



**Olimpiada Chilena de Informática  
2023**

Regional

*21 de Octubre, 2023*

## 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 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`, `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

## Rey De Los Piratas

*nombre clave:* piratas

Guffy es un joven que sueña con convertirse en el *Rey De Los Piratas*, un título que la ICPC (International Contest of Piracy and Computing) otorga al ganador de su competencia anual de piratería. La competencia consiste en que la ICPC publica  $N$  archivos online para que los participantes descarguen (de manera completamente legal). Quién descargue la mayor cantidad de archivos es coronado como el Rey De Los Piratas.

La ICPC publica sus archivos en forma de *torrent*. En una descarga por torrent, cada archivo tiene asociado un número de *seeds*, quienes son los que toman el rol de proveer el archivo a quienes lo descargan. Quienes descargan el archivo son llamados *peers*.

Para ganar la competencia, Guffy ha decidido priorizar los torrents con la mayor cantidad de seeds y la menor cantidad de peers; de esta forma puede maximizar la velocidad de descarga al tener más proveedores y menos adversarios tratando de descargar el mismo archivo.

Dada una lista de torrents publicados por la ICPC, Guffy necesita decidir cuál le conviene descargar primero. ¿Podrías ayudar a Guffy a elegir qué archivo descargar?

### Entrada

La primera línea de la entrada contiene un solo entero  $N$  ( $0 < N \leq 10^4$ ) que representa el número de torrents entre los cuales Guffy debe elegir. Cada torrent es identificado con un número entre 1 y  $N$ . Las siguientes  $N$  líneas describen los torrents. La línea  $i$ -ésima contiene dos enteros  $S_i$  y  $P_i$  ( $0 \leq S \leq 10^4$  y  $0 \leq P \leq 10^4$ ) que corresponden a la cantidad de seeds y peers del torrent  $i$ , respectivamente.

### Salida

La salida debe contener un único entero, correspondiente al número del torrent con la mayor cantidad de seeds. En caso de empate, se debe elegir el torrent con la menor cantidad de peers. Si hay un segundo empate, puede imprimirse el número de cualquiera de los torrents ganadores.

### Subtareas y puntaje

Este problema no contiene subtareas. Se dará puntaje proporcional a la cantidad de casos de prueba correctos, siendo 100 el puntaje máximo.

## Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
3	3
2 5	
4 0	
5 10	

**Explicación caso de ejemplo 1:** La respuesta es el torrent 3 que tiene 5 seeds y 10 peers, ya que es el torrent con la mayor cantidad de seeds y ningún otro torrent tiene la misma cantidad.

Entrada de ejemplo	Salida de ejemplo
5	1
10 5	
5 50	
9 0	
10 10	
0 0	

**Explicación caso de ejemplo 2:** En este caso los torrents 1 y 4 tienen la mayor cantidad de seeds con un total de 10. Como el torrent 1 tiene menos peers que el torrent 4, la respuesta es el torrent 1.

## Problema B

### Papas fritas

*nombre clave:* papas

Julieta y sus amigos fueron a comer papas fritas a su local favorito, la Industria Chilena de Papas Caseras (ICPC). Al grupo le gusta hacer todo en conjunto, así que compraron una porción de papas para compartir entre todos. Los amigos de Julieta son personas muy civilizadas y toman turnos para sacar papas (algunos incluso dirían que son demasiado civilizados). Para mantener el orden, han diseñado un sistema donde cada persona saca solo una papa a la vez.

En total son  $n$  personas en el grupo (incluyendo a Julieta) y todos están sentados alrededor de una mesa redonda. Una de las personas es elegida para sacar la primera papa. Luego saca una papa la persona de la derecha y así sucesivamente en una ronda alrededor de la mesa. Cuando ya todos sacaron una papa, el ciclo se vuelve a repetir hasta que ya no quedan más papas. Dado el orden que eligieron, Julieta es la  $k$ -ésima persona en sacar una papa por primera vez.

