



Olimpiada Chilena de Informática 2019

5 de Octubre, 2019

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

Benjamín Aguilar, Rodrigo Alonso,
Sebastián Barbieri, Robinson Castro,
Nicolás Lehmann, Javier Oliva,
Bernardo Subercaseaux

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

Azulejos

nombre clave: azulejos

Vardieri acaba de adquirir uno de los más respetados restaurantes en Portugal, famoso por sus hermosos diseños de azulejos. Paradójicamente, en su primera inspección notó que la calidad de la comida era notoriamente inferior a la de otros restaurantes de la zona, por lo que decidió indagar en el asunto. En la inspección, notó que el embaldosado de la cocina está formado por dos tipos de azulejos, 0 y 1, los cuales se muestran a continuación.

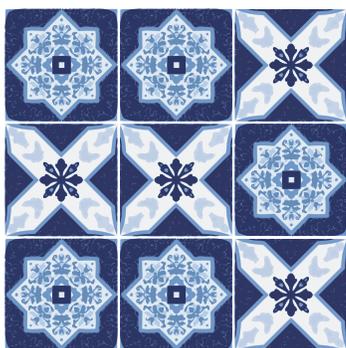


(a) azulejo de tipo 0

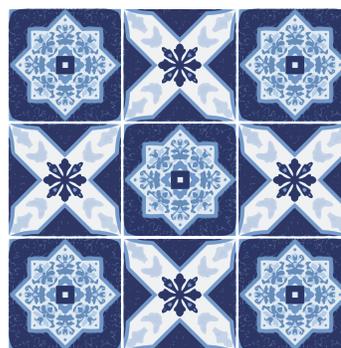


(b) azulejo de tipo 1

El diseño de estos azulejos está pensado para producir un patrón de azulejos alternados, es decir, cada azulejo de tipo 0 tiene un azulejo de tipo 1 a cada uno de sus lados y viceversa. Sin embargo, actualmente los patrones en la cocina no respetan esa regla, irritando así al chef y afectando la calidad de su comida. A continuación puedes ver a la izquierda un patrón equivocado, como los que están en la cocina, y a la derecha uno correcto con un patrón alternado.



(a) patrón incorrecto



(b) patrón correcto con azulejos alternados

Para remediar esta situación, Vardieri ha decidido reparar los patrones, calmando así el espíritu riguroso del chef y restaurando la calidad de sus platos. Sin embargo, el presupuesto que Vardieri tiene disponible es limitado y necesita saber cuál es la forma más económica de realizar esta tarea. El costo de cambiar un azulejo de tipo 0 por uno de tipo 1 es P , y el costo de cambiar uno de tipo 1 por uno de tipo 0 es Q .

¿Puedes ayudar a Vardieri a encontrar la solución más económica para restaurar correctamente los azulejos?

Entrada

La entrada tiene varias líneas.

La primera línea contiene 4 enteros, M , N , P y Q ($1 \leq M, N \leq 10^3$, $1 \leq P, Q \leq 10^2$) que corresponden a las dimensiones $M \times N$ de un segmento rectangular de azulejos, y a los costos de reemplazar los azulejos.

Las siguientes M líneas contienen N números, cuyos valores pueden ser 0 ó 1, indicando un tipo de azulejo. El j -ésimo número de la i -ésima línea indica qué tipo de azulejo está en la posición (i, j) del embaldosado.

Salida

La salida debe contener una línea con un único entero, correspondiente al costo mínimo necesario para cambiar el embaldosado a uno con patrón alternado.

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 1 1 0 1 0 1 0 1 0 1 0	0

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

Entrada de ejemplo

3 4 1 1
0 0 1 0
0 1 0 1
1 0 1 0

Salida de ejemplo

1

Problema B

Reuniones

nombre clave: reuniones

Sabina ha estado trabajando en una start-up durante los últimos años. Recientemente, Sabina ha visto su start-up crecer al punto de necesitar una nueva oficina. Uno de los aspectos más importantes para ella es asegurarse de que la nueva oficina tenga suficientes salas para todas las reuniones de la empresa. Como Sabina está muy ocupada siendo la dueña de la empresa, te ha pedido ayuda para resolver este problema.

