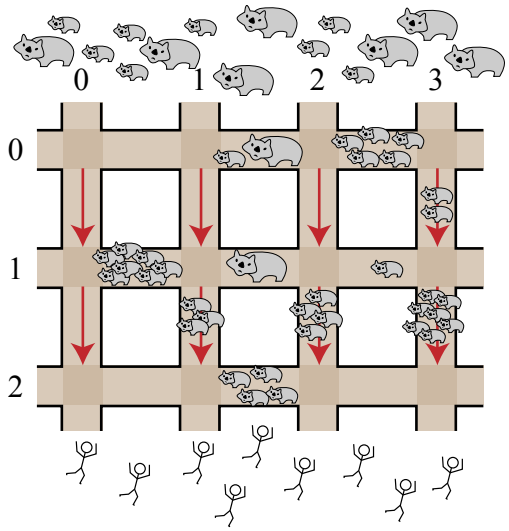
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia

wombats

Norwegian — 1.0

Brisbane har blitt tatt over av store, muterte wombater, og du må lede folket i sikkerhet.

Veiene i Brisbane danner et stort rutenett. Det finnes `R` horisontale veier, som går øst-vest og er nummerert `0, …, (R - 1)` fra nord til sør, og `C` vertikale veier, som går nord-syd og er nummerert `0, …, (C - 1)` fra vest til øst, som vist på bildet nedenfor.



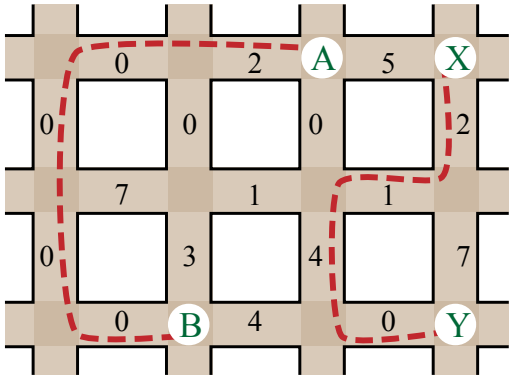
Wombatene har invadert fra nord, og folket flykter mot sør. Folk kan løpe langs de horisontale veiene i begge retninger, men langs vertikale veier vil de *bare løpe sydover* mot sikkerheten.

Veikrysset mellom den horisontale veien `P` og den vertikale veien `Q` betegnes `(P, Q)`. Hvert veisegment mellom to kryss inneholder et antall wombater, og disse antallene kan endre seg over tid. Oppgaven din er å lede hver person fra et gitt veikryss i nord (på horisontal vei nummer `0`) til et gitt veikryss i syd (på horisontal vei `R - 1`) langs en rute som inneholder så få wombater som mulig.

Til å begynne med vil du få oppgitt størrelsen på rutenettet og antallet wombater på hvert veisegment. Etter dette vil du bli gitt en serie med `E` hendelser. Hver hendelse vil en av følgende: - En *endring*, som forandrer antallet wombater på et veisegment - En *flukt*, hvor en person ankommer i et gitt veikryss på horisontal vei nummer `0`, og du må finne en rute til et gitt kryss på horisontal vei nummer `R - 1` som passerer så få wombater som mulig

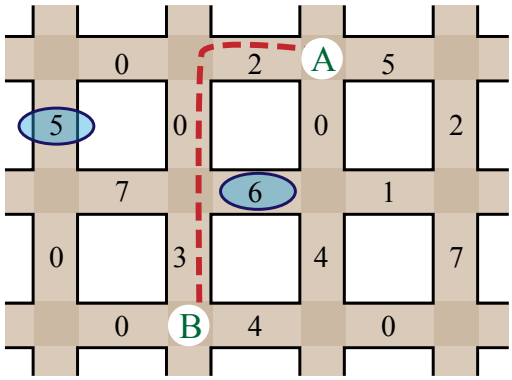
Du må håndtere disse hendelsene ved å implementere funksjonene `init()`, `changeH()`, `changeV()` og `escape()`, som beskrevet nedenfor.

Examples



Bildet ovenfor viser et initielt kart med `R = 3` horisontale veier og `C = 4` vertikale veier, med antall wombater markert på hvert segment. Betrakt den følgende serien av hendelser:

- En person ankomer krysset `A = (0, 2)` og ønsker å flykte til krysset `B = (2, 1)`. Det minste antallet wombater hun kan passere er `2`, slik den stiplede streken indikerer.
- En annen person ankommer krysset `X = (0, 3)` og ønsker å flykte til krysset `Y = (2, 3)`. Det minste antallet wombater hun kan passere er `7`, igjen indikert med en stiplet strek.
- To endringshendelser inntreffer: antallet wombater i toppsegmentet av vertikal vei nummer `0` endres til `5`, og antallet wombater i midtsegmentet av horisontal vei nummer `1` endrer seg til `6`. Se de innsirklede tallene i bildet under.



