

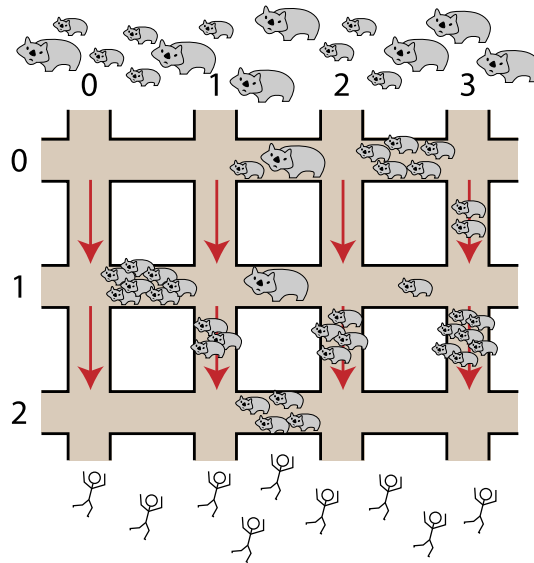


International Olympiad in Informatics
2013
6–13 July 2013
Brisbane, Australia

wombats
English — 1.1

布里斯班被变异的袋熊占领，你必须带领大家去安全的地方。

布里斯班的道路像一个大网格，有 R 条东西向的横向道路，从北向南依次编号为 $0, \dots, (R - 1)$ ，有 C 条南北向的纵向道路，从西向东依次编号为 $0, \dots, (C - 1)$ ，如下图所示。



袋熊从北方入侵，人们逃向南方。人们可以在横向道路上双方向移动，但是在纵向道路上只能往南面安全的地方走。

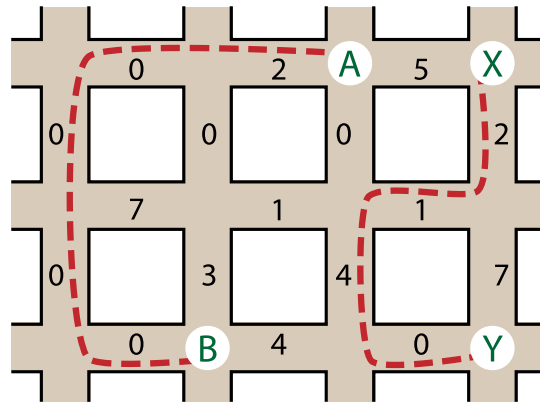
横向道路 P 和纵向道路 Q 的交点表示为 (P, Q) 。相邻2个交点之间的道路线段上有一些袋熊，且数目是随时间变化的。你的任务是引导每个人从最北边（在横向道路 0 上）的指定交点逃到最南端（在横向道路 $R - 1$ 上）的指定交点，路上经过的袋熊最少。

首先会告诉你网格的规模以及每条道路线段上的袋熊的数量。然后给你一系列 E 事件，每个事件是下列两者之一：

- 变化，表示有些道路线段上的袋熊数量发生变化；或者
- 逃离，表示有些人已到达横向道路 0 上指定交点，你必须给他们指出一条路，通往横向道路 $R - 1$ 上指定交点且路上遇到的袋熊最少。

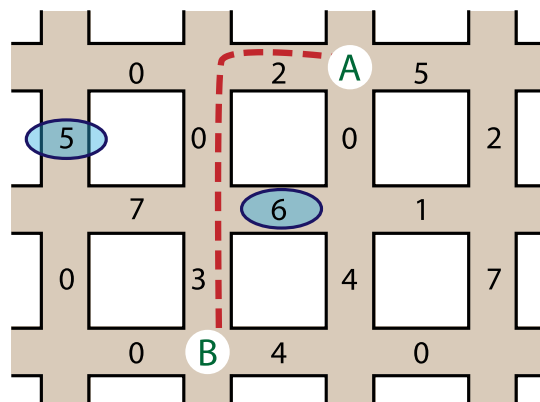
你需要实现下面描述的函数 `init()`，`changeH()`，`changeV()` 和 `escape()` 来处理这些事件。

