



**Olimpiada Chilena de Informática
2022**

Regional

15 de Octubre, 2022

Información General

Esta página muestra información general que se aplica a todos los problemas.

Envío de una solución

1. Los participantes deben enviar **un solo archivo** con el código fuente de su solución.
2. El nombre del archivo debe tener la extensión `.cpp`, `.java` o `.py` dependiendo de si la solución está escrita en **C++**, **Java** o **Python** respectivamente. Para enviar una solución en Java hay que seguir algunos pasos adicionales. Ver detalles más abajo.

Casos de prueba, subtareas y puntaje

1. La solución enviada por los participantes será ejecutada varias veces con distintos casos de prueba.
2. A menos que se indique lo contrario, cada problema define diferentes subtareas que lo restringen. Se asignará puntaje de acuerdo a la cantidad de subtareas que se logre solucionar de manera correcta.
3. A menos que se indique lo contrario, para obtener el puntaje en una subtarea se debe tener correctos todos los casos de prueba incluidos en ella.
4. Una solución puede resolver al mismo tiempo más de una subtarea.
5. La solución es ejecutada con cada caso de prueba de manera independiente y por tanto puede fallar en algunas subtareas sin influir en la ejecución de otras.

Entrada

1. Toda lectura debe ser hecha desde la **entrada estándar** usando, por ejemplo, las funciones `scanf` o `std::cin` en **C++**, la clase `BufferedReader` en **Java** o la función `input` en **Python**.
2. La entrada corresponde a un solo caso de prueba, el cual está descrito en varias líneas dependiendo del problema.
3. **Se garantiza que la entrada sigue el formato descrito** en el enunciado de cada problema.

Salida

1. Toda escritura debe ser hecha hacia la **salida estándar** usando, por ejemplo, las funciones `printf`, `std::cout` en C++, la función `System.out.println` en Java o la función `print` en Python.
2. El formato de salida es explicado en el enunciado de cada problema.
3. **La salida del programa debe cumplir estrictamente con el formato indicado**, considerando los espacios, las mayúsculas y minúsculas.
4. Toda línea, incluyendo la última, debe terminar con un salto de línea.

Envío de una solución en Java

1. Cada problema tiene un *nombre clave* que será especificado en el enunciado. Este nombre clave será también utilizado en el sistema de evaluación para identificar al problema.
2. Para enviar correctamente una solución en Java, el archivo debe contener una clase llamada igual que el nombre clave del problema. Esta clase debe contener también el método `main`. Por ejemplo, si el nombre clave es `marraqueta`, el archivo con la solución debe llamarse `marraqueta.java` y tener la siguiente estructura:

```
public class marraqueta {  
    public static void main (String[] args) {  
        // tu solución va aquí  
    }  
}
```

3. Si el archivo no contiene la clase con el nombre correcto, el sistema de evaluación reportará un error de compilación.
4. La clase no debe estar contenida dentro de un *package*. Hay que tener cuidado pues algunos entornos de desarrollo como Eclipse incluyen las clases en un *package* por defecto.
5. Si la clase está contenida dentro de un *package*, el sistema reportará un error de compilación.

Problema A

Daniela y los múltiplos

nombre clave: multiplos

Daniela es una gran fanática de los números. Tanto es su fanatismo, que frecuentemente dedica infinito tiempo a escribir listas con infinitos números en un papel infinito. La mayoría de los números le fascinan, pero con el paso del tiempo, Daniela también ha coleccionado una lista (finita) de números que no le gustan. Tanta es su aversión por estos números que siempre evita escribirlos en sus listas infinitas.

Recientemente, Daniela se ha obsesionado con los múltiplos. Para aprender más sobre ellos, Daniela acostumbra escoger un número de partida y escribir en orden todos los múltiplos de este. Como podrías esperar, al escribir los múltiplos, Daniela se salta los números que no le gustan. Por ejemplo, suponiendo que a Daniela no le gusta el 6 y el 12, al escribir los múltiplos del 3 se los saltará produciendo la lista 3, 9, 15, 18, 21, ...

