

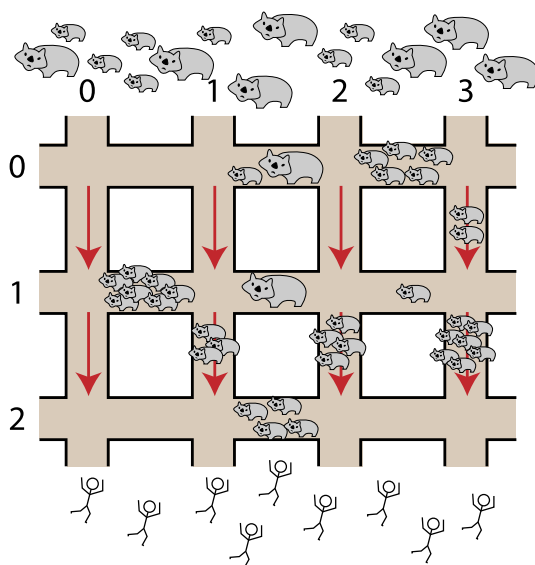


International Olympiad in Informatics  
2013  
6–13 July 2013  
Brisbane, Australia

wombats  
中文(澳門) —  
1.1

布爾斯班市正受著一群變種的大袋熊侵襲，而你必須要帶領人們到安全的地方。

布爾斯班市的道路是被編排成一個大的網格。那裡共有  $R$  條東西向的水平路，它們分別由北至南編號為  $0, \dots, (R - 1)$ ，同時亦有  $C$  條南北向的垂直路，它們分別由西至東編號為  $0, \dots, (C - 1)$ ，有如下圖所示。



那些袋熊正由北方入侵，而人們則要逃亡至南方。人們在水平路上可以走向東、西兩個方向，但他們在垂直路上只能向較安全的南方走。

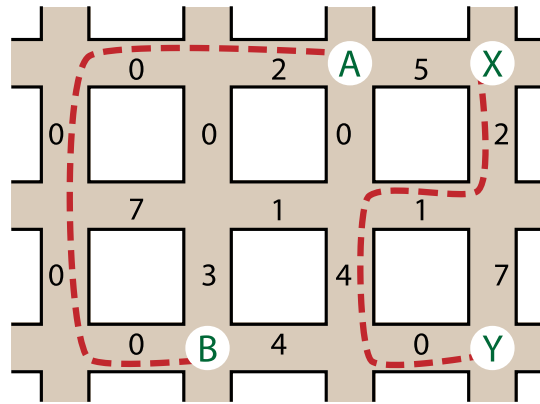
水平路  $P$  與垂直路  $Q$  的交匯點被定義為  $(P, Q)$ 。在每對交匯點之間的每條路段會有一些袋熊存在，其數目可能會隨著時間而改變。你的任務是要引領一些人從一個在北面的交匯點（在水平路  $0$  上）去到另外些在南方的指定交匯點（在水平路  $R - 1$  上），你要引領這些人們經過盡量少的袋熊。

在開始時系統會提供給你道路網格的大小及在每一段路段上袋熊的數目。在這之後，系統將提供給你一系列的事件  $E$ ，每個事件可以是以下其中之一：

- 一個 `change` 代表在某些路段上出現袋熊數目的改變，或
- 一個 `escape` 代表某人到達了一個在水平路  $0$  上的指定交匯點，你需要幫這人找出一條經過最少袋熊而到南方水平路  $R-1$  上的一個指定交匯點去的路徑。

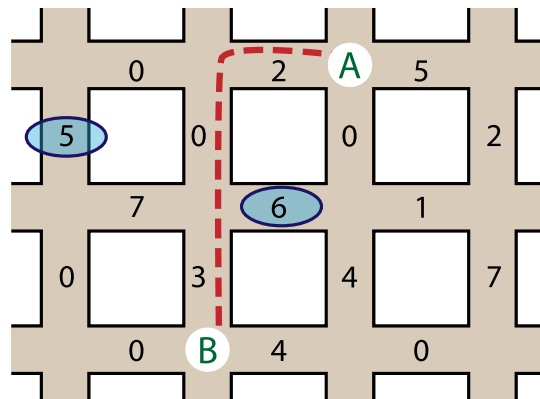
你的程式必須要通過編寫一些子程式來應對這些事件：`init()`，`changeH()`，`changeV()` 及 `escape()`，它們要如下面所描寫。

## 例子



圖中表示一張有  $R = 3$  條水平路及  $C = 4$  條垂直路的地圖的最初情況，地圖上亦標有最初時各路段上的袋熊數目。現考慮以下事件：

- 有一個人到達了交匯點  $A = (0, 2)$  並希望逃亡到交匯點  $B = (2, 1)$ 。這時，她會經過的袋熊的最少可能數目是  $2$ ，圖內以虛線標示之。
- 另一個人到達了交匯點  $X = (0, 3)$  並希望逃亡到交匯點  $Y = (2, 3)$ 。這時，她會經過的袋熊的最少可能數目是  $7$ ，同樣地，該路線在圖內以虛線標示之。
- 之後有兩個“change”事件出現：在垂直路  $0$  頂的一段路上的袋熊數目變為  $5$ ，而在水平路  $1$  的中間一段路的袋熊數目變為  $6$ 。參看下圖中圈起的數字。



