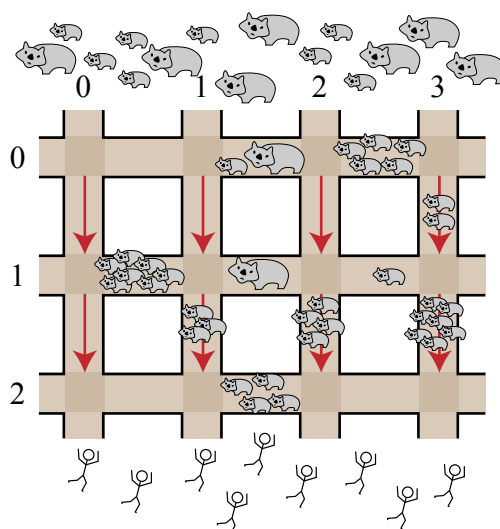


Las calles de Brisbane han sido tomadas por grandes wombats mutantes, Y tu debes poner a la gente a salvo.

Las calles de Brisbane tienen la forma de una gran cuadrícula. Existen R calles horizontales que van de este a oeste, numeradas de $0, \dots, (R-1)$ de norte a sur, y C calles verticales que van de norte a sur, numeradas de $0, \dots, (C-1)$ de oeste a este, como se muestra en la siguiente figura.



Los wombats han invadido desde el norte, y la gente se escapa a el sur. La gente puede correr a lo largo de las calles horizontales en cualquier dirección, pero en las calles verticales ellos *sólo correr hacia el sur*, hacia la seguridad.

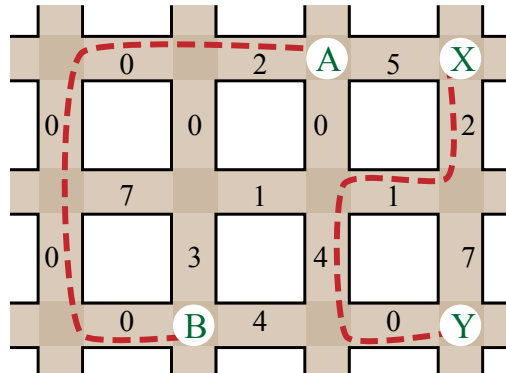
La intersección de la calle horizontal P con la calle vertical Q se denota (P, Q) . Cada segmento de calle entre dos intersecciones contiene alguna cantidad de wombats, y esta cantidad puede cambiar en el tiempo. Tu tarea es guiar a cada persona desde alguna intersección en el norte (en la calle horizontal 0) a alguna intersección en el sur (en la calle horizontal $R - 1$), llevándolos por una ruta que pase por la menor cantidad de wombats posible.

Para comenzar, se te dará el tamaño de la cuadrícula y la cantidad de wombats en cada segmento de calle. Luego de esto se te darán una serie de E eventos, que pueden ser:

- un *cambio*, que altera la cantidad de wombats en algún segmento de calle; o
- un *escape*, en donde alguna persona llega a una determinada intersección en la calle horizontal 0 , y tu debes encontrar una ruta hace una interseccion determinada en la calle horizontal $R - 1$ que pase por la menor cantidad de wombats posible.

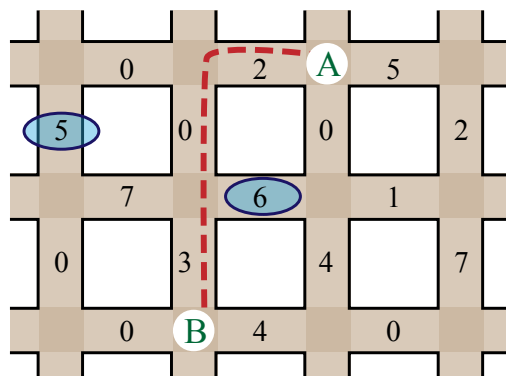
Tu debes manejar estos eventos implementando las siguientes rutinas `init()`, `changeH()`, `changeV()` y `escape()`, tal como se describen a continuación.

Ejemplos



La imagen anterior muestra un mapa inicial con $R = 3$ calles horizontales y $C = 4$ calles verticales, con el número de wombats señalados en cada segmento. Considera la siguiente serie de eventos:

- Una persona llega a la intersección $A = (0, 2)$ y desea escapar hacia la intersección $B = (2, 1)$. La menor cantidad de wombats por las que ella puede pasar es 2 , como lo indica la línea punteada.
- Otra persona llega a la intersección $X = (0, 3)$ y desea escapar a la intersección $Y = (2, 3)$. La menor cantidad de wombats por los que ella puede pasar es 7 , nuevamente indicado por una línea punteada.
- Dos eventos de cambio ocurren: el número de wombats en el segmento superior de la calle vertical 0 cambia a 5 , y el número de wombats en segmento medio de la calle horizontal 1 cambia a 6 . Ve los números circulo en la figura de más abajo.



- Una tercera persona llega a la intersección $A = (0, 2)$ y desea escapar hacia la intersección $B = (2, 1)$. Ahora la menor cantidad de wombats por los que la persona puede pasar es 5 , como lo indica la nueva línea punteada.

Implementación

Tú debes enviar un archivo implementando las funciones (procedimientos) `init()`, `changeH()` y `changeV()` y la función `escape()`, como se indica a continuación:

Tu Función: `init()`

C/C++ `void init(int R, int C, int H[5000][200], int V[5000][200]);`

Pascal
`type wombatsArrayType = array[0..4999, 0..199] of LongInt;
procedure init(R, C : LongInt; var H, V : wombatsArrayType);`

Descripción

Esta función te entrega el plano inicial del mapa, y te permite inicializar cualquier variable global y estructuras de datos globales. Esta será llamada solamente una vez, antes de cualquier llamada a `changeH()`, `changeV()` o `escape()`.