Dado un número de partida a y un entero j , a Daniela le gustaría determinar qué número se encuentra en la posición j -ésima de la lista que escribe para a . ¿Podrías ayudarla?

Entrada

La primera línea contiene un número n ($1 \leq n \leq 10^5$), indicando la cantidad de números que no le gustan a Daniela.

La segunda línea contiene n enteros positivos separados por espacios, indicando los números que no le gustan a Daniela. Se garantiza que todos los números en esta lista serán distintos y estarán entre 1 y 10^8 .

La última línea contiene dos enteros positivos a y j ($1 \leq a, j \leq 10^4$), indicando que debes encontrar el j -ésimo número en la lista que Daniela escribe para a .

Salida

La salida debe contener un entero correspondiente al j -ésimo entero en la lista para a .

Subtareas y puntaje

Este problema no contiene subtareas. Se probarán varios casos de prueba y se entregará puntaje proporcional al número de casos de prueba correctos siendo 100 el puntaje máximo.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
3 6 9 10 3 2	12
5 1 2 3 4 5 1 1000	1005

Problema B

Contraseña

nombre clave: contraseña

Diego es uno de los líderes de la Unión Chilena de Hackers (UCH), lo cual lo pone en constante riesgo de ser atacado por otros hackers de organizaciones rivales. Es por esto que Diego siempre encripta sus contraseñas para así poder almacenarlas de forma segura y protegerse ante cualquier amenaza.

Diego debe almacenar un total de N contraseñas. Cada contraseña corresponde a una cadena de caracteres en el alfabeto inglés (sin la ñ). Las contraseñas distinguen entre mayúsculas y minúsculas. Para almacenarlas, Diego encriptó cada una de las contraseñas usando el mismo esquema de encriptación y las guardó en un archivo.

El esquema de encriptación se basa en intercambiar caracteres de la contraseña aplicando una lista de reglas. Cada regla consiste en dos caracteres c_1 y c_2 indicando los caracteres que se deben intercambiar. De forma concreta, para aplicar una regla se deben reemplazar todas las ocurrencias del carácter c_1 por c_2 y todas las ocurrencias de c_2 por c_1 . Para encriptar una contraseña, todas las reglas en la lista se aplican de forma consecutiva de arriba hacia abajo.

Por ejemplo, supón que queremos encriptar la contraseña **abanico** usando la siguiente lista de reglas:

a b
b c

Comenzamos aplicando la primera regla, es decir, reemplazando todas las ocurrencias de **a** por **b** y viceversa. Después de aplicar la regla obtenemos la cadena **babnico**. Posteriormente, aplicamos la segunda regla y obtenemos **cacnibo** que es la contraseña encriptada final.

Diego está muy ocupado con sus labores de líder de la UCH y necesita ayuda para escribir un programa que le permita desencriptar rápidamente las contraseñas en el archivo. De forma concreta, dada la lista de reglas y las N contraseñas encriptadas, el programa debe imprimir la lista de contraseñas desencriptadas. ¿Podrías ayudarlo?

Entrada

La primera línea de la entrada contiene dos enteros M y N ($0 < M \leq 10000$ y $0 < N \leq 1000$) correspondientes respectivamente a la cantidad de reglas y a la cantidad de contraseñas encriptadas.

A continuación vienen M líneas describiendo la lista de reglas. Cada línea contiene 2 caracteres separados por un espacio describiendo una regla.

Finalmente, siguen N líneas, cada una conteniendo una cadena de texto de largo mayor que cero y menor o igual a 100, correspondiente a una contraseña encriptada.

Se garantiza que todos los caracteres en la entrada (en las contraseñas encriptadas y la lista de reglas) serán caracteres del alfabeto inglés en mayúsculas o minúsculas.