Cada reunión necesita tener una sala asignada, pero una sala no puede estar asignada a más de una reunión al mismo tiempo. Por lo tanto, una sala puede ser reutilizada si y solo si los tiempos de las reuniones asignadas a la sala no se intersectan. Puedes asumir que el tiempo en desocupar una oficina es despreciable y por lo tanto una reunión puede comenzar exactamente en el momento en que otra termina.

Con el fin de ahorrar dinero, Sabina está buscando una oficina con la mínima cantidad de salas que le permita acomodar las reuniones de su empresa.

Entrada

La primera línea de la entrada está compuesta de un solo entero N . Luego siguen N líneas que describen el tiempo de las reuniones. La i -ésima línea contiene dos enteros, p_i y q_i ($p_i < q_i$), separados por un espacio, representando respectivamente el tiempo en que comienza y termina la i -ésima reunión.

Salida

La salida debe contener un único entero, correspondiente a la cantidad mínima de salas que debe tener la nueva oficina para poder alojar todas las reuniones.

Subtareas y puntaje

Subtarea 1 (10 puntos)

Se probarán varios casos donde se cumplen las siguientes restricciones:

- $0 < N \leq 2$
- $0 \leq p_i < q_i \leq 10^3$

Subtarea 2 (10 puntos)

Se probarán varios casos donde se cumplen las siguientes restricciones:

- $0 < N \leq 3$
- $0 \leq p_i < q_i \leq 10^3$

Subtarea 3 (30 puntos)

Se probarán varios casos donde se cumplen las siguientes restricciones:

- $0 < N \leq 10^3$
- $0 \leq p_i < q_i \leq 10^3$

Subtarea 4 (50 puntos)

Se probarán varios casos donde se cumplen las siguientes restricciones:

- $0 < N \leq 10^6$
- $0 \leq p_i < q_i \leq 10^6$

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
3	2
0 4	
4 7	
6 9	

Entrada de ejemplo	Salida de ejemplo
5	4
0 4	
3 6	
3 9	
10 15	
0 20	

Problema C

Secuencia de ADN

nombre clave: `adn`

El ácido desoxirribonucleico, o ADN, fue descubierto por Friedrich Miescher, quien en 1869 fue el primero en lograr aislarlo. Durante las siguientes décadas, el ADN permaneció sin ser muy estudiado, hasta que en 1944 algunos experimentos de Oswald Avery, Colin MacLeod y Maclyn McCarty lograron demostrar que la manipulación de ADN dentro de una bacteria podía afectar su comportamiento. Aproximadamente una década después, James Watson y Francis Crick presentaron su famoso modelo del ADN. De acuerdo a su modelo, una cadena de ADN está formada por una secuencia de nucleótidos, existiendo 4 tipos: adenina, guanina, citosina y timina.

La secuenciación de ADN es el proceso que permite determinar el orden de los nucleótidos en una cadena de ADN. Gracias a importantes avances científicos que comenzaron en la década de los 70, hoy es posible secuenciar el ADN de organismos completos.

Una vez que la secuenciación de una cadena está completa, es posible guardarla como un arreglo de números entre 0 y 3, donde cada número representa un nucleótido (adenina (0), guanina (1), citosina (2) y timina (3)). Con esta representación es posible hacer diferentes análisis sobre la cadena que permiten, entre otras cosas, encontrar defectos que pueden ser asociados a la presencia de ciertas enfermedades.

Recientes técnicas en el análisis de ADN requieren poder encontrar patrones cortos de largo tres dentro de una cadena de ADN. Esto permite encontrar puntos de interés para su posterior análisis. Considera, por ejemplo, el siguiente arreglo que representa una cadena de ADN:

1	0	2	1	3	2	1	3	0	2	2	1	3	2	1	0	1	2	3	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

En esta cadena se quiere encontrar las ocurrencias del patrón 132. Este patrón ocurre dos veces en la cadena, primero en la posición 3 y luego en la 11.

Dado un patrón, tu tarea es contar la cantidad de veces que dicho patrón ocurre dentro de una cadena de ADN. ¿Podrías ayudar a los científicos a encontrar estos patrones?

