



## International Olympiad in Informatics 2013

6-13 July 2013

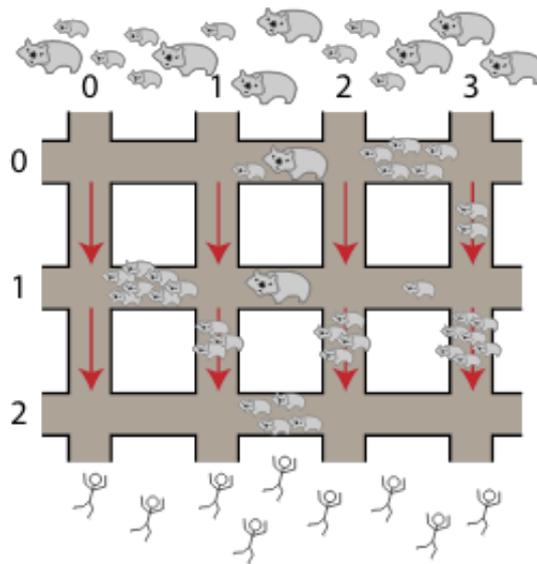
Brisbane, Australia

# wombats

Spanish – 1.1

¡Una plaga de wombats mutantes ha invadido la ciudad de Brisbane! ¡Rápido, debes poner a salvo a todos los ciudadanos!

Las calles de Brisbane forman una cuadrícula. Hay  $R$  calles horizontales que van de este a oeste, y están numeradas como  $0, \dots, (R - 1)$  en orden de norte a sur, y  $C$  calles verticales que van de norte a sur, numeradas como  $0, \dots, (C - 1)$  de oeste a este, como se ilustra en el siguiente dibujo.



Los wombats han iniciado su invasión por el norte de la ciudad, y los ciudadanos escapan hacia el sur. La gente puede moverse por las calles horizontales en cualquiera de los dos sentidos, pero en las calles verticales únicamente pueden moverse hacia el sur, buscando la seguridad.

Denotamos como  $(P, Q)$  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. Tu 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.

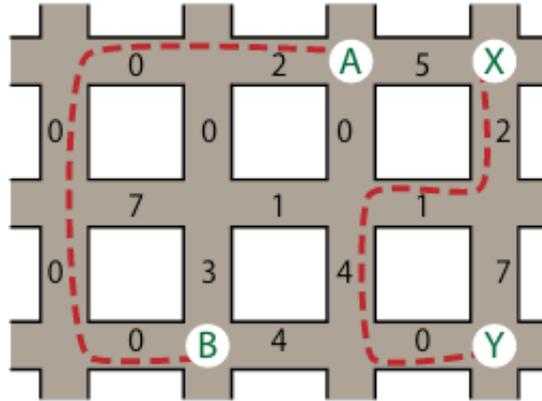
Inicialmente, se te da el tamaño de la cuadrícula y el número de wombats en cada segmento. A continuación se te darán una serie de  $E$  eventos, cada uno de los cuales puede ser:

- 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 debes encontrar una ruta en la calle horizontal  $R-1$  que pase por el mínimo número posible de wombats.

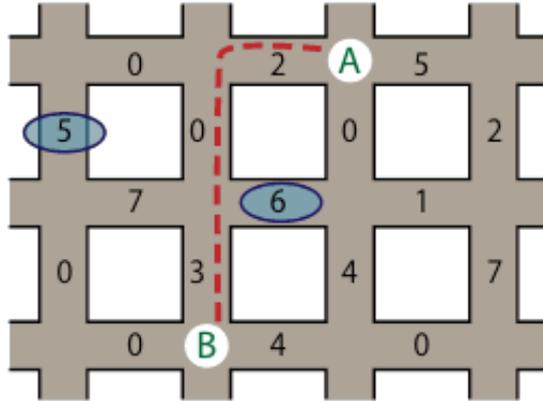
Para tratar estos eventos debes implementar las rutinas `init()`, `changeH()`, `changeV()` y `escape()`, que se describen a continuación.

## Ejemplos



La figura de arriba representa un plano inicial con  $R = 3$  calles horizontales y  $C = 4$  calles verticales, y en cada segmento se indica el número inicial de wombats. Considera la siguiente secuencia de eventos:

- Una persona llega a la intersección  $A = (0, 2)$  y quiere escapar hacia la intersección  $B = (2, 1)$ . El menor número posible de wombats con el que se puede cruzar es  $2$ , siguiendo la ruta indicada por la línea discontinua.
- Otra persona llega a la intersección  $X = (0, 3)$  y quiere escapar hacia la intersección  $Y = (2, 3)$ . El menor número de wombats con el que se puede cruzar es  $7$ , siguiendo de nuevo la ruta indicada por la línea discontinua.
- 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 de en medio perteneciente a la calle horizontal  $1$  cambia a  $6$ . Observa 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 discontinua.

## Implementación

Se te pide que envíes un archivo que implemente los procedimientos `init()`, `changeH()` y `changeV()` y la función `escape()`, de la siguiente forma:

### Tu 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 te proporciona la distribución inicial del plano, y te 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.
- $H$ : Un array bidimensional de tamaño  $R \times (C - 1)$ , donde  $H[P][Q]$  indica el número de wombats que hay en el segmento situado entre las intersecciones  $(P, Q)$  y  $(P, Q + 1)$ .

- $V$ : Un array bidimensional de tamaño  $(R - 1) \times C$ , donde  $V[P][Q]$  indica el número de wombats que hay en el segmento situado entre las intersecciones  $(P, Q)$  y  $(P + 1, Q)$ .

### Tu Procedimiento: `changeH()`

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

### Tu 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 vertical situado entre las intersecciones  $(P, Q)$  y  $(P + 1, Q)$ .

#### Parámetros

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

### Tu Función: `escape()`

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

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

## Descripción

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

## Parámetros

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

## Ejemplo de entrada

La siguiente entrada describe el ejemplo anterior:

Function Call	Returns
<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: 20 seconds
- Memory limit: 256 MiB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- Como mucho 500 cambios (llamadas tanto a `changeH()` como a `changeV()`)
- Como mucho 200,000 llamadas a `escape()`
- Como mucho 1,000 wombats en cualquier segmento en cualquier momento.

## Subtareas

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

## Experimentación

El grader de tu ordenador leerá la entrada del archivo `wombats.in`, que tiene que estar en el formato siguiente:

- 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`
- 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
```

---

## Apuntes 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.