



**Olimpiada Chilena de Informática
2022**

Final Nacional

26 de Noviembre, 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

Cuatro en línea

nombre clave: cuatro

El profesor de historia lleva una semana con licencia y la clase ha sido cancelada nuevamente. Aburrida y sin tener mucho que hacer, Romina le propone a sus compañeras y compañeros hacer un torneo de cuatro en línea.

El cuatro en línea es un juego de estrategia muy sencillo que se juega entre dos personas. Para jugarlo solo se requiere un lápiz y una hoja de cuaderno cuadrículada. El juego comienza definiendo una grilla de $N \times N$ sobre la hoja. Posteriormente, los jugadores toman turnos marcando celdas en la grilla. El primer jugador marca sus celdas usando una O (la letra “o” mayúscula) y el segundo jugador marca sus celdas usando una X (la letra “x” mayúscula). El objetivo de cada jugador es combinar *cuatro celdas en línea*, es decir, cuatro celdas consecutivas de forma vertical u horizontal (las líneas en diagonal no se cuentan). En cada turno, un jugador puede marcar cualquier celda vacía dentro de la grilla. El juego finaliza inmediatamente si al marcar una celda un jugador logra combinar cuatro celdas en línea. Alternativamente, un juego puede terminar en empate si al marcar una celda la grilla se llena completamente sin conseguir combinar cuatro celdas en línea. La siguiente figura muestra una posible secuencia de jugadas en una grilla de 4×4 que resultan en el primer jugador victorioso al combinar cuatro O’s de forma vertical.

															X		O		X		O		X		O		X
																		X				X			O		X
	O				O	X			O	X			O	X			O	X			O	X			O		X
									O				O				O				O				O		

Luego de terminar el torneo, Romina mira las grillas donde se jugó cada partida y sospecha que mucha gente no jugó en serio y simplemente marcó celdas sin seguir las reglas. ¿Podrías ayudar a Romina a determinar si sus sospechas son ciertas? Específicamente, dada una grilla donde algunas celdas han sido marcadas con una O o una X tu tarea es determinar si es posible que un juego de cuatro en línea termine con esta grilla.

Entrada

La primera línea de la entrada contiene un entero N ($4 \leq N \leq 10$) correspondiente al tamaño de la grilla. Posteriormente siguen N líneas cada una conteniendo N caracteres. El carácter j -ésimo en la línea i -ésima corresponde al contenido de la celda en la fila i y columna j . El carácter será un punto si la celda está vacía, o una O o una X si la celda fue marcada.

Salida

La salida deber contener **posible** en caso de ser posible que siguiendo las reglas descritas en el enunciado la grilla en la entrada sea el resultado de un juego de cuatro en línea. En caso de no ser

posible la salida debe contener `imposible`.

Subtareas y puntaje

Este problema no contiene subtareas. Se otorgará puntaje de acuerdo a la cantidad de casos de prueba correctos siendo 100 el puntaje máximo. Para obtener puntaje mayor que cero debes tener al menos un 50% de los casos 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
4 .0.X .OX. .OX. .0..	posible

Entrada de ejemplo	Salida de ejemplo
5 0.0.. .0.X. 0..X. ..OX. ...X.	imposible

Problema B

Reacciones

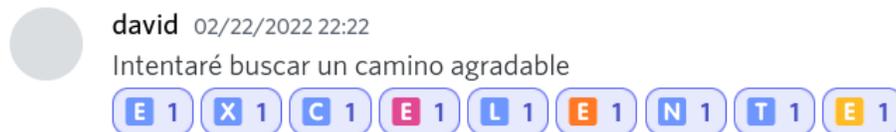
nombre clave: reacciones

Ocimatic es una conocida plataforma de mensajería instantánea por texto. Una de sus funcionalidades más populares es la posibilidad de *reaccionar* con emojis a los mensajes de otros participantes. Específicamente, un usuario puede reaccionar con uno o más emojis, pero solo una vez con cada emoji.

Las reacciones se muestran en una lista horizontal debajo de los mensajes. Asociado a cada reacción hay un contador. Si se reacciona con un emoji que ya había sido utilizado previamente, su contador aumenta en uno. En caso de reaccionar con un nuevo emoji, este se agrega al final de la lista con un contador igual a 1. Por ejemplo, en la imagen de la izquierda que se muestra a continuación hay 5 reacciones con el emoji 👍 y 2 reacciones con 🙌. Ya que el emoji 🔥 no ha sido utilizado aún, al reaccionar con este se añade a la lista como se muestra en la imagen de la derecha.



