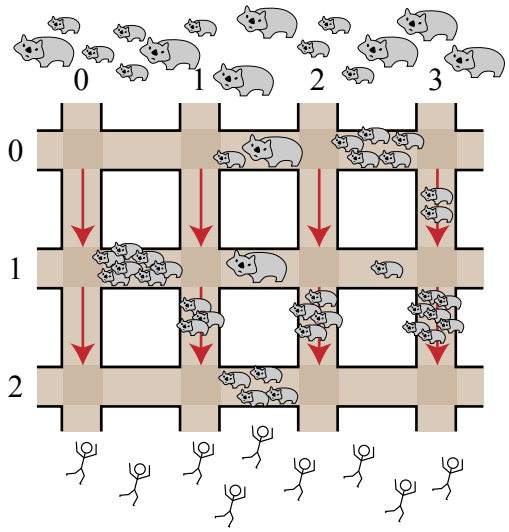
Una plaga de wombats mutantes ha invadido la ciudad de Brisbane. Es su deber poner a todos los ciudadanos a salvo.

Las calles de Brisbane forman una cuadrícula. Hay `R` calles horizontales que van de oriente a occidente, numeradas `0, …, (R - 1)` de norte a sur, y `C` calles verticales que van de norte a sur, numeradas `0, …, (C - 1)` de occidente a oriente, como se muestra en la siguiente figura.



Los wombats han iniciado su invasión por el norte de la ciudad y los ciudadanos escapan hacia el sur. Ellos pueden moverse por las calles horizontales en cualquiera de las dos direcciones, pero en las calles verticales únicamente pueden moverse hacia el sur.

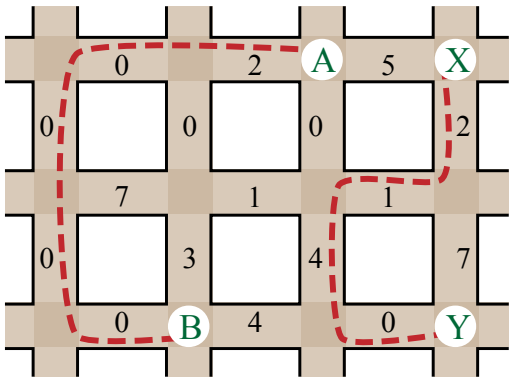
Se denota `(P,Q)` la intersección de la calle horizontal `P` con la calle vertical `Q`. Cada segmento de calle entre dos intersecciones contiene un determinado número de wombats, y ese número puede variar con el tiempo. Su trabajo consiste en guiar a cada persona desde una determinada intersección del norte (es decir, en la calle horizontal `0`) hacia una determinada intersección del sur (es decir, en la calle horizontal `R-1`), llevándola por una ruta que pase por el mínimo número posible de wombats.

Inicialmente, se le dará el tamaño de la cuadrícula y el número de wombats en cada segmento. A continuación se le darán una serie de `E` eventos, cada uno de los cuales puede ser:

- un *change*, que modifica el número de wombats en algún segmento; o
- un *escape*, donde una persona llega a una determinada intersección en la calle horizontal `0`, y su deber es encontrar una ruta en la calle horizontal `R-1` que pase por el mínimo número posible de wombats.

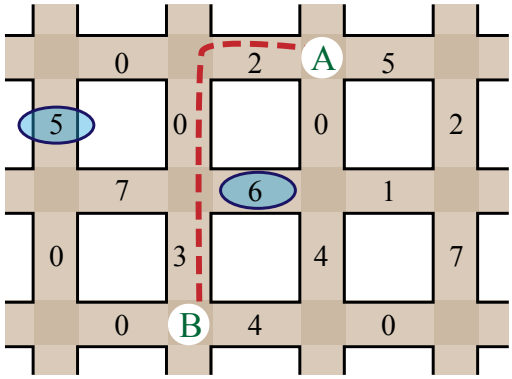
Para manejar estos eventos debe implementar las rutinas `init()`, `changeH()`, `changeV()` y `escape()`, que se describen a continuación.

Ejemplos



La figura anterior representa un plano inicial con `R = 3` calles horizontales y `C = 4` calles verticales. En cada segmento se indica el número inicial de wombats. Considere la siguiente secuencia de eventos:

- Una persona llega a la intersección `A = (0, 2)` y quiere escapar hacia la intersección `B = (2, 1)`. El mínimo número posible de wombats con el que se puede cruzar es `2`, siguiendo la ruta indicada por una línea punteada.
- Otra persona llega a la intersección `X = (0, 3)` y quiere escapar hacia la intersección `Y = (2, 3)`. El mínimo número de wombats con el que se puede cruzar es `7`, siguiendo de nuevo la ruta indicada por la línea punteada.
- A continuación suceden dos eventos de tipo change: el número de wombats del segmento de arriba perteneciente a la calle vertical `0` cambia a `5`, y el número de wombats en el segmento del medio perteneciente a la calle horizontal `1` cambia a `6`. Observe los números marcados en la figura que aparece a continuación.



