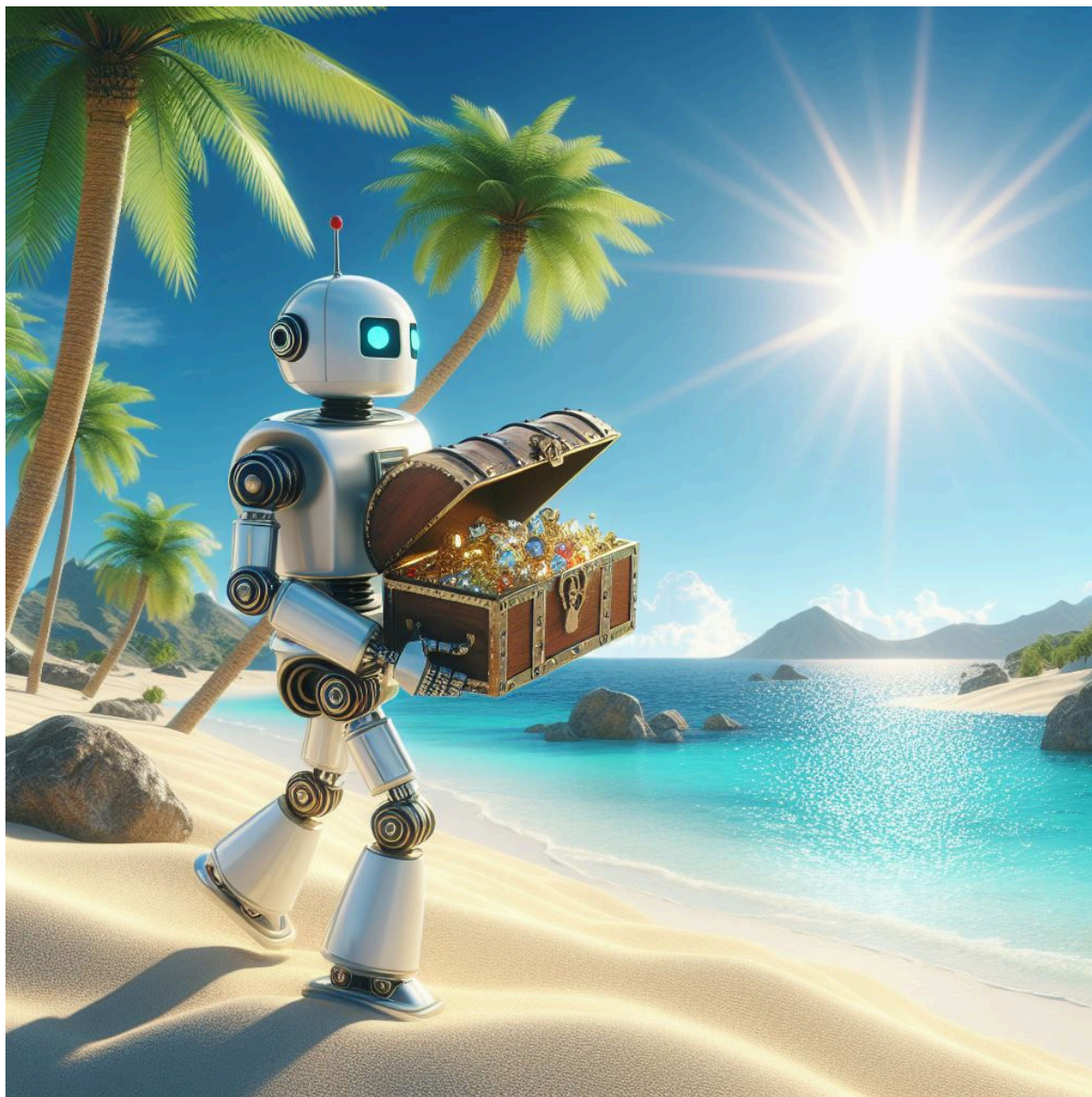
POSIBLE

IMPOSIBLE

Problema H: Recuperando tesoros



ProgramaMe Regional Online Tarde 2023/24
IES Serra Perenxisa (Torrent, Valencia)



Frikencio mediante el uso de Google Maps, ha localizado una isla desconocida. Dado el nivel de detalle de los satélites, ha podido revisar esta isla y cree que ha hallado lo que podría ser un tesoro (como toda buena isla desierta, esta también contiene un tesoro). Frikencio necesita tu ayuda para lograr esta gesta. ¿Podrás ayudarlo?

Para poder recuperar el tesoro, se plantea hacer una misión enviando a un robot “Emilio” especializado en búsqueda de tesoros. Este robot solo puede moverse en vertical y en horizontal.

Usando la imagen obtenida, hemos podido establecer una cuadrícula indicando la posición donde comienza nuestro robot, la posición del tesoro y si una casilla de la cuadrícula es atravesable por el robot o no.

Nuestra misión es desarrollar un programa que diga si el robot puede llegar hasta el tesoro o no.

Entrada

Cada caso de prueba constará de la siguiente información:

- En la primera línea tendremos dos enteros N (ancho) y M (largo) indicando el tamaño de la isla, tal que $2 \leq N, M \leq 20$
- En la segunda línea tendremos las coordenadas de inicio del robot RX y RY tal que $(1 \leq RX \leq N)$ y $(1 \leq RY \leq M)$.
- En la tercera línea tendremos las coordenadas del tesoro TX y TY tal que $(1 \leq Tx \leq N)$ y $(1 \leq TY \leq M)$.
- En la cuarta línea tendremos el número de celdas no atravesables O , tal que $(0 \leq O \leq 300)$.
- Las casillas del robot y del tesoro siempre serán distintas y ninguna estará marcada como no atravesable.

Las siguientes O líneas indicarán las coordenadas de cada zona no atravesable.

Salida

Si el tesoro es alcanzable, se mostrará la palabra TESORO. En caso contrario, se mostrará la palabra IMPOSIBLE.

Ejemplo de entrada

10 10

1 1

3 3

2

1 2

4 3

Ejemplo de salida

TESORO

Ejemplo de entrada

4 4

1 1

4 4

2

1 2

2 1

Ejemplo de salida

IMPOSIBLE