Francisca es una usuaria frecuente de Ocimatic. Una de sus costumbres es usar las reacciones para escribir mensajes. Esto es posible en algunos casos ya que Ocimatic tiene emojis por cada letra del abecedario en k colores diferentes. Si Francisca reacciona con estos emojis en el orden adecuado entonces puede escribir algunos mensajes. La siguiente imagen muestra como Francisca puede usar las reacciones para escribir **excelente**. Para esto, debe reaccionar con la letra **e** en cuatro colores distintos.



A Francisca le gustaría poder determinar si es posible o no formar un mensaje usando las reacciones antes de intentar escribirlo, de lo contrario, podría ocurrir que empiece a formar el mensaje y quede incompleto, afectando así su reputación en Ocimatic. ¿Podrías ayudarla?

Entrada

La primera línea de la entrada contiene dos enteros n y k ($1 \leq k \leq n \leq 1\,000\,000$) que corresponden respectivamente al largo del mensaje que Francisca quiere escribir con emojis y al número de colores que Ocimatic tiene por cada letra del abecedario. Representaremos cada color con un entero entre 1 y k .

La segunda línea contiene un mensaje de largo n formado por letras del abecedario inglés (es decir, sin la \tilde{n}), en minúsculas y sin espacios.

Salida

La salida consiste en una sola línea.

Si existe una manera de formar el mensaje, la línea debe contener n enteros c_i ($1 \leq c_i \leq k$) separados por espacios, donde el i -ésimo entero indica el color del que debe ser pintada la i -ésima letra del mensaje. Puedes imprimir cualquier secuencia de colores válida. Una secuencia es válida si ninguna letra aparece dos veces con el mismo color y cada color es un número válido entre 1 y k .

Si no es posible escribir el mensaje, la línea debe contener la palabra `imposible`.

Subtareas y puntaje

Subtarea 1 (20 puntos)

Se probarán varios casos en que $k = 1$.

Subtarea 2 (30 puntos)

Se probarán varios casos en que $k = 2$.

Subtarea 3 (50 puntos)

Se probarán varios casos sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo

```
9 4
excelente
```

Salida de ejemplo

```
1 1 1 2 1 3 1 1 4
```

Entrada de ejemplo

```
11 3
abracadabra
```

Salida de ejemplo

```
imposible
```

Problema C

Parkour

nombre clave: parkour

Tras muchos años de esfuerzo y dedicación, Sebastián finalmente logró clasificar a la fase final de la Increíble Competencia de Parkour con Cohetes (ICPC). En el último desafío, Sebastián debe arriesgar su vida en la ruta de las torres infernales.

La ruta está compuesta de N torres cada una con M pisos. Los pisos son numerados de abajo hacia arriba entre 0 y $M - 1$. Todos los participantes comienzan en el último piso de la primera torre (el piso $M - 1$) y para completar la ruta deben saltar de torre en torre hasta alcanzar cualquier piso de la última torre.

Para llegar de una torre a la siguiente, Sebastián puede saltar a un piso que esté a la misma o menor altura que el actual, o puede utilizar *cohetes* para propulsarse hacia un piso más alto, utilizando un cohete por cada piso que asciende. Por ejemplo, si Sebastián se encuentra en el piso 5 y desea llegar al piso 3 de la siguiente torre, puede hacerlo sin utilizar cohetes. Pero si quiere saltar hasta el piso 8 necesitará utilizar 3 cohetes.

La ruta de las torres infernales es un desafío muy extremo, por lo que muchos de los pisos estarán en llamas. Como Sebastián desea salir vivo de esta, solo saltará a los pisos *seguros*, es decir, los pisos que no están en llamas.

Además de ser un excelente deportista, Sebastián es una persona muy preocupada por el medio ambiente, por lo que desea reducir su huella de carbono utilizando la mínima cantidad de cohetes posibles. ¿Podrás ayudarlo a cumplir su meta?

Entrada

La primera línea de la entrada contiene dos enteros N y M ($2 \leq N, M \leq 100$) correspondientes respectivamente a la cantidad de torres y la cantidad de pisos en cada torre.

A continuación vienen N líneas. La i -ésima línea describe el estado de los pisos de la i -ésima torre. Cada línea comienza con un entero P ($1 \leq P \leq M$) correspondiente a la cantidad de pisos seguros en la torre. Luego siguen P enteros distintos, todos mayores o iguales que 0 y menores que M , correspondientes a los pisos seguros en la torre.

