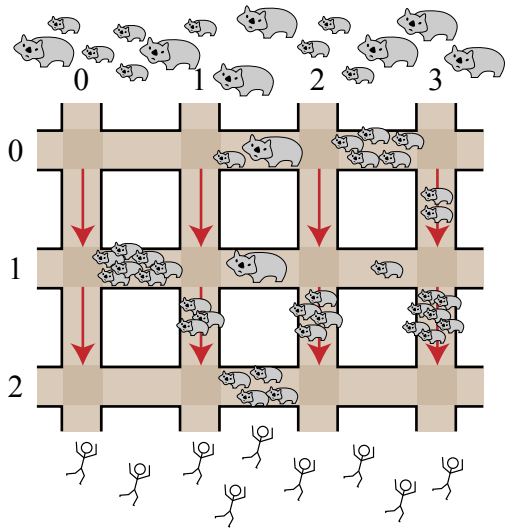
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia

wombats

Spanish — 1.0

La ciudad de Brisbane ha sido tomada por una horda de wombats mutantes, y debes guiar a la población a un lugar seguro.

Las calles en Brisbane están organizadas como una cuadricula. Existen `R` calles horizontales que van de este a oeste, numeradas `0, ..., (R - 1)` de norte a sur, y `C` calles verticales que van de norte a sur, numeradas `0, ..., (C - 1)` de oeste a este, como se muestra en la imagen de abajo.



Los wombats han invadido desde el norte y las personas están intentando escapar hacia el sur. Las personas pueden moverse sobre las calles horizontales en cualquier dirección pero, sobre las calles verticales *solo pueden moverse hacia el sur* (se mueven hacia un lugar seguro).

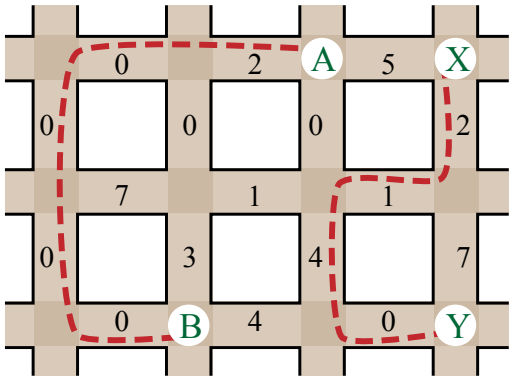
La intersección de una calle horizontal `P` y una calle vertical `Q` se denota como `(P, Q)`. Cada segmento de calle entre dos intersecciones contiene un determinado número de wombats, y ese número puede variar en el tiempo. Tu tarea es guiar a cada persona desde alguna intersección del norte dada (calle horizontal número `0`) hacia alguna intersección en el sur (calle horizontal número `R - 1`), de tal forma que en su ruta pase por la mínima cantidad de wombats posible.

Al inicio se te dará el tamaño de la matriz y el número de wombats en cada segmento de calle. Dado esto, se te darán `E` eventos, los cuales pueden ser:

- "change", que indica un cambio en la cantidad de wombats en un segmento de calle; o
- "escape", indica que una persona llegó a una determinada intersección de la calle horizontal número `0` y tú debes encontrar una ruta de esta intersección a otra intersección dada en la calle `R - 1` que pase por la mínima cantidad de wombats posible.

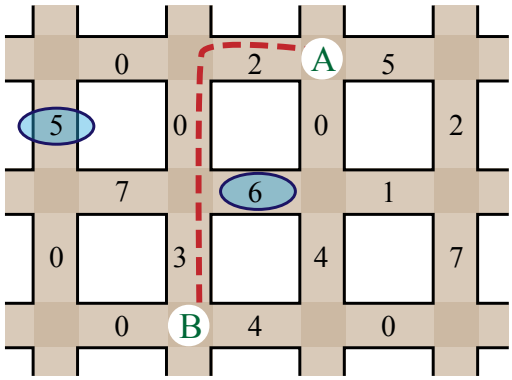
Debes implementar las funciones `init()`, `changeH()`, `changeV()` y `escape()`, como se describen a continuación:

Ejemplos



La imagen de arriba muestra el mapa inicial para `R = 3` calles horizontales y `C = 4` calles verticales, el número de wombats está especificado en cada segmento. Considera la siguiente serie de eventos:

- Una persona llega a la intersección `A = (0, 2)` y desea escapar por la intersección `B = (2, 1)`. La mínima cantidad de wombats por los que puede pasar es `2`, el camino está indicado por una línea punteada.
- Otra persona llega a la intersección `X = (0, 3)` y desea escapar por la intersección `Y = (2, 3)`. La mínima cantidad de wombats por donde puede pasar es `7`, el camino está indicado por una línea punteada.
- Dos cambios ocurren: el número de wombats del segmento vertical de hasta arriba cambia de `0` a `5` y el número de wombats del segmento vertical de en medio cambia de `1` a `6`. Los cambios están encerrados en círculos en la imagen de abajo.



