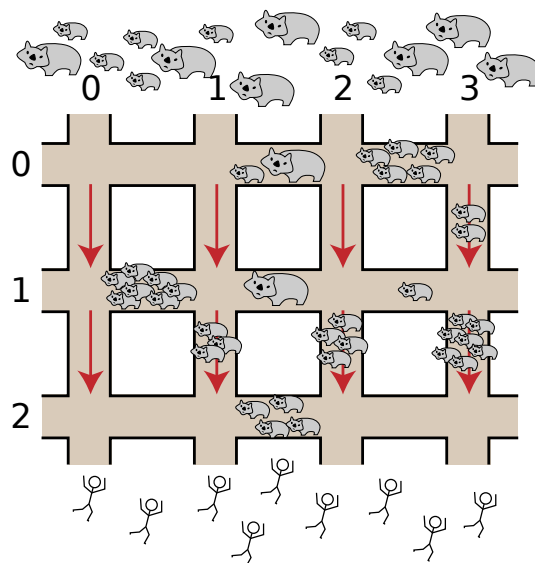


Город Брисбен был захвачен большими мутировавшими вомбатами, и вам требуется вывести людей в безопасное место.

Дороги в Брисбене организованы в виде большой решетки. Она представляет собой R горизонтальных дорог, направленных с востока на запад и пронумерованных от 0 до $R - 1$ с севера на юг. Аналогично, в Брисбене есть C вертикальных дорог, направленных с севера на юг и пронумерованных от 0 до $C - 1$ с запада на восток, как показано на рисунке ниже.



Нашествие вомбатов началось с севера, и люди пытаются спастись на юге. Люди могут бежать по горизонтальным дорогам в любом направлении, однако по вертикальным дорогам они *могут перемещаться только на юг*.

Перекресток горизонтальной дороги с номером P и вертикальной дороги с номером Q будем обозначать как (P, Q) . На каждом участке дороги между двумя перекрестками находится несколько вомбатов, и их количество может изменяться со временем. Ваша задача состоит в том, чтобы для человека, находящегося на каком-то конкретном перекрестке на самой северной дороге (горизонтальной дороге с номером 0), указать такой путь до другого конкретного перекрестка на самой южной дороге (горизонтальной дороге с номером $R - 1$), на котором общее количество вомбатов будет минимально.

Изначально известны размеры дорожной решетки и количество вомбатов на каждом из участков дороги. После этого вам необходимо обработать серию из E событий, где каждое событие представляет собой одно из двух:

- либо изменяется количество вомбатов на каком-либо участке дороги;
- либо на некотором перекрестке на горизонтальной дороге с номером 0 появляется человек, и вы должны найти путь от его местоположения до некоторого перекрестка на горизонтальной дороге с номером $R - 1$, на котором общее число вомбатов минимально.

Для обработки этих запросов вам необходимо реализовать функции `init()`, `changeH()`, `changeV()` и функцию `escape()` как описано ниже.

Примеры

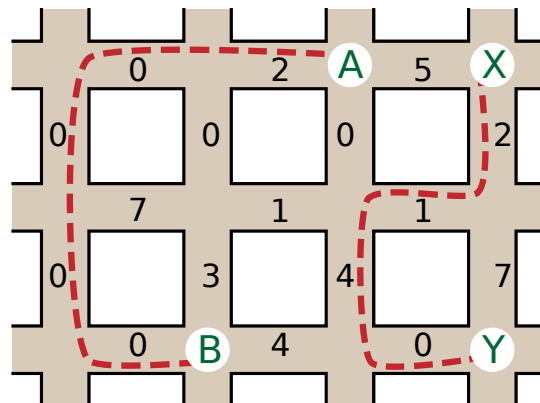
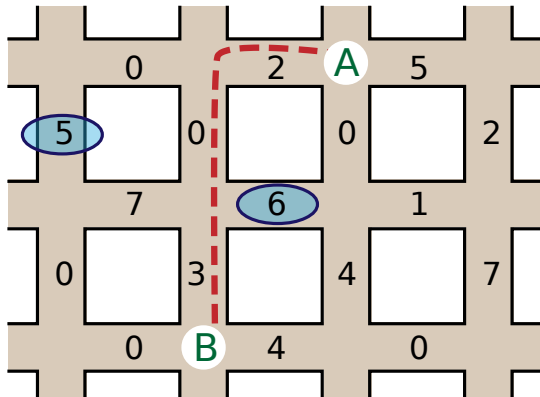


Рисунок выше описывает начальную дорожную решетку, состоящую из $R = 3$ горизонтальных дорог и $C = 4$ вертикальных дорог, для каждого участка дороги указано изначально количество вомбатов. Рассмотрим следующую последовательность событий:

- Человек появляется на перекрестке $A = (0,2)$ и собирается спастись на перекрестке $B = (2,1)$. Наименьшее число вомбатов, которое он встретит, равно 2 . Оптимальный путь показан на рисунке пунктиром.
- Еще один человек появляется на перекрестке $X = (0,3)$ и собирается спастись на перекрестке $Y = (2,3)$. Наименьшее число вомбатов, которое ему придется встретить, равно 7 . Оптимальный путь также обозначен пунктиром.

- Происходят два события-изменения: количество вомбатов на верхнем участке вертикальной дороги с номером 0 изменяется и становится равным 5, и количество вомбатов на среднем участке горизонтальной дороги с номером 1 изменяется и становится равным 6. Обратите внимание на рисунок ниже, на нем измененные числа обведены.