Entrada

La primera línea de la entrada contiene un entero N ($0 < N \leq 100\,000$) correspondiente al largo de la cadena de ADN. La segunda línea contiene N enteros, todos entre 0 y 3, que describen la secuencia de ADN. Finalmente, la tercera línea contiene tres enteros entre 0 y 3, que describen el patrón que se desea encontrar.

Salida

La salida debe contener un único entero correspondiente a la cantidad de veces que el patrón ocurre dentro de la cadena de ADN.

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
20 1 0 2 1 3 2 1 3 0 2 2 1 3 2 1 0 1 2 3 0 1 3 2	2

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

Problema D

Problema Presidencial: Paltas Promedio

nombre clave: palta

En el último discurso presidencial, se explicó el incremento de la actividad criminal a nivel país en base al reciente alza del precio de la palta Hass. Esta correlación, que la presidencia denomina ICPC (Índice de Correlación entre Paltas y Crimen), llegaría supuestamente al 65%. Ahora solo necesitan una forma de convencer a la población de que este índice completamente inventado es estadísticamente significativo. Para ello, la Presidencia de la República solicitó a la Organización de Coincidencias Intencionales, también conocida como OCI, encontrar la mayor muestra de datos posible que valide esta estadística.

Dentro de los datos que la OCI tiene a disposición, posee un arreglo con los valores diarios del ICPC en los últimos N días, y la misión es encontrar el período más largo de días consecutivos en que el promedio del ICPC alcanza un valor deseado T .

Por ejemplo, considere el siguiente arreglo con los valores del ICPC en los últimos 7 días:

0	60	100	40	33	25	75
0	1	2	3	4	5	6

Si se desea mostrar a la gente que el ICPC es de $T = 50\%$, tanto el período comprendido entre las posiciones 0 y 3 como el comprendido entre las posiciones 5 y 6 tienen un promedio de exactamente 50%. Como el período entre las posiciones 0 y 3 es más largo, la OCI prefiere reportar este.

La OCI ha descubierto que participas de la Olimpiada de Informática, y por lo mismo piensa que eres la persona perfecta para ayudarlos en su nuevo plan de manipulación social.

Entrada

La primera línea de la entrada contiene dos enteros, N y T , separados por un espacio.

La siguiente línea contiene N enteros A_i separados por espacios, que describen el arreglo A de valores del ICPC en los últimos N días.

Puedes asumir que las siguientes restricciones siempre se cumplen:

- $1 \leq N \leq 10^6$
- $0 \leq T \leq 100$
- $0 \leq A_i \leq 100$

Salida

La salida debe contener una única línea con dos enteros, S y L , separados por un espacio. Estos enteros deben describir el período de días consecutivos que se le mostrará a la población, es decir, el subarreglo más largo de A cuyo promedio es exactamente el valor T .

Específicamente, S debe ser la posición dentro de A donde comienza el subarreglo y L debe ser su largo. El índice 0 corresponde al primer elemento.

Si hay dos o más subarreglos del mismo largo que cumplen con los requisitos, se debe reportar el que comience primero (menor S).

Si no hay ningún subarreglo que cumpla con los requisitos, se debe imprimir -1 .

Subtareas y puntaje

Subtarea 1 (10 puntos)

Se probarán varios casos donde $N \leq 250$.

Subtarea 2 (20 puntos)

Se probarán varios casos donde $N \leq 5000$.

Subtarea 3 (40 puntos)

Se probarán varios casos donde $N \leq 10^6$ y además el arreglo A estará ordenado de menor a mayor, es decir, $A_i \leq A_{i+1}$ para todo i .

Subtarea 4 (30 puntos)

Se probarán varios casos donde $N \leq 10^6$ y el arreglo A no estará necesariamente ordenado.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
7 50 0 60 100 40 33 25 75	0 4

Entrada de ejemplo

13 3
2 3 4 3 15 4 4 4 0 1 2 4 2

Salida de ejemplo

0 4

Entrada de ejemplo

5 89
23 54 1 33 70

Salida de ejemplo

-1