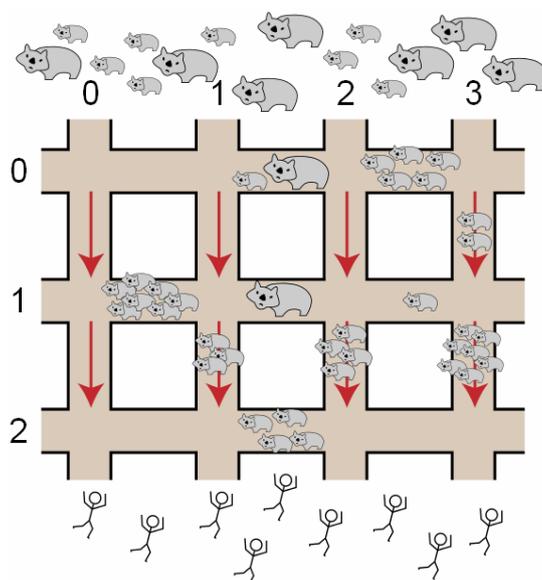


布里斯本現在已經被大量突變的袋熊 (wombat) 佔領，而你必須帶領人們到安全的地方。布里斯本的道路是以大型網格 (grid) 的形式呈現。有 R 條東西向的水平道路，由北到南依序標示為 $0, \dots, (R-1)$ ；另外有 C 條南北向的垂直道路，由西到東依序標示為 $0, \dots, (C-1)$ ，如下圖所示。



袋熊由北方入侵，而人類則逃往南方。人類在水平方向的道路可以往東或往西逃跑，但是在垂直方向的道路只能往南方，也就是安全的地方前進。

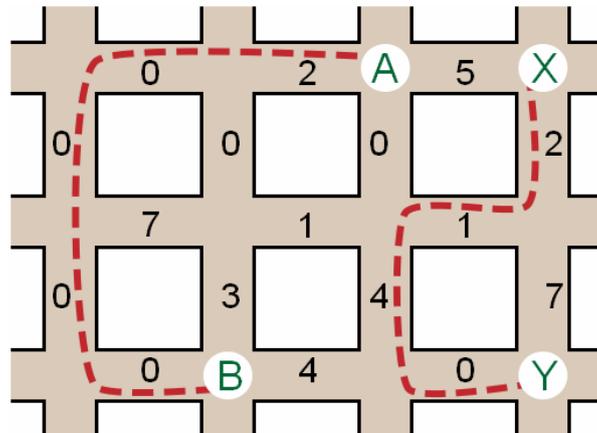
水平道路 P 與垂直道路 Q 的交會處以 (P, Q) 表示。兩個交會處之間的道路區段 (segment) 會包含某數量的袋熊，這個數量會隨著時間而改變。你的任務是引導每個人從最北方 (水平道路 0) 的某個給定交會處，到達最南方 (水平道路 $R-1$) 的某個給定交會處，使他們行經的路線盡可能遇到最少數量的袋熊。

一開始會給定網格的大小以及每個道路區段的袋熊數量。接著你將會被給予一連串共 E 個事件，每個事件可能是：

- 改變 (*change*)，會更改某個道路區段的袋熊數量；或是
- 逃跑 (*escape*)，也就是某個人出現在水平道路 0 的某個已給定交會處，而你必須找出一條路線到達水平道路 $R-1$ 的一個已給定交會處，使得途中盡可能遇到最少數量的袋熊。

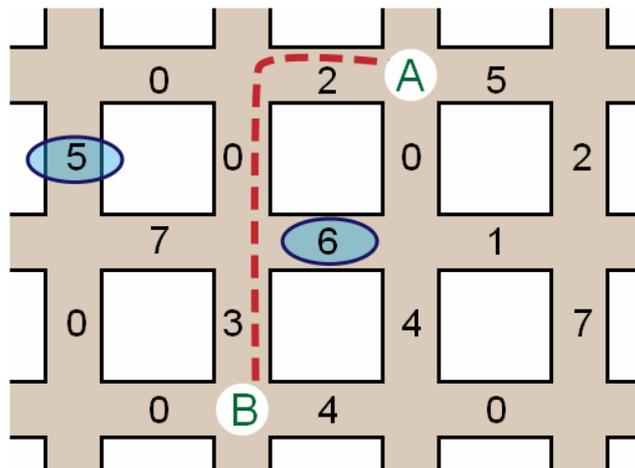
你必須透過實作如下所述的 `init()`、`changeH()`、`changeV()`，以及 `escape()` 等函式來處理這些事件。

範例 (Examples)



上圖顯示水平道路數量 $R=3$ 以及垂直道路數量 $C=4$ 的一個初始地圖，每個道路區段皆標示了該區段的袋熊數量。考量以下的連續事件：

- 一個人出現在交會處 $A = (0, 2)$ ，並且希望逃到交會處 $B = (2, 1)$ 。她遇到的袋熊數量最少可以是 2，如虛線所示。
- 另一個人出現在交會處 $X = (0, 3)$ ，並且希望逃到交會處 $Y = (2, 3)$ 。他遇到的袋熊數量最少可以是 7，同樣如虛線所示。
- 兩個改變的事件產生：垂直道路 0 最頂端區段的袋熊數量更改為 5，而水平道路 1 中間區段的袋熊數量更改為 6，詳見下圖圈起來的數字。