- Una tercera persona llega a la intersección `A = (0, 2)` y desea escapar por la intersección `B = (2, 1)`. Ahora la mínima cantidad de wombats por donde puede pasar es `5` y el camino está indicado por una línea punteada.

Implementación

Debes mandar un archivo con las siguientes funciones implementadas `init()`, `changeH()`, `changeV()` y la función `escape()` de la siguiente manera:

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 da la configuración inicial del mapa y te permite inicializar cualquier variable global y estructuras de datos. Será llamada una sola vez y antes de cualquier llamada a `changeH()`, `changeV()` o `escape()`.

Parámeteros

- `R`: El número de calles horizontales.
- `C`: El número de calles verticales.
- `H`: Una matriz de tamaño `R × (C - 1)`, donde `H[P][Q]` es el número de wombats en el segmento horizontal entre las intersecciones `(P, Q)` y `(P, Q + 1)`.
- `V`: Una matriz de tamaño `(R - 1) × C`, donde `V[P][Q]` es el número de wombats en el segmento vertical 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

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

Parámetros

- `P`: Indica que calle horizontal es afectada ($0 \le P \le R - 1$).
- `Q`: Indica entre que dos calles verticales se encuentra el segmento ($0 \le Q \le C - 2$).
- `W`: La nueva cantidad de wombats que se encuentran en el segmento ($0 \le W \le 1{,}000$).

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 cambia el número de wombats de un segmento de calle vertical entre las intersecciones (P, Q) y $(P + 1, Q)$.

Parámetros

- `P`: Indica entre que dos calles horizontales se encuentra el segmento ($0 \le P \le R - 2$).
- `Q`: Indica que calle vertical es afectada ($0 \le Q \le C - 1$).
- `W`: La nueva cantidad de wombats que se encuentran en el segmento ($0 \le W \le 1{,}000$).

Tu función: `escape()`

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

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

Descripción

Esta función debe calcular la mínima cantidad de wombats por los que la persona tiene que pasar si quiere viajar de la intersección $(0, V1)$ a la $(R-1, V2)$.

Parámetros

- `V1`: Indica donde inicia la persona sobre el renglón `0` ($0 \le V1 \le C-1$).
- `V2`: Indica donde la persona termina sobre el renglón `R-1` ($0 \le V2 \le C-1$).
- *Returns*: La mínima cantidad de wombats por los que tiene que pasar la persona.

Sesión de ejemplo

La siguiente sesión describe el ejemplo de arriba:

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

- Tiempo Límite: 20 segundos
- Límite de Memoria : 256 MiB
- $2 \le R \le 5{,}000$
- $1 \le C \le 200$
- A lo más 500 cambios (llamadas a `changeH()` o `changeV()`)
- A lo más 200,000 llamadas a la función `escape()`
- A lo más 1,000 wombats se encontrarán en cada uno de los segmentos en cualquier momento.

Subtareas

Subtarea	Puntos	Restricciones Adicionales a la Entrada
1	9	$C = 1$
2	12	$R,C \le 20$, y no habrá llamadas a `changeH()` o `changeV()`
3	16	$R,C \le 100$, y habrá a lo más 100 llamadas a `escape()`
4	18	$C = 2$
5	21	$C \le 100$
6	24	*(Ninguna)*

Experimentación

El evaluador de ejemplo de tu computadora leerá la entrada del archivo `wombats.in`, que debe estar en el siguiente formato:

- linea 1: `R C`
- linea 2: `H[0][0] … H[0][C-2]`
- …
- linea $(R + 1)$: `H[R-1][0] … H[R-1][C-2]`
- linea $(R + 2)$: `V[0][0] … V[0][C-1]`
- …
- linea $(2R)$: `V[R-2][0] … V[R-2][C-1]`
- siguiente linea : `E`
- siguientes `E` lineas: una por linea, en el orden que ocurren los eventos.

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

La linea de cada evento debe estar en el siguiente formato:

- 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 ejemplo de arriba debiera ser provisto 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++	Debes incluir `#include "wombats.h"`.
Pascal	Debes definir la Unidad `unit Wombats`. Todos los arreglos están numerados iniciando en `0` (no `1`).

Revisa los templates de las soluciones dentro de tu computadora para ver ejemplo.