Se garantiza que siempre es posible completar la ruta y que el último piso de la primera torre es seguro.

Salida

La salida debe contener un único entero correspondiente a la cantidad mínima de cohetes con los que es posible completar la ruta.

Subtareas y puntaje

Subtarea 1 (20 puntos)

Se probarán varios casos en que $(2 \leq N, M \leq 5)$.

Subtarea 2 (20 puntos)

Se probarán varios casos en que $M = 2$.

Subtarea 3 (60 puntos)

Se probarán varios casos sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
5 4	2
1 3	
2 0 1	
2 0 3	
1 3	
1 1	

Entrada de ejemplo	Salida de ejemplo
8 2	1
2 0 1	
2 0 1	
1 0	
1 0	
2 0 1	
1 0	
1 1	
2 0 1	

Problema D

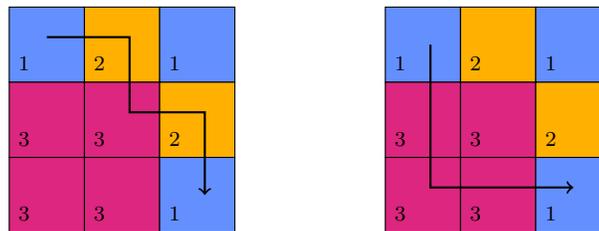
Viaje a la playa

nombre clave: playa

David está organizando un viaje a la playa con sus amigos. Su ciudad puede ser representada por una grilla de n filas y m columnas. Para referirnos a la celda en la fila i -ésima y columna j -ésima usaremos el par (i, j) . David comienza su viaje en la esquina superior izquierda $(1, 1)$ y debe moverse por la grilla siguiendo un *camino* hasta alcanzar la esquina inferior derecha (n, m) . En un movimiento, David puede pasar de la celda actual a una celda contigua que comparta un lado. El largo del camino se define como la cantidad de movimientos.

Adicionalmente, cada celda tiene un color asociado. El color de cada celda es representado por un número $C_{(i,j)}$ que va entre 1 y k . David le prometió a sus amigos buscar un camino *agradable*. Para que un camino sea agradable ningún par de celdas consecutivas puede tener el mismo color. Es decir, si en algún momento el camino pasa de la celda (i_1, j_1) a la celda (i_2, j_2) este solo puede ser agradable si $C_{(i_1, j_1)} \neq C_{(i_2, j_2)}$.

Por ejemplo, las siguientes figuras representan un posible mapa de la ciudad para una grilla con $n = 3$, $m = 3$ y $k = 3$. El número en cada celda corresponde a su color.



Cada figura muestra con una flecha un camino que va desde $(1, 1)$ hasta $(3, 3)$. El camino en la figura de la izquierda es agradable pues nunca atraviesa dos celdas consecutivas con el mismo color. El camino de la derecha no es agradable, ya que la segunda, tercera y cuarta celda son del mismo color. El largo del camino en ambos casos es 4.

David observa el mapa muy confundido sin poder encontrar un camino agradable y necesita de tu ayuda. Dada la descripción de la grilla, tu tarea es determinar si existe un camino agradable y, si existe, imprimir la distancia del camino agradable más corto.

Entrada

La primera línea contiene tres enteros n, m y k ($1 \leq n, m \leq 1000$, $2 \leq n \times m \leq 10^5$ y $2 \leq k \leq n \times m$) que corresponden a las dimensiones de la ciudad y la cantidad de colores que hay.

A continuación siguen n líneas, cada una con m enteros que van entre 1 y k . El j -ésimo entero de la i -ésima línea representa a $C_{(i,j)}$, el color de la celda (i, j) .

Salida

La salida debe contener el largo del camino agradable más corto entre las celdas $(1, 1)$ y (n, m) . Si este camino no existe debes imprimir -1.

Subtareas y puntaje

Subtarea 1 (20 puntos)

Se prouan varios casos en que $n = 1$, es decir, el mapa es solo una fila.

Subtarea 2 (20 puntos)

Se prouan varios casos en que $n = 2$ y $k = 2$, es decir, son dos filas con solo dos colores.

Subtarea 3 (20 puntos)

Se prouan varios casos en que $n = 2$.

Subtarea 4 (40 puntos)

Se prouan varios casos sin más restricciones.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
3 3 3 1 2 1 3 3 2 3 3 1	4

Entrada de ejemplo	Salida de ejemplo
2 2 2 1 1 1 2	-1