

Olimpiada Chilena de Informática 2026

Regional

14 de diciembre, 2026

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 deben 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++ 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` o `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
Una noche de ronquidos

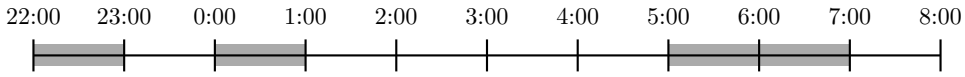
nombre clave: `horario`

Mañana Emilia tocará su flautanahoria en la Orquesta de Comida Instrumental (OCI). Está nerviosa por el concierto y quiere dormir bien, pero comparte habitación con su hermano Javier, que ronca tan fuerte que la despierta.



Una flautanahoria.

Por suerte, después de años de escucharlo, Emilia ya puede predecir en qué horas de la noche roncará Javier. La noche se divide en diez horas, desde las 22:00 hasta las 8:00. La siguiente figura muestra una noche en la que las horas en que Javier ronca están marcadas en gris.



Emilia puede acostarse al inicio de cualquiera de esas horas y debe levantarse a más tardar a las 8:00. No puede quedarse dormida mientras Javier ronca y, una vez dormida, se despierta en cuanto él empieza a roncar y ya no puede volver a dormirse. Emilia quiere elegir a qué hora acostarse para dormir la mayor cantidad de horas posible.

Emilia ya escribió un programa que calcula cuántas horas puede dormir como máximo. Ahora te desafía a escribir el tuyo.

Entrada

La entrada contiene una sola línea con una cadena de exactamente 10 caracteres, sin espacios. Cada carácter representa una hora de la noche. El primero corresponde al tramo entre las 22:00 y las 23:00; el segundo, al tramo entre las 23:00 y las 0:00; y así hasta el último, que corresponde al tramo entre las 7:00 y las 8:00.

Un `0` indica que Javier no roncará durante esa hora y un `1` indica que sí roncará.

Salida

Imprime la mayor cantidad de horas que Emilia puede dormir. Si Javier ronca durante toda la noche, imprime `0`.

Este problema tiene una sola subtarea de 100 puntos.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
1010000110	4

Este ejemplo corresponde a la noche de la figura. Emilia puede dormir cuatro horas, desde la 1:00 hasta las 5:00.

Entrada de ejemplo	Salida de ejemplo
1111111111	0

Problema B
Batalla lokomon

nombre clave: `lokomones`

Después de una larga travesía, Ashrias está a punto de salir de la calle Accepted para competir en la Liga Lokomon. A pocos metros de la salida aparece Mr. Nachianni, líder del equipo WAcket, y lo reta a una batalla.

Los lokomones de Ashrias están agotados por los combates del camino. Mr. Nachianni quiere una victoria de la que pueda presumir, así que le permite capturar un nuevo lokomon antes de la batalla. La batalla será de uno contra uno: el nuevo lokomon de Ashrias contra Vichochú, el lokomon favorito de Mr. Nachianni.

Por suerte, Ashrias tomó nota de los N lokomones que vio mientras recorría la calle Accepted. Además, conoce las características de Vichochú. Ahora necesita tu ayuda para decidir cuál capturar.

Cada lokomon tiene una cantidad de **vida**, un valor de **ataque** y un **tipo**. Hay K tipos, numerados del 1 al K. La efectividad de un ataque depende tanto del tipo de quien ataca como del tipo de quien lo recibe. Esta relación se describe mediante una matriz P: si un lokomon con ataque A y tipo i ataca a uno de tipo j, le resta $A \cdot P[i][j]$ puntos de vida. Un lokomon queda derrotado cuando su vida llega a 0 o menos.

Vichochú ataca primero. Desde ese momento, los lokomones se turnan para atacar. La batalla termina en cuanto uno de los dos cae derrotado, y gana el entrenador del lokomon que sigue en pie.

Ayuda a Ashrias a elegir uno de los N lokomones con el que pueda derrotar a Vichochú, o determina que ninguno le permite ganar.

Entrada

La primera línea contiene un entero K ($1 \leq K \leq 100$), la cantidad de tipos de lokomon.

Las siguientes K líneas contienen K enteros cada una. El j-ésimo entero de la i-ésima línea es $P[i][j]$ ($1 \leq P[i][j] \leq 10$), el multiplicador del ataque de un lokomon de tipo i contra uno de tipo j.

