

OMI 2021 Día 1 — A. Encuentra Extremos

| | | | |
|-------------------------|-------|--------------------------|---------|
| Puntos | 100 | Límite de memoria | 128 MiB |
| Límite de tiempo (caso) | 200ms | Límite de tiempo (total) | 1m0s |

Existe un arreglo $A = [a_1, a_2, \dots, a_n]$ tal que $a_i \neq a_j$ para $i \neq j$. Debes encontrar la posición del menor y el mayor de ellos. Para hacerlo puedes invocar la función `esMenor(i, j)` que te devuelve verdadero si $a_i < a_j$ y falso si $a_i > a_j$.

Problema

Encuentra la posición del mayor y menor número del arreglo invocando la menor cantidad de veces la función `esMenor(i, j)`.







Implementación

Deberás implementar la función `void buscaExtremos(int n)`.

Para obtener información debes invocar la función `bool esMenor(int i, int j)` que devolverá `true` si el número en la posición i es menor que el número en la posición j y `false` en caso de que no lo sea. **Recuerda que los números van de la posición 1 a la n .**

Una vez que sepas la posición del mayor y menor número del arreglo deberás invocar la función `void respuesta(int posMenor, int posMayor)` a la que deberás pasarle como parámetros la posición de número menor y la posición del número mayor. **Sólo puedes invocar la función `respuesta` una vez.**

Ejemplo

| Entrada | Salida | Descripción |
|--|----------------|---|
| Función llamada  | Valor devuelto | Descripción |
| buscaExtremos(3)  | `-` | Esta será la llamada inicial a tu procedimiento <code>buscaExtremos</code> . |
| <code>esMenor(1, 2)</code>  | false | El número en la posición 1 es mayor que el número en la posición 2 . |
| <code>esMenor(2, 3)</code>  | true | El número en la posición 2 es menor que el número en la posición 3 por lo tanto es el menor de todos. |
| <code>esMenor(1, 3)</code>  | true | El número en la posición 1 es menor que el número en la posición 3 por lo tanto el número en la posición 3 es el mayor de todos. |
| <code>respuesta(2, 3)</code>  | `-` | Tras devolver el resultado usando la función <code>respuesta</code> tu programa debe terminar. |

Evaluación

Tu puntaje dependerá de la cantidad de veces que invoques la función `esMenor`. En algunas subtareas basta con que la cantidad de invocaciones sea menor que un límite para obtener los puntos. En la última subtarea tu puntaje dependerá de qué tanto te acerques al número de invocaciones de la solución oficial.

En cualquiera de los siguientes casos:

Si al invocar la función `esMenor(i, j)` utilizas posiciones que están fuera del rango del arreglo, es decir, menores a **1** o mayores a ***n***,

Si invocas la función `respuesta(posMenor, posMayor)` más de una vez o

Si excedes la cantidad máxima de invocaciones a `esMenor(i, j)` para esa subtarea, obtendrás **0** puntos.

Limites

$$1 \leq N \leq 500$$

Subtareas

(15 puntos): Obtienes *todos* los puntos si invocas `esMenor` a lo más n^2 veces.

(30 puntos): Obtienes *todos* los puntos si invocas `esMenor` a lo más $2n$ veces.

(55 puntos):

Si invocas `esMenor` $2n$ veces obtendrás **0** puntos.

Si invocas `esMenor` a lo más el número de veces que lo hace la solución oficial obtendrás el **100** de los puntos.

Si invocas `esMenor` entre $2n$ y número de la solución oficial obtendrás una cantidad parcial de los puntos.

Experimentación

OMI 2021 Día 1 — B. Marconacci

| | | | |
|-------------------------|------|--------------------------|---------|
| Puntos | 100 | Límite de memoria | 128 MiB |
| Límite de tiempo (caso) | 1.5s | Límite de tiempo (total) | 1m0s |

Marco acaba de descubrir la *secuencia de Fibonacci*. La *secuencia de Fibonacci* es una secuencia numérica muy famosa de los números **1, 1, 2, 3, 5, 8, 13, 21, 34, ...**. Al *i-ésimo* término de la secuencia de Fibonacci se le llama F_i .