- 有第三個人到達交匯點  $A = (0, 2)$  並希望逃亡到交匯點  $B = (2, 1)$ 。而這時她最少會經過袋熊的數目是  $5$ ，她所經的路線在圖中以虛線標示出來。

## 程式實現

你應該提交一個檔案編寫子程序 `init()`，`changeH()` and `changeV()` 和函數 `escape()`，如下：

你的子程序：**`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);`

### 描述

這個子程序會給你地圖的初始布局和允許你初始化所有的全局變量和其他數據結構。它只能被調用一次，而且要在調用 `changeH()`，`changeV()` 或 `escape()` 前。

### 函數的參數

- `R`：水平路的數目。
- `C`：垂直路的數目。
- `H`：一個二維數組，它的大小是  $R \times (C - 1)$ ，其中 `H[P][Q]` 給出了在交匯點 `(P, Q)` 和 `(P, Q + 1)` 之間水平路段中的袋熊數目。
- `V`：一個二維數組，它的大小是  $(R - 1) \times C$ ，其中 `V[P][Q]` 給出了在交匯點 `(P, Q)` 和 `(P + 1, Q)` 之間的垂直路段中的袋熊數目。

你編寫的子程序：**`changeH()`**

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

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

### 描述

當在交匯點 `(P, Q)` 和 `(P, Q + 1)` 間的水平路段上的袋熊數目有所改變時，這子程式就會被調用。

### 函數的參數

- `P`：表示哪條水平路受影響 ( $0 \leq P \leq R - 1$ )。
- `Q`：表示路段在哪兩條垂直路之間 ( $0 \leq Q \leq C - 2$ )。
- `W`：表示在一路段出的袋熊的最新數目 ( $0 \leq W \leq 1,000$ )。

你編寫的子程序：**changeV()**

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

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

### 描述

當在交匯點  $(P, Q)$  及  $(P + 1, Q)$  之間的垂直路段上的袋熊數目有所變化時，這個子程式就會被調用。

### 函數的參數

- $P$ ：表示一條路段是在哪兩條水平路之間 ( $0 \leq P \leq R - 2$ )。
- $Q$ ：表示哪一條垂直路受到影響 ( $0 \leq Q \leq C - 1$ )。
- $W$ ：表示在一條路段上的最新袋熊數目 ( $0 \leq W \leq 1,000$ )。

你的函數：**escape()**

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

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

### 描述

這個函數 (function) 應該能計算當一個人由交匯點  $(0, V1)$  至  $(R-1, V2)$  時，所經過的最少袋熊數目。

### 函數的參數

- $V1$ ：指出第 0 橫行中的人在那裏開始 ( $0 \leq V1 \leq C-1$ )。
- $V2$ ：指出第  $R-1$  橫行中的人在那裏結束 ( $0 \leq V2 \leq C-1$ )。
- 返回值：某人必須經過的最少的袋熊數目。

## 樣例部分

以下的部份描述了上述的例子：

| 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       |

## 限制條件

- 時間限制：20 秒
- 記憶體限制：256 MB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- 最多 500 次調用 `change`（調用 `changeH()` 或 `changeV()`）
- 最多 200,000 次調用 `escape()`
- 在任何路段上，任何時間最多只有 1,000 隻袋熊

## 子任務

| 子任務 | 得分 | 附加輸入限制   |
|-----|----|--|
| 1   | 9  | $C = 1$  |
| 2   | 12 | $R, C \leq 20$ ，永遠不會調用 <code>changeH()</code> 或 <code>changeV()</code> |
| 3   | 16 | $R, C \leq 100$ ，最多調用 <code>escape()</code> 100次                       |
| 4   | 18 | $C = 2$  |
| 5   | 21 | $C \leq 100$   |
| 6   | 24 | (沒有)   |

## 實驗

樣例評分程式會在你的電腦上讀取以下名稱的輸入檔案 `wombats.in`，它的格式如下：

- 第1行： `R C`

- 第2行: `H[0][0] ... H[0][C-2]`
- ...
- 第  $(R + 1)$  行: `H[R-1][0] ... H[R-1][C-2]`
- 第  $(R + 2)$  行: `V[0][0] ... V[0][C-1]`
- ...
- 第  $(2R)$  行: `V[R-2][0] ... V[R-2][C-1]`
- 下一行: `E`
- 下 `E` 行: 每行有一個事件，以事件發生的順序出現

若  $C = 1$ ，則第 2 至第  $R + 1$  行用來表示水平路上有的袋熊數目的空行是沒有需要的。

每行對於每個事件必須是以下其中一種形式：

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

例如，上述例子的格式如下：

```
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
```

## 編程語言注意事項

C/C++ 你必須包含 `#include "wombats.h"` .

Pascal 你必須定義 (define) `unit Wombats` . 所有的數組必需由 0 (不是 1) 開始。

請參閱在你的電腦上的例子題解模板。