Olimpiada Chilena de Informática 2021

11 de Diciembre, 2021

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
Alcohol Gel

nombre clave: `alcohol-gel`

Sebastián siempre tiene un negocio en mente. El año pasado su idea fue hacerse rico vendiendo alcohol gel. Lamentablemente su negocio fue un fracaso, y ahora tiene problemas para convencer inversionistas de que inviertan en su siguiente emprendimiento. Afortunadamente, Sebastián tuvo una brillante idea que cree le ayudará a ocultar su reciente fracaso. Si bien el balance total de la venta de alcohol gel fue negativo, si logra encontrar el periodo de tiempo en que el negocio tuvo el mejor rendimiento, podría mostrar solo esa parte de los datos a sus potenciales inversores.

Sebastián compró y vendió alcohol gel durante n días numerados de 1 a n. Para cada uno de esos días cuenta con los siguientes datos:

- La cantidad c_i de paquetes de alcohol gel que compró el día i.
- El precio p_i al que compró cada paquete el día i.
- La cantidad k_i de paquetes que logró vender el día i.
- El precio v_i al que vendió cada paquete el día i.

Notar que los paquetes que no son vendidos en un día pueden ser vendidos en los días siguientes y por lo tanto para un día i el valor k_i puede ser mayor que c_i.

El balance b_i del día i se define como la resta entre la cantidad de dinero que Sebastián recibió ese día producto de sus ventas y la cantidad de dinero que Sebastián gastó en comprar alcohol gel. Específicamente, $b_i = k_i \times v_i - c_i \times p_i$. Por ejemplo, si en el día 1 ocurre que $c_1 = 2, p_1 = 6, k_1 = 1, v_1 = 5$, entonces el balance b_1 para el día 1 es $(1 \times 5) - (2 \times 6) = -7$. El balance de un periodo de días consecutivos corresponde a la suma de los balances de esos días. Por ejemplo, continuando con el ejemplo anterior, si en el día 2 ocurre que $c_2 = 3, p_2 = 3, k_2 = 4, v_2 = 7$ entonces el balance del día 2 es $b_2 = 19$, y por tanto el balance del periodo comprendido entre los días 1 y 2 (incluidos) es $(-7) + 19 = 12$.

Dados los datos correspondientes a los n días, tu tarea es ayudar a Sebastián a identificar los días j y k tales que el periodo comprendido entre los días j y k tenga ganancia máxima. Es decir, el periodo cuyo balance sea positivo y que además su valor sea máximo. Si hay múltiples periodos con ganancia máxima, deberás identificar el de mayor largo. Si hay múltiples periodos con igual ganancia máxima e igual largo máximo entonces cualquiera de ellos es una respuesta válida. En caso de que ningún periodo tenga un balance positivo, la respuesta deberá ser `IMPOSIBLE`.

Entrada

La primera línea contiene un entero n ($0 < n \leq 10^5$) correspondiente al número de días. Las siguientes n líneas describen cada uno de los días. La línea i-ésima contiene los enteros c_i, p_i, k_i y v_i correspondientes al día i. Los valores cumplen $0 \leq c_i \leq 100$, $0 \leq k_i \leq 100$, $0 < p_i \leq 100$ y $0 < v_i \leq 100$. Se

garantiza además que la cantidad v_i de paquetes vendidos es siempre menor o igual que la cantidad c_i de paquetes comprados el día i más la cantidad de paquetes no vendidos en días anteriores, es decir, Sebastián solo puede vender lo que tiene en stock.

Salida

En caso de haber solución la salida debe contener tres enteros j, k y g. Los enteros j y k representan respectivamente el día de inicio y de fin del periodo con ganancia y largo máximo. El valor g corresponde a la ganancia en el periodo comprendido entre los días j y k. En caso de no haber ningún período de balance positivo deberás imprimir `IMPOSIBLE`.

Subtareas y puntaje

Subtarea 1 (20 puntos)

Se probarán varios casos donde $1 \leq n \leq 100$.

Subtarea 2 (30 puntos)

Se probarán varios casos donde $1 \leq n \leq 5000$.

Subtarea 3 (50 puntos)

Se probarán varios casos sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
2 2 6 1 5 3 3 4 7	2 2 19

Entrada de ejemplo	Salida de ejemplo
5 2 3 2 3 4 7 1 6 2 7 2 9 3 5 4 3 9 7 9 9	3 5 19

Entrada de ejemplo	Salida de ejemplo
3 2 6 1 5 5 3 4 3 9 9 9 9	IMPOSIBLE

Problema B

El juego del calendario

nombre clave: `calendario`

Alicia acaba de inventar un juego muy particular. Para poder jugarlo solo se necesita un lápiz y el calendario de un mes de 30 días. Los días del mes se numeran de 1 a 30 y dependiendo del mes, el primer día puede caer en distintos días de la semana. Por ejemplo, la siguiente imagen muestra el calendario de un mes donde el primer día cae un viernes.