El siguiente número de la secuencia se obtiene sumando los dos anteriores, por ejemplo:

$$F_1 = 1$$

$$F_2 = 1$$

$$F_3 = F_1 + F_2 = 1 + 1 = 2$$

$$F_4 = F_2 + F_3 = 1 + 2 = 3$$

...

De modo que la secuencia está determinada por los dos primeros términos F_1 y F_2 y todos los demás se van calculando en base a esos dos.

Marco quiere inventar su propia secuencia a la que llamará *secuencia de Marconacci* y se pregunta que pasaría si usa la misma construcción pero con F_1 y F_2 distintos, por ejemplo, si $F_1 = 3$ y $F_2 = 4$, en ese caso la secuencia sería: **3, 4, 7, 11, 18, 29, ...**

Ayuda a Marco a experimentar con diferentes valores iniciales para que pueda elegir la *secuencia de Marconacci* ideal. Marco te hará P preguntas, en cada pregunta te dará los dos valores iniciales A y B de la secuencia y te pedirá que calcules el valor del término i .

Problema

Escribe un programa que sea capaz de contestar las preguntas de Marco.

NOTA: Debido a que los valores en la *secuencia de Marconacci* pueden ser **muy** grandes deberás expresar el resultado módulo $10^9 + 7$. Recuerda que para sacar el módulo en C/C++ se usa el operador % y que para manipular números de ese rango lo recomendable es utilizar variables de tipo `long long int`.

Entrada

La primera línea tiene el número P , la cantidad de preguntas que te hará Marco.


Las siguientes P líneas contienen tres números A , B e i que representan el primer y segundo número de la secuencia respectivamente y el término del que se desea conocer el valor.

Salida

Debes escribir P líneas, una por cada pregunta. La línea debe contener un entero que representa el valor del término solicitado módulo $10^9 + 7$.

NOTA: El módulo es el residuo que deja un número al ser dividido por otro. Por ejemplo: $10 \bmod 3 = 1$ porque 10 deja residuo 1 al ser dividido por 3. En C++ esa operación puede hacerse escribiendo `10 % 3` (eso, para C++, significa "10 módulo 3"). En el caso del problema debes usar $10^9 + 7 = 1000000007$ en vez de **3** como el módulo a aplicar.

Ejemplo

| Entrada | Salida | Descripción |
|--|---|--|
| 5 1 1 9 3 4 1 5 10 2 3 4 6 7 3 7585 |  34 3 10 29 814888492 | <p>La primera pregunta usa los valores iniciales de la secuencia de Fibonacci. El 9no término de Fibonacci es 34.</p> <p>La segunda pregunta pide el primer término de la secuencia, el resultado es 3.</p> <p>La tercera pregunta pide al segundo término de la secuencia, el resultado el 10.</p> <p>La cuarta pregunta usa los valores de ejemplo de la descripción, el sexto término de la serie es 29.</p> <p>La quinta pregunta te permitirá ver si estás haciendo un correcto manejo de el módulo y los tipos de variable.</p> |

Límites

$$0 \leq A, B \leq 10^6$$

$$1 \leq P \leq 10^5$$

$$1 \leq i \leq 10^6$$

Subtareas

(5 puntos): $i \leq 30$

(10 puntos): $P = 1$

(15 puntos): En todas las preguntas se utiliza el mismo valor de A y de B .