Como las papas fritas son caseras, cada papa puede tener un tamaño radicalmente distinto. Julieta es una persona muy observadora y ha determinado el tamaño  $t_i$  de cada una de ellas. Dependiendo del tamaño de las papas que saque, su ración será más o menos grande. El tamaño de su ración corresponde a la suma de los tamaños de todas las papas que saca.

Naturalmente, Julieta quiere comer lo más posible y le preocupa que su ración sea lo suficientemente grande. Específicamente, si saca papas maximizando su ración, e independiente de la elección de sus amigos, quiere saber el tamaño mínimo que tendrá su ración. ¿Podrías ayudarla?

#### Entrada

La primera línea de entrada contiene tres enteros  $p$ ,  $n$  y  $k$  ( $1 \leq k \leq n \leq p \leq 10^5$  y  $2 \leq n$ ), los cuales indican la cantidad de papas, la cantidad de personas en el grupo y la posición de Julieta, respectivamente.

La segunda línea contiene  $p$  enteros, donde el entero  $i$ -ésimo corresponde al tamaño  $t_i$  ( $1 \leq t_i \leq 10^3$ ) de la papa número  $i$ .

#### Salida

La salida debe contener un único entero, el tamaño mínimo que tendrá la ración de Julieta si saca las papas de forma óptima (maximizando su ración) e independiente de la elección de sus amigos.

#### Subtareas y puntaje

##### Subtarea 1 (15 puntos)

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

**Subtarea 2 (15 puntos)**

Se probarán varios casos en que  $n = k = p$ .

**Subtarea 3 (30 puntos)**

Se probarán varios casos en que  $p \leq 250$ .

**Subtarea 4 (40 puntos)**

Se probarán varios casos sin restricciones adicionales.

**Ejemplos de entrada y salida**

**Entrada de ejemplo**

6 2 1  
1 2 3 4 5 6

**Salida de ejemplo**

12

**Entrada de ejemplo**

5 5 5  
80 20 4 23 100

**Salida de ejemplo**

4

# Problema C

## Paltas

*nombre clave:* paltas

Francisca es fanática de las paltas. Las come al desayuno, al almuerzo y a la cena. Con pan, en una ensalada o con leche. A veces simplemente pela una palta y se la come a mascadas. Cada vez que va a la feria compra una malla entera para así tener suficientes paltas.

Comprar paltas es un proceso complejo. Dependiendo de cuándo quieres comerlas, debes elegir entre comprar paltas ya maduras —listas para consumir— o paltas que aún no han madurado para poder consumirlas en un par de días más. Más aún, como cualquier otra fruta, algunas paltas maduran más rápido que otras, haciendo todo más difícil. Francisca tiene que pensar en todo esto cada vez que va a la feria, por lo que necesita un poco de ayuda para poder comprar las paltas de forma más eficiente.

Francisca quiere comprar una malla de  $n$  paltas. Enumeraremos cada una de las paltas en la malla con un entero entre 1 y  $n$ . Además, representaremos el *estado de maduración* de una palta con un entero mayor o igual a cero, donde 0 corresponde al estado al momento de comprar la malla y cualquier entero mayor que 0 indica un grado de madurez mayor. Es decir, en un inicio todas las paltas en la malla tienen un estado de maduración igual a 0 y el valor irá aumentando con el tiempo.

Cada palta madura a una velocidad distinta. Denotaremos con  $r_i$  la *tasa de maduración* de la palta  $i$ -ésima. Es decir, la palta  $i$ -ésima madura  $r_i$  unidades por cada unidad de tiempo. En otras palabras, en el tiempo  $t$ , la palta  $i$ -ésima tiene un estado de maduración igual a  $t \cdot r_i$ .

Una palta solo puede consumirse si su estado de maduración se encuentra en el rango adecuado. Específicamente, una palta puede consumirse si su estado de maduración es mayor o igual que  $k_{min}$  y menor o igual que  $k_{max}$ . Si el estado de maduración es menor que  $k_{min}$ , la palta aún está muy verde. Si el estado es mayor que  $k_{max}$ , la palta ya está podrida.