Salida

La salida debe contener N líneas, correspondientes a las contraseñas originales descriptadas en el mismo orden en que aparecieron en la entrada.

Subtareas y puntaje

Subtarea 1 (15 puntos)

Se probarán varios casos en que $M = 1$, es decir, hay solo una regla.

Subtarea 2 (35 puntos)

Se probarán varios casos en que un mismo carácter no aparece en más de una regla.

Subtarea 3 (50 puntos)

Se probarán varios casos sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo

```
1 2
g n
tagnagagica
tagnagaga
```

Salida de ejemplo

```
tangananica
tanganana
```

Entrada de ejemplo

```
2 1
a b
b c
cacnibo
```

Salida de ejemplo

```
abanico
```

Entrada de ejemplo

```
3 1
A a
E e
O o
cOntrAsEnA
```

Salida de ejemplo

```
contrasena
```

Problema C

El lokómon de Olivier

nombre clave: lokomones

La Liga Lokómon es la mayor instancia a la que un entrenador Lokómon puede aspirar. Para participar, un entrenador debe clasificar teniendo un buen rendimiento en la pretemporada. Debido a la gran cantidad de talentosos entrenadores, clasificar a la liga es una tarea casi imposible.

Marinier es un obstinado entrenador Lokómon que lleva años tratando de clasificar a la liga. Su participación en la pretemporada fue casi impecable, pero luego de una mala racha en su última batalla, quedó abajo por solo unos puntos y no logró clasificar. Ya cansado de quedar fuera de la liga, este año Marinier ha decidido que es tiempo de dejar de jugar limpio.

Luego de una intensa investigación, Marinier se enteró de que su mayor rival, Olivier, cometió un error al inscribir uno de sus Lokómones. A pesar de ser un simple error y haber ganado sus batallas clasificatorias de forma justa, inscribir mal un Lokómon puede ser motivo de descalificación. Marinier no perderá la oportunidad de denunciarlo para así dejar un cupo abierto que el mismo podrá llenar. Lamentablemente, Marinier no sabe exactamente cuál de los Lokómones fue el que estuvo mal inscrito.

Para inscribir sus Lokómones, los participantes envían una *lista de inscripción* detallando para cada Lokómon los valores de cada uno de sus 6 *atributos* asociados. El valor de un atributo puede corresponder a una cadena de texto o a un entero. La siguiente tabla detalla cada uno de los 6 atributos, junto con un carácter identificador del atributo y si corresponden a una cadena de texto o a un entero.

atributo	identificador	tipo de atributo
nombre	n	texto
tipo	t	texto
ataque	a	entero
defensa	d	entero
evasión	e	entero
vida	v	entero

Marinier sabe que *solo una* de las entradas en la lista de inscripción de Olivier contiene errores. Adicionalmente, ha logrado recolectar un conjunto de atributos que sabe que el Lokómon mal inscrito *no tiene*. Por ejemplo, Marinier sabe que el nombre del Lokómon mal inscrito *no* es **Castillo** o que su ataque *no* es 100. Dada la lista de inscripción y el conjunto de atributos que el Lokómon mal inscrito no tiene, tu tarea es ayudar a Marinier escribiendo un programa que determine cuál entrada en la lista de inscripción contiene errores o imprimir que es imposible determinarlo de forma definitiva.

Entrada

La primera línea de la entrada consiste de dos enteros X y Y ($1 \leq X \leq 10^5$, $1 \leq Y \leq 100$). El entero X corresponde al largo de la lista que recopiló Marinier con los atributos que el Lokómon mal inscrito no tiene. El entero Y corresponde al largo de la lista de inscripción.

Las siguientes X líneas contienen la descripción de los atributos que el Lokómon mal inscrito no tiene. Cada línea comienza con un carácter identificador de atributo seguido del valor para el atributo, el que puede ser un entero o una cadena de texto dependiendo del atributo.