La siguiente línea contiene tres enteros H, A y T: la vida, el ataque y el tipo de Vichochú, respectivamente ($1 \leq H \leq 10^9$; $1 \leq A \leq 10^8$; $1 \leq T \leq K$).

La siguiente línea contiene un entero N ($1 \leq N \leq 100$), la cantidad de lokomones que Ashrias tiene como opción para capturar.

Luego vienen N líneas. La i-ésima contiene tres enteros H_i, A_i y T_i: la vida, el ataque y el tipo del i-ésimo lokomon que Ashrias puede capturar ($1 \leq H_i \leq 10^9$; $1 \leq A_i \leq 10^8$; $1 \leq T_i \leq K$). Estos lokomones están numerados del 1 al N según su orden en la entrada.

Salida

Si existe algún lokomon con el que Ashrias pueda ganar la batalla, imprime su índice (un entero entre 1 y N). Si hay varias opciones, puedes imprimir cualquiera de ellas. Si ninguno le permite ganar, imprime -1.

Subtareas y puntaje

- **Subtarea 1 (30 puntos)**

  La vida de Vichochú y la de cada uno de los N lokomones que Ashrias puede capturar son a lo más 1000 ($H \leq 1000$ y $H_i \leq 1000$ para todo i).

- **Subtarea 2 (70 puntos)**

  Sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
2 1 3 2 1 12 3 2 3 6 4 1 7 4 1 12 3 2	2

En este ejemplo, si Ashrias captura el segundo lokomon, Vichochú le quita $3 \cdot 2 = 6$ puntos de vida. Como este tenía 7 puntos de vida, alcanza a responder. Su ataque causa $4 \cdot 3 = 12$ puntos de daño, suficientes para derrotar a Vichochú, que tenía 12 puntos de vida.

Entrada de ejemplo	Salida de ejemplo
1 1 8 2 1 1 8 2 1	-1

Problema C
Poleras

nombre clave: `poleras`

Emilia guarda sus N poleras colgadas en una fila, numeradas de 1 a N. La polera en la posición i es de color C_i. Las poleras de un mismo color son indistinguibles.

Mientras Emilia tocaba su flautanahoria en el concierto de la OCI, su hermano Javier fue a un cumpleaños. Uno de los juegos era ver quién se ponía más poleras, así que sacó exactamente K poleras del clóset de Emilia y se las puso todas, una encima de otra. Al volver a casa las lavó y, para que Emilia no se diera cuenta, las colgó al final de la fila en cualquier orden. Las poleras que no sacó conservaron su orden relativo.

Cuando Emilia vuelve, mira la fila y ve que los colores de sus poleras, de izquierda a derecha, son $D_1, D_2, ..., D_N$. Javier jura que sacó exactamente K poleras, pero Emilia no le cree.

Dadas la configuración inicial $(C_1, C_2, ..., C_N)$ y la configuración final $(D_1, D_2, ..., D_N)$, determina si la historia de Javier es posible. Es decir, si se pueden elegir K poleras de la fila inicial de modo que, al sacarlas y colgarlas al final de la fila en algún orden, los colores queden exactamente como en la fila final.

Entrada

La entrada consiste en tres líneas. La primera línea contiene dos enteros N y K ($1 \leq K \leq N \leq 10^5$).

La segunda línea contiene N enteros $C_1, ..., C_N$ ($1 \leq C_i \leq N$): la configuración inicial.

La tercera línea contiene N enteros $D_1, ..., D_N$ ($1 \leq D_i \leq N$): la configuración objetivo.

Salida

La salida debe contener el string `SI` si se puede obtener la segunda configuración a partir de la primera, y `NO` en caso contrario.

Subtareas y puntaje

- **Subtarea 1 (15 puntos)**
  Javier saca todas las poleras ($K = N$).
- **Subtarea 2 (25 puntos)**
  Javier saca una sola polera ($K = 1$).
- **Subtarea 3 (60 puntos)**
  Sin restricciones adicionales.

Ejemplos de entrada y salida

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

Entrada de ejemplo	Salida de ejemplo
3 1 1 2 3 1 3 2	SI

Problema D
El arco de Lonk

nombre clave: `zolda`

En la última aventura de la saga *La leyenda de Zolda*, Lonk está a punto de enfrentarse a Mitundorf. Solo le queda cruzar un corredor lleno de meblins. Su espada se rompió en el último combate y ya no puede usar su ataque de giro. Para abrirse paso, tendrá que recurrir a su arco.