Dadas las tasas de maduración de las  $n$  paltas, tu trabajo es responder a  $q$  consultas del siguiente tipo: ¿Cuántas paltas pueden consumirse en el tiempo  $t$ ?

### Entrada

La primera línea contiene 4 enteros  $n$ ,  $q$ ,  $k_{min}$  y  $k_{max}$  que indican la cantidad de paltas, el número de consultas y los límites del rango de madurez aceptado. Los valores satisfacen las siguientes restricciones:

- $1 \leq n \leq 10^6$
- $1 \leq q \leq 10^5$
- $0 \leq k_{min} \leq k_{max} \leq 10^9$

La segunda línea contiene  $n$  enteros. El entero  $i$ -ésimo corresponde a la tasa de maduración  $r_i$  ( $1 \leq r_i \leq 10^4$ ) de la palta  $i$ -ésima.

Luego vienen  $q$  líneas que describen la consultas. Cada línea contiene un entero  $t$  ( $0 \leq t \leq 10^5$ ). La respuesta a la consulta debe corresponder a cuántas paltas pueden consumirse en el tiempo  $t$ .

## Salida

La salida debe contener  $q$  líneas con la respuesta a cada una de las consultas en el orden en que fueron entregadas.

## Subtareas y puntaje

### Subtarea 1 (35 puntos)

Se probarán varios casos en donde  $n, q \leq 10^3$ .

### Subtarea 2 (65 puntos)

Se probarán varios casos sin restricciones adicionales.

## Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
4 3 6 10	2
1 2 3 4	1
3	0
7	
12	

**Explicación caso de ejemplo 1:** En la primera consulta se pregunta cuántas paltas pueden consumirse en el tiempo 3. En este tiempo, las paltas tendrán estados de maduración 3, 6, 9 y 12. Como el rango de madurez aceptado es  $[6, 10]$ , solo dos paltas pueden consumirse (las paltas 2 y 3). La siguiente consulta pregunta por el tiempo 7. En este tiempo, las paltas tendrán estados de maduración 7, 14, 21 y 28. Por tanto, solo una palta puede consumirse (la palta 1). En la última consulta se pregunta por el tiempo 12. En este tiempo las paltas tendrán estados de maduración 12, 24, 36 y 48. Por tanto, ninguna palta puede consumirse.

**Entrada de ejemplo**

```
5 1 5 5
1 1 1 1 1
5
```

**Salida de ejemplo**

```
5
```

**Explicación caso de ejemplo 2:** La única consulta pregunta por el tiempo 5. En este tiempo los estados de maduración serán 5, 5, 5, 5 y 5. Como el rango permitido es  $[5, 5]$  todas las paltas pueden consumirse.

## Problema D

### Poleras

*nombre clave:* poleras

Javier va a viajar a la Olimpiada Internacional de Informática (IOI) acompañando al equipo Chileno. En su emoción, decidió mandar a hacer poleras oficiales del equipo.

Como todos sabemos, las poleras vienen en  $n$  tallas, siendo la talla 1 la más pequeña y la talla  $n$  la más grande. Para cada talla  $i$ , Javier mandó a hacer  $a_i$  poleras de esa talla.

Un par de días antes de iniciar el viaje, Javier se dio cuenta de un gran error que había cometido. ¡Había registrado mal las tallas de poleras de los miembros de la delegación! En vez de  $a_i$  poleras de talla  $i$  necesitaba  $b_i$  poleras de esa talla.

Como ya era muy tarde para mandar a hacer las poleras de nuevo, Javier empezó desesperadamente a buscar una solución alternativa. En un momento de inspiración, recordó que se puede encoger una polera metiéndola a la secadora de ropa a alta potencia.