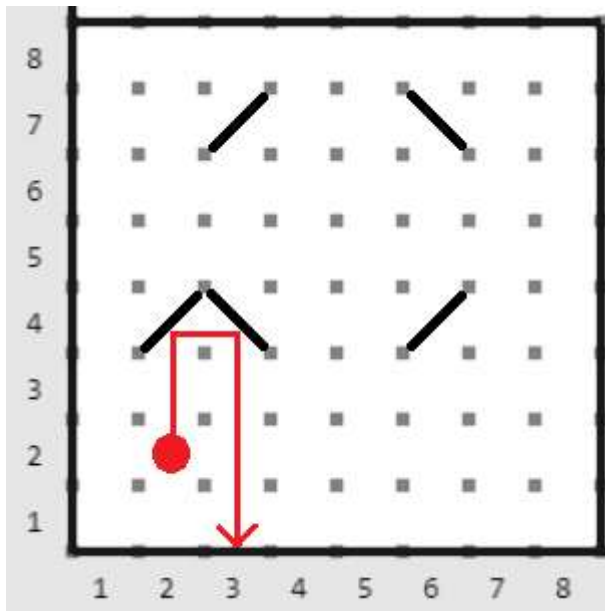
(30 puntos): $P, i \leq 1000$

(40 puntos): Sin restricciones adicionales

OMI 2021 Día 1 — C. Fotones

| | | | |
|-------------------------|-----|--------------------------|---------|
| Puntos | 100 | Límite de memoria | 128 MiB |
| Límite de tiempo (caso) | 3s | Límite de tiempo (total) | 1m0s |

Tu profesor de física construyó un dispositivo para experimentar con lasers. Este dispositivo consta de una cuadrícula de F filas por C columnas que tiene N espejos en algunas de sus casillas. Los espejos se encuentran colocados en diagonal de forma que cuando un láser choca con alguno se desvía 90° como se muestra en la siguiente figura:



Tu profesor repetirá el siguiente experimento E veces: él pondrá un rayo láser en alguna casilla orientado hacia norte, sur, este u oeste y lo disparará. Tu profesor desea saber en cuantos espejos se reflejará el láser antes de salir del dispositivo o si se quedará dando vueltas en el mismo.

Si en su camino, el láser rebota dos veces con un mismo espejo antes de salir del dispositivo, debes considerar que fueron dos rebotes distintos.

Para este problema considera que la fila **1** está al sur y la fila F está al norte. De igual forma la columna **1** está al oeste y la columna C está al este.

Problema

Escribe un problema que dado el tamaño de la cuadrícula y las posiciones de los N espejos pueda dar el resultado para los E experimentos de tu profesor.

Entrada

En la primera línea tres números enteros N , F y C , el número espejos, de filas y de columnas del dispositivo.

En las siguientes N líneas habrá un carácter que \ o / indicando la posición del espejo seguida de dos números f_i y c_i que indican la fila y la columna del i -ésimo espejo.

Un entero E , la cantidad de experimentos que hará tu profesor.

En las siguientes E líneas habrá un carácter que indica la orientación cuyo valor es **N**, **E**, **S** u **O** y dos enteros f_j y c_j que indican la fila y la columna del laser en el j -ésimo experimento.

Se te asegura que el láser nunca se coloca en la misma casilla que un espejo.

Salida

Deberás escribir E líneas, una por cada experimento. En cada línea deberá haber un número que indique la cantidad de espejos en los que el laser rebotó antes de salir del dispositivo o -1 si el laser se queda rebotando por siempre.

Ejemplo

| Entrada | Salida | Descripción |
|--|-----------------------|--|
| <pre> 5 8 8 / 4 2 \ 4 3 / 4 6 / 7 3 \ 7 6 4 N 2 2 E 4 4 S 8 3 O 6 8 </pre> | <pre> 2 -1 1 0 </pre> | <p>La imagen de la descripción muestra el primer experimento del profesor y como rebota con dos espejos antes de salir del dispositivo.</p> <p>La imagen previa muestra el segundo experimento en el que el laser continúa rebotando por siempre dentro del dispositivo.</p> |

Límites

$$\begin{aligned}
1 &\leq N \leq 10^5 \\
1 &\leq E \leq 10^5 \\
3 &\leq F, C \leq 10^5 \\
1 &\leq f_i, f_j \leq F \\
1 &\leq c_i, c_j \leq C
\end{aligned}$$

Subtareas

- (9 puntos): $N \leq 500, F, C \leq 100, E = 1$
- (10 puntos): $N \leq 500, F, C \leq 100$
- (13 puntos): $N \leq 1000, F, C \leq 10^5, E = 1$
- (20 puntos): $N \leq 1000, F, C \leq 10^5$

(18 puntos): $N \leq 10^5$, $F, C \leq 10^5$, $E = 1$

(30 puntos): $N \leq 10^5$, $F, C \leq 10^5$

OMI 2021 Día 1 — D. Capa de Invisibilidad

| | | | |
|-------------------------|-----|--------------------------|---------|
| Puntos | 100 | Límite de memoria | 128 MiB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 1m0s |

Harry anda de turista en la ciudad de Yakarta previo a la IOI 2022.

