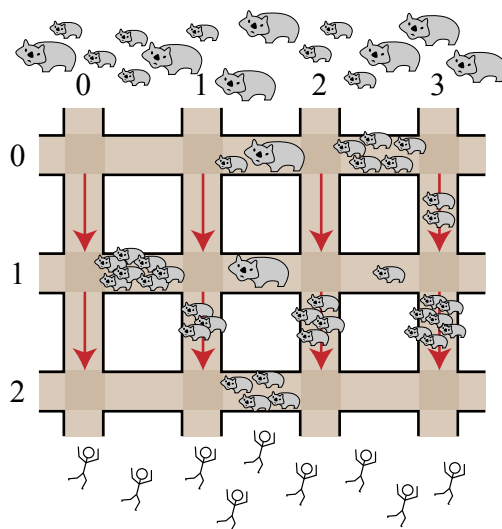


Градот Бризбејн е окупиран од страна на големи измутирани нинџи, и вашата задача е да ги одведете жителите на место каде што ќе бидат безбедни.

Патиштата во Бризбејн се поставени во облик на правоаголна мрежа. Постојат R улици (хоризонтални патишта) по кои што движењето е од исток кон запад, означени со целите броеви $0, \dots, (R - 1)$ во редослед од север кон југ, како и C булевары (вертикални патишта) по кои што движењето е од север кон југ, означени со целите броеви $0, \dots, (C - 1)$ во редослед од запад кон исток, како што е прикажано на сликата подолу.



Нинџите надоаѓаат од север, а луѓето бегаат кон југ. Луѓето можат да се движат по улиците во која било насока, но по булеварите ќе се движат само кон југ т.е. кон место каде што ќе бидат безбедни.

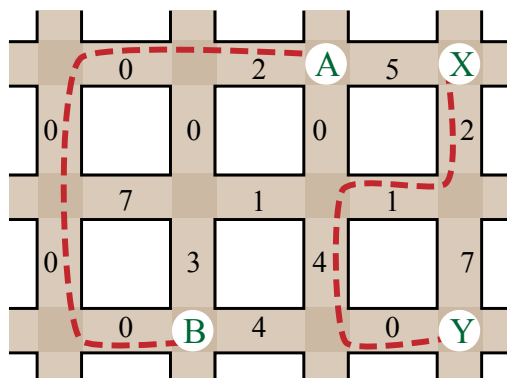
Пресекот на дадена улица P со даден булевар Q е означен со (P, Q) . Секој сегмент од пат што се наоѓа помеѓу два пресека содржи одреден број на нинџи, при што овие броеви може да се менуваат со текот на времето. Ваша задача е да го одведете секој човек од даден пресек кој се наоѓа на улицата 0 до друг пресек кој што се наоѓа на улицата $R - 1$, при што рутата по која ќе го одведете ќе поминува преку колку што е можно помалку нинџи.

На почетокот, ќе ви бидат дадени димензиите на мрежата и бројот на нинџи на секој од патните сегменти. Следно, ќе ви биде дадена секвенца од E настани, каде секој од настаните ќе биде точно еден од следниве:

- *промена (change)*, кој што означува менување на бројот на нинџи на некој патен сегмент;
- *спасување (escape)*, каде што некој човек пристигнува во даден пресек кој се наоѓа на улицата 0, и вие треба да ја најдете рутата до одреден пресек кој се наоѓа на улицата R - 1 која што поминува преку најмал можен број на нинџи.

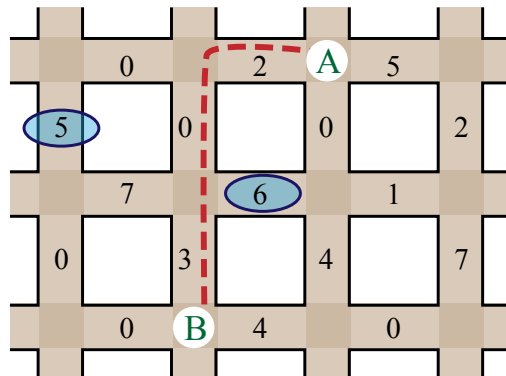
Справувањето со овие настани треба да го направите преку имплементирање на функциите `init()`, `changeH()`, `changeV()` и `escape()`, како што е опишано подолу.

Примери



На сликата погоре прикажана е почетна мапа со R = 3 улици и C = 4 булевары, при што е означен и бројот на нинџи за секој патен сегмент. Да ја разгледаме следнава секвенца од настани:

- Еден човек пристигнува во пресекот A = (0, 2) и сака да се спаси придвижувајќи се до пресекот B = (2, 1). Најмалиот број на нинџи преку кои што може да помине е 2, како што е прикажано со соодветната испрекината линија на сликата.
- Друг човек пристигнува во пресекот X = (0, 3) и сака да се спаси придвижувајќи се до пресекот Y = (2, 3). Најмалиот број на нинџи преку кои што може да помине е 7, како што (повторно) е прикажано со соодветна испрекината линија на сликата.
- Следно, се појавуваат два настани - промени: бројот на нинџи во најгорниот сегмент на булеварот 0 се менува во 5, а бројот на нинџи во средниот сегмент на улицата 1 се менува во 6. Погледнете ги заокружените броеви на сликата подолу.



