Olimpiada Chilena de Informática 2023

Final Nacional

11 de Enero, 2023

Las siguientes personas participaron en la elaboración de este conjunto de problemas:

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` o `.java` dependiendo de si la solución está escrita en `C++` o `Java` 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 incluídos 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++ o la clase `BufferedReader` en Java.
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++ o `System.out.println` en Java.
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
Balancín

nombre clave: `balancin`

Camila y Gabriel fueron al parque a jugar a su juego favorito: el balancín. En este emocionante juego de coordinación y equilibrio, dos personas se sientan en mitades opuestas de una tabla con un eje de rotación central. Ambos participantes ocupan su peso para ejercer fuerza en su lado de la tabla, de forma que mientras uno bajo el otro sube, ocasionando así un movimiento continuo hacia arriba y hacia abajo. ¡Es divertidísimo!

Para que el balancín se mueva adecuadamente, ambas personas deben ejercer la misma fuerza en cada lado. La fuerza que ejerce una persona depende de su peso y de su distancia al centro del balancín. Específicamente, la fuerza se calcula como $peso \times distancia$.

Para poder jugar, Camila y Gabriel deben buscar una forma de ejercer la misma fuerza. Ellos no pueden cambiar su peso, pero sí pueden elegir en qué parte del balancín se sientan. El balancín del parque tiene n asientos en cada lado, estos son numerados del 1 al n, y el i-ésimo asiento está a una distancia i del centro.

Por ejemplo, si Camila y Gabriel pesan $50kg$ y $30kg$ respectivamente y el balancín tiene 10 asientos por lado, Camila puede sentarse en la posición 6 y Gabriel en la 10. De esta forma, la fuerza que ejercerá cada uno será $50 * 6 = 300 = 30 * 10$, y como son iguales, podrán jugar.

Ambos están muy emocionados por comenzar, pero no están seguros de si existe una forma de sentarse que les permita jugar, por lo que te pidieron que lo averigues por ellos. ¿Podrás ayudarlos?

Entrada

La entrada consiste en una línea con tres enteros n, c, g ($1 \leq n, c, g \leq 10^9$) correspondientes respectivamente a la cantidad de asientos del balancín, el peso de Camila y el peso de Gabriel.

Salida

Debes imprimir `SI` si existe una forma de que Camila y Gabriel se sienten en el balancín tal que puedan jugar, y `NO` en caso contrario.

Subtareas y puntaje

Subtarea 1 (?? puntos)
Se probarán varios casos de prueba donde $1 \leq n, c, g \leq 10^3$.

Subtarea 2 (?? puntos)
Se probarán varios casos de prueba donde $1 \leq n, c, g \leq 10^6$.

Subtarea 3 (?? puntos)
Se probarán varios casos de prueba sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
`10 50 30`	`SI`

Entrada de ejemplo	Salida de ejemplo
`3 40 50`	`NO`

Problema B

Suma de ejemplo

nombre clave: `cachipun`

Finalmente ha llegado el día que todos estaban esperando: la fase final de la Olimpíada de Cachipún Interregional (OCI). En esta competencia, n oponentes se enfrentan en k rondas en la modalidad *todos contra todos*. En cada ronda, los participantes deben elegir entre jugar piedra, papel o tijera. Como es tradicional, la tijera le gana al papel, el papel a la piedra y la piedra a la tijera. Luego de que todos los participantes revelan su opción de forma simultánea, cada participante se enfrenta a cada otro participante y luego se calcula el puntaje de acuerdo a la cantidad de victorias y derrotas. Específicamente, cada participante gana 1 punto por cada victoria y pierde 1 punto por cada derrota. Los empates no afectan la puntuación.

Por ejemplo, considera una ronda para una competencia con 5 participantes donde cada uno juega de la siguiente forma:

Si nos concentramos en el participante 1, este le gana a los participantes 3 y 5 (la piedra le gana al papel), sumando 2 puntos. Adicionalmente, pierde contra el participante 2 (la piedra pierde contra el papel) lo cual resta 1 punto. Finalmente, empata contra el participante 4 lo que no afecta el puntaje. Por consiguiente, el participante 1 obtiene puntaje igual a 1 al final de la ronda.

El *puntaje final* de un participante luego de jugadas la k rondas es igual a la suma de los puntajes que obtuvo en cada ronda.

Históricamente, el registro de puntuación se ha manejado utilizando el Cachipún Management System (CMS), sin embargo, la organización ha encontrado una vulnerabilidad en el software, que parece ser muy difícil de arreglar. Es por esto que han decidido pedirte a ti que crees un nuevo programa que lo sustituya.

Quedan menos de 4 horas para que comience la competencia, ¿podrás salvar la OCI?

Entrada

La primera línea de la entrada contiene dos enteros n ($2 \leq n \leq 3000$) y k ($1 \leq k \leq 3000$), correspondientes a la cantidad de jugadores y de rondas respectivamente. Cada jugador es identificado con un entero entre 1 y n, mientras que las rondas con un entero entre 1 y k.