- Третий человек появляется на перекрестке $A = (0,2)$ и собирается спастись на перекрестке $B = (2,1)$. Теперь наименьшее число вомбатов, которое ему придется встретить на своем пути, равно 5. Оптимальный путь обозначен пунктиром.

Детали реализации

Ваше решение должно содержать функции `init()`, `changeH()`, `changeV()` и функцию `escape()`, которые должны быть описаны следующим образом:

Ваша функция: `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);`

Описание

Эта функция задаёт изначальную ситуацию и позволяет инициализировать любые глобальные переменные и структуры данных. Она будет вызвана только один раз перед какими-либо другими вызовами функций `changeH()`, `changeV()` или `escape()`.

Параметры

- R : количество горизонтальных дорог.
- C : количество вертикальных дорог.
- H : двумерный массив размера $R \times (C - 1)$, где $H[P][Q]$ задает количество вомбатов на участке горизонтальной дороги между перекрестками (P, Q) и $(P, Q + 1)$.
- V : двумерный массив размера $(R - 1) \times C$, где $V[P][Q]$ задает количество вомбатов на участке вертикальной дороги между перекрестками (P, Q) и $(P + 1, Q)$.

Ваша функция: `changeH()`

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

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

Описание

Эта функция будет вызываться, когда изменяется число вомбатов на участке горизонтальной дороги между перекрестками (P, Q) и $(P, Q + 1)$.

Параметры

- P : обозначает, на какой именно горизонтальной дороге расположен участок, на котором изменилось количество вомбатов ($0 \leq P \leq R - 1$).
- Q : обозначает, между какими именно двумя вертикальными дорогами находится участок дороги, на котором изменилось число вомбатов ($0 \leq Q \leq C - 2$).
- W : обозначает новое количество вомбатов на измененном участке дороги ($0 \leq W \leq 1\,000$).

Ваша функция: `changeV()`

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

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

Описание

Эта функция будет вызываться, когда изменяется число вомбатов на участке вертикальной дороги между перекрёстками (P, Q) и $(P + 1, Q)$.

Параметры

- P : обозначает, между какими именно двумя горизонтальными дорогами находится участок дороги, на котором изменилось число вомбатов ($0 \leq P \leq R - 2$).
- Q : обозначает, на какой именно вертикальной дороге расположен участок, на котором изменилось количество вомбатов ($0 \leq Q \leq C - 1$).
- W : обозначает новое количество вомбатов на измененном участке дороги ($0 \leq W \leq 1\,000$).

Ваша функция: `escape()`

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

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

Описание

Эта функция должна вычислять минимальное возможное количество вомбатов, которое может встретить человек на пути от перекрёстка $(0, V1)$ до перекрёстка $(R - 1, V2)$.

Параметры

- $V1$: обозначает, где именно человек изначально находится на горизонтальной дороге с номером 0 ($0 \leq V1 \leq C - 1$).
- $V2$: обозначает, где именно человек должен закончить свой путь на горизонтальной дороге с номером $R - 1$ ($0 \leq V2 \leq C - 1$).
- *Возвращаемое значение*: минимальное количество вомбатов, которое человек может встретить на пути от одного перекрёстка до другого.

Пример

Следующий пример соответствует рисунку выше:

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

Ограничения

- Ограничение по времени: 20 секунд.
- Ограничение по памяти: 256 МиБ.
- $2 \leq R \leq 5\,000$.
- $1 \leq C \leq 200$.
- Не более 500 изменений (вызовов функций `changeH()` или `changeV()`).
- Не более 200 000 вызовов функции `escape()`.
- Не более 1000 вомбатов на любом участке дороги в любой момент времени.

Подзадачи

Подзадача	Баллы	Дополнительные ограничения
1	9	$C = 1$
2	12	$R, C \leq 20$, и не будет ни одного вызова функций <code>changeH()</code> или <code>changeV()</code>
3	16	$R, C \leq 100$, и будет не более 100 вызовов функции <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(Нет)

Взаимодействие с проверяющим модулем

Проверяющий модуль на вашем компьютере будет читать входные данные из файла `wombats.in`, который должен иметь следующий формат:

- строка 1: `R C`
- строка 2: `H[0][0] ... H[0][C-2]`
- ...
- строка $(R + 1)$: `H[R-1][0] ... H[R-1][C-2]`
- строка $(R + 2)$: `V[0][0] ... V[0][C-1]`
- ...
- строка $(2R)$: `V[R-2][0] ... V[R-2][C-1]`
- следующая строка: `E`
- следующие `E` строк: каждая строка описывает одно событие. События перечислены в том порядке, в котором они происходят.

Если `C = 1`, то пустые строки, описывающие число вомбатов на горизонтальных дорогах (строки с номерами от 2 до $R + 1$), могут быть пропущены.

Для каждого события должен быть использован один из следующих форматов:

- для описания вызова функции `changeH(P, Q, W) : 1 P Q W`
- для описания вызова функции `changeV(P, Q, W) : 2 P Q W`
- для описания вызова функции `escape(V1, V2) : 3 V1 V2`

В частности, вышеописанный пример должен быть задан в таком формате:

```
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
```

Особенности конкретных языков программирования

- C/C++ Вы должны подключить заголовочный файл с помощью `#include "wombats.h"`.
- Pascal Вы должны написать модуль с заголовком `unit Wombats`. Все массивы нумеруются, начиная с `0` (а не с `1`).

Для примера посмотрите шаблоны решений на вашем компьютере.