Parámetros

- `R` : El número de calles verticales.
- `C` : El número de calles horizontales.
- `H` : una matriz de dos dimensiones de tamaño $R \times (C-1)$ donde `H[P][Q]` da el número e wombats en el segmento horizontal de calle entre las intersecciones `(P,Q)` y `(P, Q+1)`.
- `V` : Una matriz de dos dimensiones de tamaño $(R-1) \times C$, donde `V[P][Q]` da el número e wombats en el segmento vertical de calle entre las intersecciones `(P,Q)` y `(P+1,Q)`.

Tu Función: changeH()

C/C++ `void changeH(int P, int Q, int W);`

Pascal `procedure changeH(P, Q, W: LongInt);`

Descripción

Esta función será llamada cuando el número de wombats en el segmento de calle horizontal entre las intersecciones (P, Q) y $(P + 1, Q)$ cambie.

Parámetros

- P : Indica que calle horizontal es afectada ($0 \leq P \leq R - 1$).
- Q : Indica entre cual par de calles verticales se encuentra el segmento ($0 \leq Q \leq C - 2$).
- W : El nuevo número de wombats en este segmento ($0 \leq W \leq 1,000$).

Tu Función: `changeV()`

C/C++ `void changeV(int P, int Q, int W);`

Pascal `procedure changeV(P, Q, W: LongInt);`

Descripción

Esta función será llamada cuando el número de wombats en el segmento de calle vertical entre las intersecciones `(P, Q)` y `(P + 1, Q)` cambie.

Parámetros

- `P` : Indica entre que calles horizontales se encuentra el segmento ($0 \leq P \leq R - 2$).
- `Q` : indica cual calle vertical es afectada ($0 \leq Q \leq C - 1$).
- `W` : El nuevo número de wombats en este segmento ($0 \leq W \leq 1,000$).

Tu Función: `escape()`

C/C++ `int escape(int V1, int V2);`

Pascal `function escape(V1, V2 : LongInt) : LongInt;`

Descripción

Esta función debe calcular cuál es la menor cantidad de wombats posibles por los que una persona tiene que pasar cuando viaja desde la intersección `(0, V1)` a `(R-1, V2)`.

Parámetros

- `V1` : Indica la posición en la que la persona comienza en la fila horizontal 0 ($0 \leq V1 \leq C-1$).
- `V2` : Indica la posición en la que la persona comienza en la fila horizontal `R-1` ($0 \leq V2 \leq C-1$).
- *Retorna*: La menor cantidad posible de wombats por los que la persona debe pasar.

Sesión de Muestra

La siguiente sesión describe el ejemplo anterior:

Function Call	Returns
<code>init(3, 4, [[0,2,5], [7,1,1], [0,4,0]], [[0,0,0,2], [0,3,4,7]])</code>	
<code>escape(2,1)</code>	2
<code>escape(3,3)</code>	7
<code>changeV(0,0,5)</code>	
<code>changeH(1,1,6)</code>	
<code>escape(2,1)</code>	5

Restricciones

- Time limit: 15 segundos
- Memory limit: 256 MiB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- A lo más 500 cambios (llamadas a `changeH()` o `changeV()`)
- A lo más 200,000 llamadas a `escape()`
- A lo más 1,000 wombats en cualquier segmento en cualquier momento

Sub-tareas

Sub-tarea	Puntos	Restricciones adicionales
1	9	$C = 1$
2	12	$R, C \leq 20$, y no habrán llamadas a <code>changeH()</code> o <code>changeV()</code>
3	16	$R, C \leq 100$, y habrán a lo más 100 llamadas a <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(None)

Experimentación

El evaluador de ejemplo en tu computador lee la entrada desde el archivo `wombats.in`, el cuál debe tener el siguiente formato:

- línea 1: `R C`
- línea 2: `H[0][0] ... H[0][C-2]`
- ...
- línea $(R + 1)$: `H[R-1][0] ... H[R-1][C-2]`
- línea $(R + 2)$: `V[0][0] ... V[0][C-1]`
- ...
- línea $(2R)$: `V[R-2][0] ... V[R-2][C-1]`
- siguiente línea: `E`
- siguiente `E` líneas: un evento por línea, en el orden en que ocurren los eventos

Si $C = 1$, las líneas vacías que contienen la cantidad de wombats en las calles horizontales (líneas 2 a la $R + 1$) no son necesarias.

La línea para cada evento debe encontrarse en los siguientes formatos:

- para indicar `changeH(P, Q, W)`: `1 P Q W`
- para indicar `changeV(P, Q, W)`: `2 P Q W`
- para indicar `escape(V1, V2)`: `3 V1 V2`

Por ejemplo el caso mostrado anteriormente debe ser codificado como de la siguiente forma:

```
3 4
0 2 5
7 1 1
0 4 0
0 0 0 2
0 3 4 7
5
3 2 1
3 3 3
2 0 0 5
1 1 1 6
3 2 1
```

Notas del Lenguaje

C/C++ Tienes que `#include "wombats.h"`.

Pascal Tienes que definir la `unit Wombats`. Todos los arrays están numerados desde `0` (no `1`).

Mira las plantillas de solución en tu ordenador como ejemplo.