Lu	**Ma**	**Mi**	**Ju**	**Vi**	**Sa**	**Do**
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

El juego comienza escogiendo un día inicial x, el cual debe ser tachado inmediatamente marcándolo como visitado. Posteriormente, en cada paso hay que aplicar una regla que indica el siguiente día al que hay que moverse el cual debe ser tachado marcándolo como visitado. El juego termina cuando una regla indica que hay que moverse a un día que ya había sido tachado o cuando no es posible aplicar una regla. La regla que hay que aplicar en cada paso depende del día de la semana en que cae el día actual. Las reglas para cada día de la semana se detallan a continuación:

Lunes Avanzar de uno en uno hasta encontrar el primer día no tachado y moverse a este día. Si al avanzar buscando un día sin tachar se llega al final del mes se debe *dar la vuelta* y continuar avanzando desde el inicio del mes. Si no queda ningún día sin tachar la regla no puede aplicarse y el juego termina en el día actual.

Martes Moverse al día que corresponde al *reflejo* del día actual. Es decir, si el día actual es i moverse al día $31 - i$.

Miércoles Si el día actual es par, moverse al día anterior. Si el día actual es impar, moverse al día siguiente.

Jueves Moverse al día que esta 10 posiciones adelante dando la vuelta en caso de llegar al final del mes.

Viernes Si el día actual i es par, moverse al día $i/2$. En caso contrario, moverse al día $3 \times i + 1$. Si $3 \times i + 1$ es mayor que 30 restarle 30 al resultado hasta que este sea menor o igual que 30.

Sábado Si el día actual es i moverse al día $2 \times i$. Si $2 \times i$ es mayor que 30 restarle 30 al resultado hasta que este sea menor o igual que 30.

Domingo Avanzar de dos en dos hasta encontrar un día no tachado y moverse a este día. Dar la vuelta al mes en caso de ser necesario. Si no puede avanzarse a ningún día sin tachar la regla no puede aplicarse y el juego termina en el día actual.

Considera el calendario del ejemplo mostrado anteriormente donde el primer día cae un viernes. Si escogemos el día inicial $x = 16$ este cae un sábado. La regla para el sábado dice que hay que moverse al día $2 \times 16 = 32$, como este es mayor que 30 restamos 30 una vez y nos movemos al día 2. El día 2 cae un sábado nuevamente y por lo tanto nos movemos al día $2 \times 2 = 4$. El día 4 cae un lunes. La regla del lunes dice que hay que encontrar el siguiente día no tachado el cual es el martes 5. La regla para el martes nos pide movernos al reflejo que en este caso corresponde al día $31 - 5 = 26$. El día 26 es nuevamente un martes y la regla nos pide movernos al día $31 - 5 = 5$ el cual ya había sido tachado y por lo tanto el juego termina en el día 5. La siguiente imagen muestra la secuencia de movimientos descritos anteriormente.

Lu	Ma	Mi	Ju	Vi	Sa	Do
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Dado el día de la semana en que cae el primer día del mes y el día inicial x con que comienza el juego tu tarea es determinar el día en que finaliza el juego.

Entrada

La entrada contiene una línea con dos enteros d ($0 \leq d \leq 6$) y x ($1 \leq x \leq 30$). El entero d corresponde al día de la semana en que cae el primer día del mes. Si d es 0 el mes comienza un lunes, si d es 1 el mes comienza un martes y así hasta el domingo. El entero x indica el día del mes en que comienza el juego.

Salida

La salida debe contener un único entero indicando el día del mes en el que termina el juego.

Subtareas y puntaje

Este enunciado no contiene subtareas, se entregará 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
`<sample input data here>`	`<sample output data here>`

Entrada de ejemplo	Salida de ejemplo
`<sample input data here>`	`<sample output data here>`

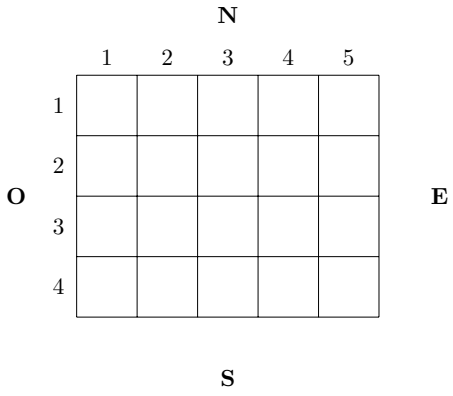
Problema C

Descenso en parapente

nombre clave: `parapente`

La nación de Nlogonia está cubierta completamente de montañas. Por esta razón, se ha convertido en un popular destino para los fanáticos de los parapentes. Debido a la multitud de montañas, Nlogonia presenta a los pilotos de parapente la posibilidad de escoger entre una inmensidad de *rutas* diferentes. Para escoger una nueva ruta, los pilotos solo deben elegir la cima de una montaña desde donde partir y una montaña de menor altura a la cual descender.

La geografía de Nlogonia es además peculiar pues sus montañas están dispuestas en una perfecta grilla de M filas y N columnas. Las filas en la grilla son numeradas de 1 a M y avanzan en la dirección sur. Las columnas están numeradas de 1 a N y avanzan en la dirección este. La siguiente imagen muestra un ejemplo de la numeración para una grilla de 4 filas y 5 columnas.