Después de investigar en internet, Javier descubrió que, controlando el tiempo de secado, su secadora puede encoger una polera hasta  $k$  tallas. Es decir, una polera de talla  $s$  puede encogerse a las tallas  $s - 1, s - 2, \dots, s - k$ . Naturalmente, una polera de talla menor o igual que  $k$  solo puede encogerse hasta la talla 1. Cada polera puede meterse solamente una vez a la secadora para encogerla.

Antes de iniciar el proceso, Javier quiere asegurarse de que es posible convertir las poleras que ya compró en poleras de las tallas que necesita. Específicamente, quiere saber si para cada talla  $i$  es posible terminar con al menos  $b_i$  poleras de esa talla si encoge algunas de las poleras que ya compró. Por ejemplo, considera un escenario representado por la siguiente tabla, donde la cantidad de tallas es tres ( $n = 3$ ):

	1	2	3
$a$	10	21	32
$b$	12	20	30

La fila  $a$  representa las poleras que compró de cada talla y la fila  $b$ , la cantidad de poleras que necesita. Si la secadora puede encoger poleras hasta dos tallas ( $k = 2$ ), entonces Javier tiene varias opciones para obtener las poleras que necesita. Algunas de las opciones son:

- Encoger la polera extra de talla 2 en una talla y una de las poleras extra de talla 3 en dos tallas.
- Encoger las dos poleras extra de talla 3 en dos tallas.
- Encoger dos poleras de talla 2 en una talla y luego una polera de talla 3 en una talla.

Como Javier está muy apurado con los preparativos del viaje, no tiene el tiempo para determinar con certeza si puede encoger las poleras adecuadamente. ¿Podrías ayudarlo?

## Entrada

La primera línea de la entrada contiene los enteros  $n$  y  $k$  ( $1 \leq n \leq 10^6, 0 \leq k \leq 10^6$ ), que indican la cantidad total de tallas y la cantidad máxima de tallas que la secadora puede encoger, respectivamente.

La segunda línea contiene los números enteros  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^9$ ), que corresponden a la cantidad de poleras de cada talla que compró Javier.

La tercera línea contiene los números enteros  $b_1, b_2, \dots, b_n$  ( $0 \leq b_i \leq 10^9$ ), que indican la cantidad de poleras de cada talla que necesita Javier.

## Salida

Si para cada  $i$  es posible terminar con al menos  $b_i$  poleras de esa talla, tu programa debe imprimir SI. En caso contrario, debe imprimir NO.

## Subtareas y puntaje

### Subtarea 1 (10 puntos)

Se probará varios casos con  $k = 0$ , es decir, la secadora no puede encoger poleras.

### Subtarea 2 (20 puntos)

Se probará varios casos con  $k = 1$ , es decir, la secadora puede encoger poleras a lo más una talla.

### Subtarea 3 (30 puntos)

Se probará varios casos con  $k \leq 100$ .

### Subtarea 4 (40 puntos)

Se probará varios casos sin restricciones adicionales.

## Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
3 0 10 21 32 12 20 30	NO

**Explicación caso de ejemplo 1:** Javier compró 10 poleras de talla 1, 21 de talla 2 y 32 de talla 3, pero necesita 12 de talla 1, 20 de talla 2 y 30 de talla 3. Javier tiene suficientes poleras de talla 2 y 3, pero le faltan dos de talla 1. Como no es posible encoger poleras ( $k = 0$ ), es imposible obtener las poleras que le falta y por lo tanto la respuesta es NO.

**Entrada de ejemplo**

```
3 2
10 21 32
12 20 30
```

**Salida de ejemplo**

```
SI
```

**Explicación caso de ejemplo 2:** Este es el ejemplo explicado en el enunciado.

**Entrada de ejemplo**

```
3 2
10 21 30
12 20 30
```

**Salida de ejemplo**

```
NO
```

**Explicación caso de ejemplo 3:** En este caso la respuesta es **NO**, porque si Javier encoge algunas poleras para obtener las poleras de talla 1 que le faltan, se quedará sin poleras suficientes de talla 2 o 3.