- Трет човек пристигнува во пресекот $A = (0, 2)$ и сака да се спаси придвижувајќи се до пресекот $B = (2, 1)$. Овојпат, најмалиот број на нинџи преку кои што тој може да помине е `5`, како што е прикажано со новата испрекината линија.

Имплементација

Треба да предадете датотека во која што се имплементирани процедурите `init()`, `changeH()` и `changeV()`, како и функцијата `escape()`, според следниве спецификации:

Ваша Процедура: `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);`

Опис

Оваа процедура ви го дава почетниот изглед на мапата, и ви овозможува да направите иницијализација на глобалните променливи и податочните структури што ќе ги користите. Таа ќе биде повикана само еднаш, пред кои било повици до `changeH()`, `changeV()` или `escape()`.

Параметри

- `R` : Бројот на улици (хоризонтални патишта).
- `C` : Бројот на булевари (вертикални патишта).
- `H` : Дводимензионална низа со големина $R \times (C - 1)$, каде елементот `H[P][Q]` го дава бројот на нинџи на сегментот од улицата помеѓу пресеците `(P, Q)` и `(P, Q + 1)`.
- `V` : Дводимензионална низа со големина $(R - 1) \times C$, каде елементот `V[P][Q]` го дава бројот на нинџи на сегментот од булеварот помеѓу пресеците `(P, Q)` и `(P + 1, Q)`.

Ваша Процедура: changeH()

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

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

Опис

Оваа процедура ќе биде повикана кога ќе се промени бројот на нинџи на сегментот од улицата помеѓу пресеците (P, Q) и $(P, Q + 1)$.

Параметри

- P : Означува на која улица се однесува промената ($0 \leq P \leq R - 1$).
- Q : Означува помеѓу кои два булевари лежи сегментот ($0 \leq Q \leq C - 2$).
- W : Новиот број на нинџи на овој патен сегмент ($0 \leq W \leq 1,000$).

Ваша Процедура: `changeV()`

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

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

Опис

Оваа процедура ќе биде повикана кога ќе се промени бројот на нинџи на сегментот од булеварот помеѓу пресеците `(P, Q)` и `(P + 1, Q)`.

Параметри

- `P` : Означува помеѓу кои две улици лежи сегментот ($0 \leq P \leq R - 2$).
- `Q` : Означува на кој булевар се однесува промената ($0 \leq Q \leq C - 1$).
- `W` : Новиот број на нинџи на овој патен сегмент ($0 \leq W \leq 1,000$).

Ваша Функција: `escape()`

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

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

Опис

Оваа функција треба да го пресметува најмалиот можен број на нинџи преку кои што треба да помине некој човек патувајќи од пресекот `(0, V1)` до пресекот `(R-1, V2)`.

Параметри

- `V1` : Означува од каде тргнува човекот на улицата 0 ($0 \leq V1 \leq C-1$).
- `V2` : Означува каде треба да стигне човекот на улицата `R-1` ($0 \leq V2 \leq C-1$).
- *Го враќа:* Најмалиот број на нинџи преку кои што треба да помине човекот.

Пробна сесија

Следната сесија го објаснува погорниот пример:

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

Ограничувања

- Временско ограничување: 15 секунди
- Мемориско ограничување: 256 MiB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- Најмногу 500 промени (повици до една од процедурите `changeH()` или `changeV()`)
- Најмногу 200,000 повици до `escape()`
- Најмногу 1,000 нинџи на кој било сегмент во кое било време

Подзадачи

Подзадача	Поени	Дополнителни Ограничувања на Влезните Податоци
1	9	$C = 1$
2	12	$R, C \leq 20$, и нема да има повици до <code>changeH()</code> или <code>changeV()</code>
3	16	$R, C \leq 100$, и ќе има најмногу 100 повици до <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(Нема)

Експериментирање

Пробниот оценувач на вашиот компјутер ќе чита влезни податоци од датотеката `wombats.in`, кои мора да бидат во следниот формат:

- линија 1: `R C`
- линија 2: `H[0][0] ... H[0][C-2]`
- ...
- линија $(R + 1)$: `H[R-1][0] ... H[R-1][C-2]`
- линија $(R + 2)$: `V[0][0] ... V[0][C-1]`
- ...
- линија $(2R)$: `V[R-2][0] ... V[R-2][C-1]`
- следната линија: `E`
- следните `E` линии: по еден настан во секоја линија, во редоследот на појавување на настаните

Ако $C = 1$, празните линии што го содржат бројот на нинџи на улиците (линиите од 2 до $R + 1$) не се неопходни.

Линијата за секој од настаните мора да биде во еден од следните формати:

- за да се индицира `changeH(P, Q, W)` : 1 P Q W
- за да се индицира `changeV(P, Q, W)` : 2 P Q W
- за да се индицира `escape(V1, V2)` : 3 V1 V2

На пример, погорниот пример треба да е даден во следниот формат:

```
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
```

Забелешки за јазикот

C/C++ You must `#include "wombats.h"`.

Pascal You must define the `unit Wombats`. All arrays are numbered beginning at `0` (not `1`).

See the solution templates on your machine for examples.