Posteriormente, cada una de las siguientes k líneas contiene n enteros describiendo una ronda. Específicamente, el j-ésimo entero de la i-ésima fila contiene la jugada del participante j en la ronda i. Un 0 representa que el participante jugó piedra, un 1 que jugó un papel y un 2 que jugó tijera.

Salida

La salida debe contener n enteros, donde el i-ésimo entero corresponde al puntaje final del jugador i.

Subtareas y puntaje

Subtarea 1 (20 puntos)

Se probarán varios casos de prueba donde $n = 2$, es decir, hay exactamente 2 jugadores.

Subtarea 2 (40 puntos)

Se probarán varios casos de prueba donde $n \leq 100$.

Subtarea 3 (40 puntos)

Se probarán varios casos de prueba sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
2 3 0 1 2 2 2 0	-2 2

Entrada de ejemplo	Salida de ejemplo
5 1 0 1 2 0 2	1 0 -1 1 -1

Entrada de ejemplo	Salida de ejemplo
3 3 1 2 2 0 1 2 1 1 0	-1 2 -1

Problema C

Dados para Sótanos y Lagartos

nombre clave: `dado`

Hernán es fanático de los juegos de rol. Su juego favorito es Sótanos y Lagartos. Cómo en muchos otros juegos de rol, para jugar Sótanos y Lagartos se necesita un conjunto especial de dados que además del tradicional dado cúbico de 6 caras contiene dados con otras formas y una mayor cantidad de caras. La siguiente imagen muestra un conjunto básico con dados de 6, 8, 12 y 20 caras.

Luego de años ganando experiencia, Hernán ha llegado al punto donde el conjunto básico ya no le es suficiente para las campañas más avanzadas y necesita dados con aún más caras. Tras días mirando en el mercado, aún no ha podido encontrado un diseño que lo satisfaga así que decidió hacer sus propios dados.

Hernán quiere que sus dados cumplan la norma internacional de Sótanos y Dragones. Esta norma indica que un dado de n caras debe cumplir las siguientes dos propiedades:

1. Cada cara debe tener un valor diferente entre 1 y n.
2. La suma de los valores de dos caras opuestas debe ser siempre $n + 1$.

Para un dado de n caras y un valor v en una de sus caras, llamamos complemento al valor que sumado a v da $n + 1$. Por ejemplo, en el tradicional dado de 6 caras, el complemento de 1 es 6 (y viceversa) el complemento de 2 es 5 (y viceversa) y el complemento de 3 es 4 (y viceversa).

Hernán ya ha escogido los valores de las caras para la mitad de un dado y ahora se pregunta si es posible completar la otra mitad cumpliendo con la norma internacional. Específicamente, dada la lista de valores para la mitad de las caras, un dado puede completarse si para cada valor en la lista, su complemento no se encuentra en la lista. ¿Podrías ayudarlo?

Entrada

La primera línea de la entrada contiene un entero n ($2 \leq n \leq 10^6$) correspondiente a la cantidad de caras en el dado. Se garantiza que n es un número par.

La siguiente línea contiene $n/2$ enteros distintos entre 1 y n indicando el valor en las caras de la primera mitad del dado.

Salida

En caso de ser posible completar el dado de acuerdo a las restricciones del enunciado, la salida debe decir `SI`. En caso contrario, la salida debe decir `NO`.

Subtareas y puntaje

Subtarea 1 (50 puntos)

Se probarán varios casos de prueba donde $n \leq 500$.

Subtarea 2 (50 puntos)

Se probarán varios casos de prueba sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
6 1 2 3	SI

Entrada de ejemplo	Salida de ejemplo
8 1 2 3 8	NO

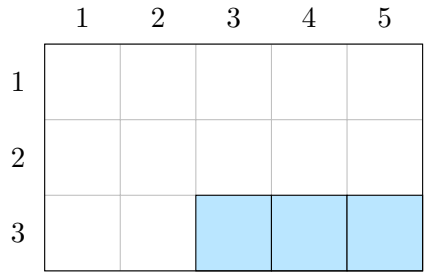
Problema D
Figuras

nombre clave: `figuras`

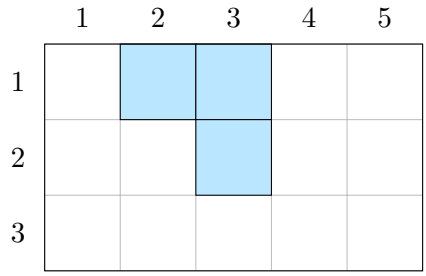
Cuenta la leyenda que el templo sagrado de Cumpeo alberga una cámara secreta llena de tesoros. En su última expedición, Sofía, una entrépita exploradora oriunda de Pelotillehue, encontró un pergamino que podría contener la clave para acceder a la cámara. Lamentablemente, sus fuentes le han confirmado que Lucía, una exploradora de Buenas Peras, también tuvo acceso al pergamino. Sofía no puede soportar la idea de perder contra alguien de Buenas Peras así que debe apresurarse para ser la primera en entrar a la cámara.