Decimos que la montaña en la fila i y columna j se encuentra en la posición (i, j) y que tiene altura $A(i, j)$. Por lo tanto, podemos representar una ruta con dos pares (i, j) y (k, l), donde el primer par corresponde a la posición de la montaña de inicio y el segundo par a la posición de la montaña final. La ruta entre las posiciones (i, j) y (k, l) es considerada *válida* si $A(i, j) > A(k, l)$. La *distancia de descenso* de una ruta válida corresponde a la diferencia de alturas entre las montañas de inicio y final $(A(i, j) - A(k, l))$. Notar que la distancia de descenso solo está definida para las rutas válidas y por lo tanto será siempre mayor que cero.

Maiki, una experimentada piloto, quiere batir el récord del mayor descenso en parapente. Naturalmente, Nlogonia presenta el lugar adecuado para encontrar la ruta perfecta. Maiki sabe que para poder siquiera acercarse al récord debe elegir una ruta donde la dirección del viento le favorezca. El viento en Nlogonia sopla siempre en dirección sureste. Por lo tanto, Maiki está interesada en encontrar una ruta cuya distancia de descenso sea máxima, pero solo entre las rutas válidas en dirección sureste. Específicamente, una ruta es en dirección sureste si las posiciones de inicio (i, j) y final (k, l) cumplen las siguientes condiciones:

- $i \leq k$
- $j \leq l$
- $i \neq k$ o $j \neq l$.

Maiki es una piloto experto, pero la programación no es lo suyo y necesita de tu ayuda para encontrar la ruta válida en dirección sureste con mayor distancia de descenso.

Entrada

La primera línea de la entrada contiene dos enteros M y N ($0 < M \leq 1000$, $0 < N \leq 1000$) correspondientes respectivamente a la cantidad de filas y columnas en la grilla. Cada una de las siguientes M líneas contiene N enteros y describe una fila en la grilla. El entero j-ésimo de la fila i-ésima corresponde a la altura $A(i, j)$ ($0 < A(i, j) \leq 10^9$) de la montaña en la posición (i, j).

Salida

La salida debe contener un único entero correspondiente a la mayor distancia de descenso entre las rutas válida en dirección sureste. Si no hay ninguna ruta válida en dirección sureste la salida debe contener un -1.

Subtareas y puntaje

Subtarea 1 (30 puntos)

Se probarán varios casos donde $M = 1$, es decir, la grilla contiene solo una fila. Notar que en este caso la dirección sureste es equivalente a la dirección este.

Subtarea 2 (70 puntos)

Se probarán varios casos sin restricciones adicionales.

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
1 5 3 20 95 80 20	75

Entrada de ejemplo	Salida de ejemplo
4 5 3 23 95 86 26 67 96 89 33 88 64 10 62 50 76 16 64 93 96 41	86

Problema D
Tormenta China

nombre clave: `segundo`

Sergio es un fanático de las apuestas. Cada semana va al hipódromo para levantar apuestas sobre los resultados de las carreras. Hasta ahora no ha tenido mucha suerte, pero tras años de estudios cree haber encontrado la fórmula infalible para hacerse millonario.

Su plan es entrar en una de las apuestas más riesgosas, pero a su vez una de las que más paga. La apuesta consiste en adivinar el tiempo exacto en que un caballo terminará la carrera. Esto parece imposible, pero Sergio tiene la información de todas las carreras pasadas de Tormenta China, su caballo favorito, con la cual está seguro que puede predecir el resultado.

Específicamente, para cada carrera pasada, Sergio tiene anotado en su libreta el tiempo en que Tormenta China terminó la carrera. Sergio sabe que Tormenta China está en una racha y por lo tanto obtendrá un buen tiempo. Además, después de mucho pensarlo, también determinó que es poco probable que alcance su mejor tiempo y por tanto se decidió por apostar por el segundo mejor tiempo. ¿Podrías ayudar a Sergio a encontrar este valor?

Entrada

La entrada del problema consiste en dos líneas. La primera línea contiene un entero n ($2 \leq n \leq 10^6$): el número de carreras pasadas para las cuales Sergio tiene información. Las segunda línea contiene n enteros. El i-ésimo entero t_i ($0 < t_i \leq 10^9$) indica el tiempo en milisegundos en que Tormenta China terminó la carrera i-ésima. Se garantiza que *no* todos los valores t_i serán iguales, es decir, hay al menos un valor que es distinto a los demás.

Salida

La salida debe contener un entero correspondiente al segundo mejor tiempo en que Tormenta China terminó una carrera pasada.

Subtareas y puntaje

Subtarea 1 (10 puntos)
Se probarán varios casos dónde $n \leq 3$.

Subtarea 2 (30 puntos)
Se probarán varios casos dónde $t_i \neq t_j$ para todo $1 \leq i, j \leq n$, es decir, la entrada no contiene valores repetidos.

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

Ejemplos de entrada y salida

Entrada de ejemplo	Salida de ejemplo
6 47000 57000 84000 56000 97000 20000	47000

Entrada de ejemplo	Salida de ejemplo
3 1 1 2	2