Las siguientes Y líneas describen las entradas en la lista de inscripción. Cada línea contiene los valores para el nombre, el tipo, el ataque, la defensa, la evasión y la vida (en ese orden) del Lokómon correspondiente. Se garantiza que todos los Lokómones en la lista tienen un nombre distinto.

Todos los atributos de tipo texto serán siempre no vacíos, tendrán un largo menor o igual que 20 y estarán compuestos por letras minúsculas del alfabeto inglés. Todos los atributos de tipo entero serán siempre mayores que cero y menores o iguales que 10^6 .

Salida

La salida debe contener el nombre del Lokómon en la entrada que contiene errores. En caso de no ser posible determinar de forma exacta cuál es la entrada que contiene errores, tu programa debe imprimir **IMPOSIBLE**.

Subtareas y puntaje

Este problema no contiene subtareas. Se probarán varios casos de prueba y se entregará puntaje proporcional al número de casos de prueba correctos siendo 100 el puntaje máximo.

Ejemplos de entrada y salida

Entrada de ejemplo

```
3 4
n ana
t planta
a 10
vicente fuego 10 5 15 20
emilia electrico 12 14 17 11
ana agua 11 4 30 7
david planta 9 3 19 2
```

Salida de ejemplo

```
emilia
```

Entrada de ejemplo

```
1 2
e 4
jorge fuego 2 2 2 2
andres agua 3 3 1 1
```

Salida de ejemplo

```
IMPOSIBLE
```

Problema D

Pájaros furiosos

nombre clave: pajaros

Los pájaros furiosos deben salvar sus huevos de la amenaza de los cerditos y están planeando una embestida para eliminarlos a todos de forma definitiva.

Los cerditos están distribuidos en distintas canastas, donde cada canasta puede contener 1 o 2 cerditos. Durante la embestida, cada pájaro puede elegir *una sola* canasta para atacar y eliminar a los cerditos en ella. Los pájaros pueden ser rojos o amarillos los cuales tienen diferente poder de ataque. Al atacar una canasta, un pájaro rojo puede eliminar a lo más un cerdito de esta. Si una canasta contiene dos cerditos y es atacada por un pájaro rojo, solo se eliminará un cerdito de esta y el otro deberá ser eliminado por el ataque de otro pájaro. Los pájaros amarillos pueden eliminar hasta dos cerditos. Un pájaro amarillo puede atacar una canasta con solo un cerdito (eliminándolo de la canasta) pero se perderá parte de su poder de ataque.

Tu tarea es ayudar a los pájaros furiosos escribiendo un programa que determine si es posible elegir el ataque de cada pájaro de forma que se eliminen todos los cerditos.

Entrada

La primera línea consiste en dos números enteros n y m ($1 \leq n, m \leq 5 \cdot 10^5$), indicando respectivamente la cantidad de pájaros y canastas.

La segunda línea contiene n enteros separados por espacios describiendo el color de cada pájaro. Un 1 indica un pájaro rojo mientras que un 2 indica uno amarillo.

La última línea describe las canastas y contiene m enteros separados por espacios. Los enteros pueden ser 1 o 2, indicando la cantidad de cerditos en cada canasta.

Salida

La salida debe contener SI en caso de ser posible que los pájaros eliminen a todos los cerditos, y NO en caso contrario.

Subtareas y puntaje

Este problema no contiene subtareas y se otorgará puntaje de acuerdo a la cantidad de casos correctos. Para obtener puntaje mayor que cero debes tener al menos un 50% de los casos de prueba correctos. Específicamente, si T es la cantidad total de casos y C la cantidad de casos correctos, el puntaje será 0 si $C < \frac{T}{2}$. Si $C \geq \frac{T}{2}$ el puntaje será $\frac{2 \cdot C - T}{T} \cdot 100$.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
5 4 1 2 1 1 2 2 1 2 2	SI
3 4 1 1 1 1 1 1 1	NO