En la sala principal del templo, una de las paredes contiene un mural formado por una grilla de n filas y m columnas. Las filas son numeradas de arriba hacia abajo entre 1 y n, mientras que las columnas de izquierda a derecha entre 1 y m. La celda en la fila i y columna j es identificada con el par (i, j).

Algunas de las casillas en la grilla tienen azulejos que forman una *figura conexa*. Es decir, todos los azulejos son *adyacentes* a al menos un azulejo. Dos azulejos se consideran adyacentes si comparten un borde vertical u horizontal. La siguiente imagen muestra un ejemplo para una grilla de 3×5 con azulejos en las posiciones $(3, 3)$, $(3, 4)$ y $(3, 5)$.

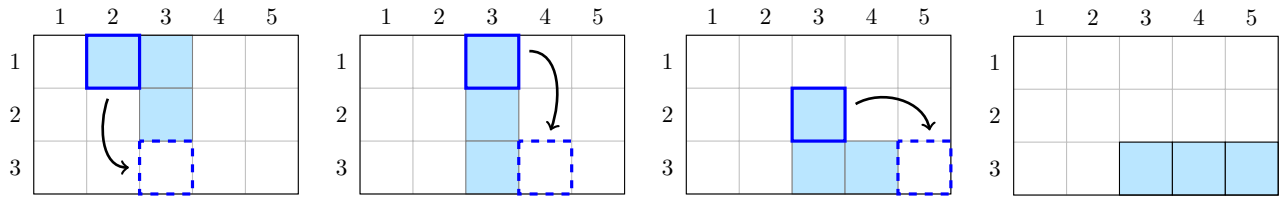


En el centro de la sala, hay un tablero formado también por una grilla de $n \times m$. Sobre el tablero, hay fichas que pueden moverse entre las casillas. La cantidad de fichas es la misma que la de azulejos, pero están dispuestas sobre el tablero en posiciones distintas. La siguiente imagen muestra un ejemplo del tablero con fichas en las posiciones $(1, 2)$, $(1, 3)$ y $(2, 3)$.



De acuerdo a las instrucciones en el pergamino, para abrir la puerta mágica a la cámara secreta hay que resolver el acertijo del tablero. El acertijo consiste en mover las fichas del tablero hasta que coincidan con los azulejos en el mural. Esto pareciera ser sencillo, pero para activar el mecanismo mágico, las fichas deben moverse siguiendo una regla especial. Específicamente, el único movimiento válido es tomar una ficha del tablero y colocarla adyacente a otra ficha de manera que en todo momento se forme una figura conexa.

Dado el mural y el tablero en las imágenes anteriores, la siguiente imagen muestra una posible secuencia de movimientos que pueden usarse para que las fichas en el tablero coincidan con los azulejos.



En el primer movimiento la ficha en la posición $(1, 2)$ se mueve a la posición $(3, 3)$. Posteriormente, la ficha en la posición $(1, 3)$ se mueve a la posición $(3, 4)$. Finalmente, la ficha en la posición $(2, 3)$ se mueve a la posición $(3, 5)$.

Sofía quiere resolver el acertijo lo más rápido posible, pero está teniendo problemas. En particular, le gustaría saber la mínima cantidad de movimientos en que es posible hacer coincidir las fichas con los azulejos. ¿Podrías ayudarla?

Entrada

La primera línea de la entrada contiene dos enteros n y m ($n \times m \leq 2 \times 10^5$) correspondientes a las dimensiones del tablero y el mural. Luego vienen $2 \times n$ líneas describiendo el tablero y el mural.

Las primeras n líneas describen el tablero. Cada línea contiene m enteros iguales a 0 o 1. El j-ésimo entero en la línea i describe la casilla (i, j) del tablero. Un 1 indica que la posición contiene una ficha y un 0 indica que la casilla está vacía.

Las últimas n líneas describen el mural usando el mismo formato para describir las casillas que tienen azulejos.

Se garantiza que la cantidad de unos en el tablero y en el mural será la misma.

Salida

La salida debe contener un entero, el número mínimo de movimientos requeridos para mover las casillas de forma que coincidan con los azulejos.

Subtareas y puntaje

Subtarea 1 (15 puntos)

Se probarán varios casos donde $n = 1$ y $m \leq 2 \times 10^3$.

Subtarea 2 (25 puntos)
Se probarán varios casos donde $n \times m \leq 2 \times 10^3$.

Subtarea 3 (60 puntos)
Se probarán varios casos sin restricciones adicionales.

Ejemplos de entrada y salida

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