- Una tercera persona llega a la intersección `A = (0, 2)` y quiere escapar hacia la intersección `B = (2, 1)`. Ahora el mínimo número de wombats con el que se puede cruzar es `5`, como se indica en la nueva línea punteada.

Implementación

Debe enviar un archivo que implemente los procedimientos `init()`, `changeH()` y `changeV()` y la función `escape()`, de la siguiente forma:

Su Procedimiento: `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

Este procedimiento le proporciona la distribución inicial del mapa, y le permite inicializar variables globales y estructuras de datos. Sólo se llamará una vez, antes de cualquier llamada a `changeH()`, `changeV()` o `escape()`.

Parámetros

- `R`: El número de calles horizontales.
- `C`: El número de calles verticales.
- `H`: Una matriz de dos dimensiones `R × (C - 1)`, donde `H[P][Q]` indica el número de wombats que hay en el segmento situado entre las intersecciones `(P, Q)` y `(P, Q + 1)`.
- `V`: Una matriz de dos dimensiones `(R - 1) × C`, donde `V[P][Q]` indica el número de wombats que hay en el segmento situado entre las intersecciones `(P, Q)` y `(P - 1, Q)`.

Su Procedimiento: `init()`

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

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

Descripción

Se llamará a este procedimiento cuando cambie el número de wombats en el segmento horizontal situado entre las intersecciones (P,Q) y $(P, Q + 1)$.

Parámetros

- `P`: Indica qué calle horizontal es la afectada ($0 \le P \le R - 1$).
- `Q`: Indica entre cuales dos calles verticales está situado el segmento ($0 \le Q \le C - 2$).
- `W`: El nuevo número de wombats en este segmento ($0 \le W \le 1{,}000$).

Su Procedimiento: `changeV()`

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

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

Descripción

Se llamará a este procedimiento cuando cambie el número de wombats en el segmento horizontal situado entre las intersecciones (P,Q) y $(P, Q + 1)$.

Parámetros

- `P`: Indica qué calle horizontal es la afectada ($0 \le P \le R - 1$).
- `Q`: Indica entre cuales dos calles verticales está situado el segmento ($0 \le Q \le C - 2$).
- `W`: El nuevo número de wombats en este segmento ($0 \le W \le 1{,}000$).

Su Función: `escape()`

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

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

Descripción

Esta función tiene que calcular el mínimo número de wombats que una persona debe pasar cuando va de la intersección $(0, V1)$ a la $(R-1, V2)$.

Parámetros

- `V1`: Indica donde empieza la persona en la fila horizontal 0 ($0 \le V1 \le C-1$).
- `V2`: Indica donde acaba la persona en la fila horizontal $R-1$ ($0 \le V2 \le C-1$).
- *Returns*: El mínimo número de wombats que la persona debe pasar.

Ejemplo de entrada

La siguiente entrada describe el ejemplo anterior:

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

Restricciones

- Límite de tiempo: 20 segundos
- Límite de memoria: 256 MB
- $2 \le R \le 5{,}000$
- $1 \le C \le 200$
- A lo más 500 cambios (llamadas tanto a `changeH()` como a `changeV()`)
- A lo más 200,000 llamadas a `escape()`
- A lo más 1,000 wombats en cualquier segmento en cualquier momento.

Subtareas

Subtarea	Puntos	Restricciones adicionales de entrada
1	9	$C = 1$
2	12	$R,C \le 20$, y no habrá llamadas a `changeH()` ni a `changeV()`
3	16	$R,C \le 100$, y como mucho habrá 100 llamadas a `escape()`
4	18	$C = 2$
5	21	$C \le 100$
6	24	*(Ninguna)*

Experimentación

El calificador en su computador leerá la entrada del archivo `wombats.in`, que tiene que estar en el siguiente formato:

- Línea 1: `R C`
- Línea2: `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`
- Siguientes `E` líneas: un evento por línea en el orden en el que ocurren los eventos

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

La línea para cada evento tiene que tener uno de los formatos siguientes:

- 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`

El ejemplo anterior se tendría que dar en el siguiente formato:

```
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++	Debe incluir la línea `#include "wombats.h"`.
Pascal	Debe definir `unit Wombats`. Todos los arreglos están numerados desde `0` (no desde `1`).

Vea los esqueletos provistos en su computador.