Lonk tiene E unidades de energía y quiere saber si le alcanzan para derrotar a todos los meblins del corredor o si tendrá que descansar antes de intentarlo.

El corredor se representa como una recta con posiciones enteras. Lonk comienza en la posición 0 y su objetivo es derrotar a los N meblins distribuidos a lo largo del corredor. Los meblins se numeran de 1 a N y el meblin número i se mantiene fijo en la posición x_i hasta ser derrotado. Dadas dos posiciones p y q sobre el corredor, la distancia entre ellas se define como $|p - q|$.

Para derrotar a los meblins, Lonk puede realizar dos acciones:

- **Saltar.** Lonk elige otra posición entera no negativa a su izquierda o a su derecha y salta a ella, gastando tanta energía como la distancia que recorre. Puede pasar por encima de los meblins, pero siempre debe aterrizar en una posición libre, es decir, sin un meblin vivo.
- **Disparar.** Lonk elige el meblin vivo más cercano a su izquierda o el más cercano a su derecha y le dispara una de sus flechas. Gasta tanta energía como la distancia al objetivo. El disparo derrota al meblin de inmediato y deja libre su posición, por lo que luego Lonk puede aterrizar en ella.

Lonk puede combinar saltos y disparos en cualquier orden y no necesita terminar en ninguna posición en particular. Ayúdalo a determinar si puede derrotar a todos los meblins sin gastar más de E unidades de energía.

Entrada

La primera línea contiene dos enteros N y E: la cantidad de meblins ($1 \leq N \leq 10^5$) y la energía disponible de Lonk ($1 \leq E \leq 10^9$).

La segunda línea contiene N enteros $x_1, x_2, ..., x_N$, donde x_i es la posición del meblin número i ($1 \leq x_i \leq 10^9$). Todas las posiciones son distintas.

Salida

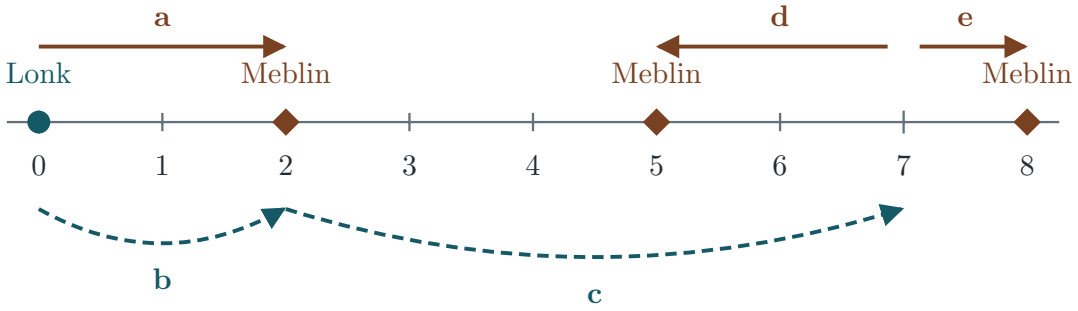
Imprime `YES` si Lonk puede derrotar a todos los meblins gastando como máximo E unidades de energía. En caso contrario, imprime `NO`.

Subtareas y puntaje

- **Subtarea 1 (5 puntos)**
  Hay un solo meblin ($N = 1$).
- **Subtarea 2 (25 puntos)**
  Los meblins ocupan N posiciones consecutivas.
- **Subtarea 3 (70 puntos)**
  Sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
3 13 2 5 8	YES



En el primer ejemplo, una estrategia posible es la siguiente:

a. Desde la posición 0, dispara al meblin en la posición 2 (gasta 2).
b. Salta a la posición 2, que ya está libre (gasta 2).
c. Salta por encima del meblin en 5 hasta la posición 7 (gasta 5).
d. Desde la posición 7, dispara al meblin en la posición 5 (gasta 2).
e. Desde la posición 7, dispara al meblin en la posición 8 (gasta 1).

En total, Lonk gasta $2 + 2 + 5 + 2 + 1 = 12$ unidades de energía. Como dispone de 13, la respuesta es `YES`.

Entrada de ejemplo	Salida de ejemplo
1 7 8	NO