- En tredje person ankommer krysset `A = (0, 2)` og ønsker å flykte til krysset `B = (2, 1)`. Det minste antallet wombater hun kan passere er nå `5`, som indikert av den nye, stiplede linjen.

Implementation

You should submit a file implementing the procedures `init()`, `changeH()` and `changeV()` and the function `escape()`, as follows:

Your Procedure: `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);
```

Description

This procedure gives you the initial layout of the map, and allows you to initialise any global variables and data structures. It will be called only once, before any calls to `changeH()`, `changeV()` or `escape()`.

Parameters

- `R`: The number of horizontal roads.
- `C`: The number of vertical roads.
- `H`: A two-dimensional array of size `R × (C - 1)`, where `H[P][Q]` gives the number of wombats on the segment of horizontal road between intersections `(P, Q)` and `(P, Q + 1)`.
- `V`: A two-dimensional array of size `(R - 1) × C`, where `V[P][Q]` gives the number of wombats on the segment of vertical road between intersections `(P, Q)` and `(P + 1, Q)`.

Your Procedure: changeH()

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

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

Description

This procedure will be called when the number of wombats changes on the horizontal road segment between intersections (P, Q) and $(P, Q + 1)$.

Parameters

- `P`: Indicates which horizontal road is affected ($0 \le P \le R - 1$).
- `Q`: Indicates between which two vertical roads the segment lies ($0 \le Q \le C - 2$).
- `W`: The new number of wombats on this road segment ($0 \le W \le 1{,}000$).

Your Procedure: `changeV()`

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

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

Description

This procedure will be called when the number of wombats changes on the vertical road segment between intersections `(P, Q)` and `(P + 1, Q)`.

Parameters

- `P`: Indicates between which two horizontal roads the segment lies (`0 ≤ P ≤ R - 2`).
- `Q`: Indicates which vertical road is affected (`0 ≤ Q ≤ C - 1`).
- `W`: The new number of wombats on this road segment (`0 ≤ W ≤ 1,000`).

Your Function: `escape()`

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

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

Description

This function should calculate the fewest possible wombats a person must pass when travelling from intersection `(0, V1)` to `(R-1, V2)`.

Parameters

- `V1`: Indicates where the person begins on horizontal row 0 (`0 ≤ V1 ≤ C-1`).
- `V2`: Indicates where the person ends on horizontal row `R-1` (`0 ≤ V2 ≤ C-1`).
- *Returns*: The smallest number of wombats the person must pass.

Sample Session

The following session describes the example above:

Function Call	Returns
`init(3, 4, [[0,2,5], [7,1,1], [0,4,0]], [[0,0,0,2], [0,3,4,7]])`	
`escape(2,1)`	`2`
`escape(3,3)`	`7`
`changeV(0,0,5)`	
`changeH(1,1,6)`	
`escape(2,1)`	`5`

Constraints

- Time limit: 15 seconds
- Memory limit: 256 MiB
- $2 \le R \le 5{,}000$
- $1 \le C \le 200$
- At most 500 changes (calls to either `changeH()` or `changeV()`)
- At most 200,000 calls to `escape()`
- At most 1,000 wombats on any segment at any time

Subtasks

Subtask	Points	Additional Input Constraints
1	9	`C = 1`
2	12	`R,C ≤ 20`, and there will be no calls to `changeH()` or `changeV()`
3	16	`R,C ≤ 100`, and there will be at most 100 calls to `escape()`
4	18	`C = 2`
5	21	`C ≤ 100`
6	24	*(None)*

Experimentation

The sample grader on your computer will read input from the file `wombats.in`, which must be in the following format:

- line 1: `R C`
- line 2: `H[0][0] … H[0][C-2]`
- …
- line `(R + 1)`: `H[R-1][0] … H[R-1][C-2]`
- line `(R + 2)`: `V[0][0] … V[0][C-1]`
- …
- line `(2R)`: `V[R-2][0] … V[R-2][C-1]`
- next line: `E`
- next `E` lines: one event per line, in the order in which events occur

If `C = 1`, the empty lines containing the number of wombats on horizontal roads (lines `2` through `R + 1`) are not necessary.

The line for each event must be in one of the following formats:

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

For instance, the example above should be provided in the following format:

```
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
```

Language Notes

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.