- 第三個人出現在交會處 $A = (0, 2)$ ，並且希望逃到交會處 $B = (2, 1)$ 。現在她遇到的袋熊數量最少會是 5，如新的虛線所示。

程式實作 (Implementation)

你應該提交 (submit) 一個檔案來實作程序 (procedures) `init()`、`changeH()` 與 `changeV()`，以及函式 (function) `escape()` 如下：

你的程序： `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);`

敘述

這個程序給你地圖的初始佈局，並且允許你初始化任何全域變數與資料結構。在呼叫 `changeH()`、`changeV()`，或 `escape()` 之前必須先呼叫這個程序，並且只會呼叫一次。

參數

- R ：水平道路的數目
- C ：垂直道路的數目
- H ：一個大小為 $R \times (C - 1)$ 的二維陣列，其中 $H[P][Q]$ 給定介於交會處 (P, Q) 與 $(P, Q + 1)$ 的水平道路區段所包含的袋熊數量。
- V ：一個大小為 $(R - 1) \times C$ 的二維陣列，其中 $V[P][Q]$ 給定介於交會處 (P, Q) 與 $(P + 1, Q)$ 的垂直道路區段所包含的袋熊數量。

你的程序： changeH()

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

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

敘述

當介於交會處 (P, Q) 與 $(P, Q + 1)$ 之間的水平道路區段的袋熊數量變更時，這個程序將會被呼叫。

參數

- P：指出哪個水平道路被影響 ($0 \leq P \leq R - 1$)。
- Q：指出區段位於哪兩個垂直道路之間 ($0 \leq Q \leq C - 2$)。
- W：在這個道路區段新的袋熊數量 ($0 \leq W \leq 1,000$)。

你的程序： `changeV()`

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

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

敘述

當介於交會處 (P, Q) 與 $(P + 1, Q)$ 之間的垂直道路區段的袋熊數量變更時，這個程序將會被呼叫。

參數

- P ：指出區段位於哪兩個水平道路之間 ($0 \leq P \leq R - 2$)。
- Q ：指出哪個垂直道路被影響 ($0 \leq Q \leq C - 1$)。
- W ：在這個道路區段新的袋熊數量 ($0 \leq W \leq 1,000$)。

你的函式： `escape()`

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

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

敘述

這個函式將計算當一個人由交會處 $(0, V1)$ 行進到 $(R - 1, V2)$ 時，至少一定會遇到的袋熊數量。

參數

- $V1$ ：指出這個人從水平道路 0 的哪個地方開始出現 ($0 \leq V1 \leq C - 1$)。
- $V2$ ：指出這個人最後出現在水平道路 $(R - 1)$ 的哪個地方 ($0 \leq V2 \leq C - 1$)。
- *Returns*：這個人至少一定會遇到的袋熊數量。

實例 (Sample Session)

以下描述上述例子：

| 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

- 時間限制：20 秒
- 記憶體限制：256 MiB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- 最多 500 次改變（呼叫 `changeH()` 或 `changeV()`）
- 最多呼叫 200,000 次 `escape()`
- 在任何時間在任何區段最多 1,000 隻袋熊

子任務

| 子任務 | 配分 | 額外的輸入限制 |
|-----|----|---|
| 1 | 9 | $C = 1$ |
| 2 | 12 | $R, C \leq 20$, 並且不呼叫 <code>changeH()</code> 或是 <code>changeV()</code> |
| 3 | 16 | $R, C \leq 100$, 並且最多呼叫 100 次 <code>escape()</code> |
| 4 | 18 | $C = 2$ |
| 5 | 21 | $C \leq 100$ |
| 6 | 24 | (無) |

實驗

在你電腦的範例評分機制（grader）將會由檔案 `wombats.in` 讀取輸入資訊，並且必須是以下格式：

- 第 1 行: $R\ C$
- 第 2 行: $H[0][0] \dots H[0][C - 2]$
- ...
- 第 $(R + 1)$ 行: $H[R - 1][0] \dots H[R - 1][C - 2]$
- 第 $(R + 2)$ 行: $V[0][0] \dots V[0][C - 1]$
- ...
- 第 $(2R)$ 行: $V[R - 2][0] \dots V[R - 2][C - 1]$
- 下一行: E
- 接下來 E 行: 依照事件發生的順序，每個事件一行

若 $C = 1$ ，不需要空白數行（第 2 行到第 $R + 1$ 行）來表示在水平道路上的袋熊數量。

若要表示事件，則必須是以下其中一種格式：

- 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，上述的例子應該以下列的格式呈現：

```
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
```

語言註釋 (Language Notes)

C/C++ 你必須 `#include "wombats.h"`。

Pascal 你必須定義 `unit Wombats`。所有陣列由 0 開始標示（不是 1）。

請看你機器裡的解答樣版（solution templates）來當作範例。