举例说明



上图所示的初始地图中有3条横向道路（ $R = 3$ ）和4条纵向道路（ $C = 4$ ），每条道路线段上的袋熊数目如线段上的标记所示。考虑下列事件：

- 一个人到达交点 $A = (0, 2)$ ，希望逃到交点 $B = (2, 1)$ 。如图上虚线所示，他最少需要经过 2 只袋熊。
- 又一个人到达交点 $X = (0, 3)$ ，希望逃到交点 $Y = (2, 3)$ 。如图上虚线所示，他最少需要经过 7 只袋熊。
- 发生 2 个变化事件：纵向道路 0 上最上面那条道路线段上的袋熊数目变为 5，横向道路 1 上中间那条道路线段上的袋熊数目变为 6，见下图中圈出来的两个数字。



- 第 3 个人到达交点 $A = (0, 2)$ ，希望逃到交点 $B = (2, 1)$ ，现在他最少需要经过 5 只袋熊，如图中虚线所示。

实现

你需要提交一个文件，实现过程 `init()`，`changeH()` 和 `changeV()`，以及函数 `escape()`，如下描述：

你的过程：**`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

该过程给你初始地图的布局，并允许你初始化任何全局变量和数据结构。该过程只能被调用一次，且要在调用 `changeH()`、`changeV()` 和 `escape()` 之前调用。

参数

- `R`：横向道路的数目。
- `C`：纵向道路的数目。
- `H`：二维数组 $R \times (C - 1)$ ，其中 `H[P][Q]` 表示交点 `(P, Q)` 和交点 `(P, Q + 1)` 之间的横向道路线段上的袋熊数目。
- `V`：二维数组 $(R - 1) \times C$ ，其中 `V[P][Q]` 表示交点 `(P, Q)` 和交点 `(P + 1, Q)` 之间的纵向道路线段上的袋熊数目。

你的过程：**`changeH()`**

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

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

描述

当交点 `(P, Q)` 和交点 `(P, Q + 1)` 之间的横向道路线段上的袋熊数目发生变化时调用此过程。

参数

- `P`：表示变化影响的横向道路的编码 ($0 \leq P \leq R - 1$)。
- `Q`：表示发生变化的道路线段介于哪两条纵向道路之间 ($0 \leq Q \leq C - 2$)。
- `W`：表示变化后的袋熊数目 ($0 \leq W \leq 1,000$)。

你的过程: **changeV()**

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

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

描述

当交点 (P, Q) 和交点 $(P + 1, Q)$ 之间的纵向道路线段上的袋熊数目发生变化时调用此过程。

参数

- P : 表示变化影响的横向道路的编码 ($0 \leq P \leq R - 2$)。
- Q : 表示发生变化的道路线段介于哪两条纵向道路之间 ($0 \leq Q \leq C - 1$)。
- W : 表示变化后的袋熊数目 ($0 \leq W \leq 1,000$)。

你的函数: **escape()**

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

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

描述

该函数计算一个人从交点 $(0, V1)$ 逃到交点 $(R-1, V2)$ 最少需要经过多少只袋熊。

参数

- $v1$: 表示人从横向道路 0 上的交点 ($0 \leq v1 \leq C-1$) 出发。
- $v2$: 表示这个人要逃到横向道路 $R-1$ 上的交点 ($0 \leq v2 \leq C-1$)。
- 返回值: 表示这个人最少经过的袋熊数目。

 样例

下面描述一下题目中的例子。

| 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

- 时间限制：20秒
- 内存限制：256 MiB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- 最多500个变化(调用 `changeH()` 或者 `changeV()`)
- 最多200,000次调用 `escape()`
- 任意时刻一条道路线段上最多1,000只袋熊

子任务

| 子任务 | 分数 | 输入限制 |
|-----|----|--|
| 1 | 9 | $C = 1$ |
| 2 | 12 | $R, C \leq 20$ ，不会调用 <code>changeH()</code> 和 <code>changeV()</code> |
| 3 | 16 | $R, C \leq 100$ ，最多调用 <code>escape()</code> 100次 |
| 4 | 18 | $C = 2$ |
| 5 | 21 | $C \leq 100$ |
| 6 | 24 | 无 |

测试

你电脑上的样例评分程序的输入是文件 `wombats.in`，该文件的格式如下：

- 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: 每行一个事件, 按照事件发生的顺序给出。

如果 $C = 1$, 表示横向道路上每条道路线段上的袋熊数目的若干空行(第 2 到 $R + 1$ 行)将会被省略。

表示每个事件的那一行格式如下:

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

例如: 题目中的例子应该表示为以下格式

```
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
```

编程语言提示

C/C++ 你必须 `#include "wombats.h"`。

Pascal 你必须定义 `unit Wombats`。所有的数组从 0 开始(不是 1)。

参考你电脑上的模板程序。