Yakarta puede representarse como un rectángulo de F filas por C columnas. Harry desea recorrer toda la ciudad. Para eso Harry comienza en algún lugar de la fila 1 y quiere llegar a la fila F . Como cualquier turista, Harry puede desplazarse en 4 direcciones: norte, sur, este u oeste.

Yakarta tiene N zonas peligrosas en las que Harry debe usar su capa de invisibilidad para pasar sin peligro. Sin embargo, sacar la capa es riesgoso (no debe usar artefactos mágicos en presencia de la gente), por lo que Harry quiere minimizar las veces que usa su capa. Además, Harry quiere comprar *recuerditos* (y con la capa no puede interactuar con la gente), por lo que sólo piensa usar la capa en las zonas peligrosas.

Cuando usa la capa, Harry no requiere respetar las reglas de tránsito, por lo que puede moverse a cualquiera de las 8 casillas circundantes. Además, Harry puede pasar de una zona peligrosa a otra sin quitarse la capa siempre y cuando ambas zonas peligrosas estén contiguas (tengan un lado o una esquina en común).

Cada zona peligrosa se representan como un rectángulo dentro de la ciudad.

Problema

Escribe un programa que dadas las dimensiones de la ciudad y las ubicaciones de las zonas peligrosas determine ¿cuál es el menor número de veces que Harry debe usar su capa para moverse desde la fila 1 hasta la fila F ?

Entrada

En la primera línea los números N , F y C . La cantidad de zonas peligrosas, la cantidad total de filas y la cantidad total de columnas en la ciudad.

En las siguientes N líneas habrá cuatro enteros f_{i_1} , c_{i_1} , f_{i_2} y c_{i_2} que corresponden a la fila y columna de la esquina superior-izquierda y la fila y columna de la esquina inferior-derecha de la i -ésima zona peligrosa.





Se asegura que no hay zonas peligrosas ni en la fila 1 ni en la fila F .

Se asegura que las zonas peligrosas no se traslapan entre sí.

Salida

Un número que indique la cantidad mínima de veces que Harry debe usar la capa para moverse de la fila 1 a la fila F

Ejemplo

| Entrada | Salida | Descripción |
|--|--------|---|
| <pre> 2 5 5 2 4 3 5 3 1 4 2 </pre>  | 0 | <p>El mapa de la ciudad se ve así. La letra X representa una zona peligrosa</p> <pre>XX XX.XX XX... </pre> <p>Si Harry se mueve de forma óptima, no es necesario usar la capa.</p> |
| <pre> 2 5 5 2 4 3 5 4 1 4 3 </pre>  | 1 | <pre>XX ...XX XXX.. </pre> <p>Aunque hay dos zonas peligrosas, Harry puede pasar de una a otra sin quitarse la capa, por lo que sólo tiene que ponérsela 1 vez.</p> |
| <pre> 2 5 1 2 1 2 1 4 1 4 1 </pre>  | 2 | |
| <pre> 6 6 3 2 2 2 2 3 1 3 1 3 3 3 3 4 2 4 2 5 1 5 1 5 3 5 3 </pre>  | 1 | |

Límites

$$1 \leq N \leq 50,000$$

$$1 \leq F, C \leq 10^9$$

Subtareas

(6 puntos): $F = 3, N = 1$

(10 puntos): $F = 3, C \leq 1000$

(12 puntos): $F = 3$

(9 puntos): $C = 1$

(20 puntos): $F, C \leq 500, N \leq 100$

(20 puntos): $N \leq 2000$

(23 puntos): Sin restricciones adicionales.