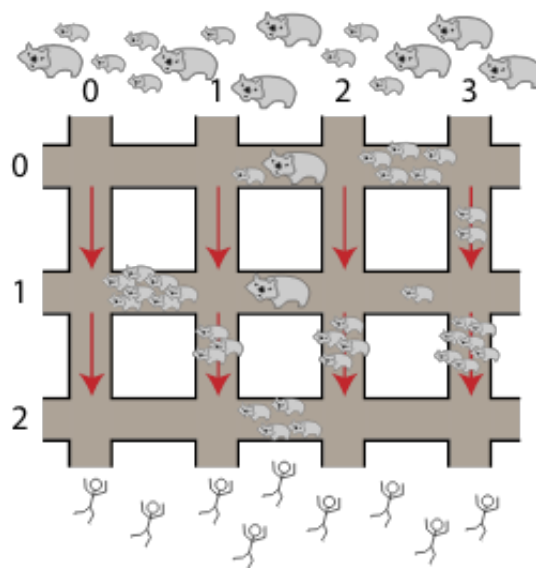


Grad Brisbane okupiran je velikim mutiranim vombatima, i vi trebate pomoći građanima da se spase.

Putevi u Brisbane-u su u obliku velike rešetke. Postoji  $R$  horizontalnih puteva u pravcu zapad-istok, numerisanih  $0, \dots, (R - 1)$  redom od sjevera ka jugu, i  $C$  vertikalnih u pravcu sjever-jug, numerisanih  $0, \dots, (C - 1)$  redom od zapada ka istoku, kao što je prikazano na slici ispod.



Vombati su napali sa sjevera, i ljudi pokušavaju da pobjegnu na jug. Ljudi se mogu kretati horizontalnim putevima u bilo kojem smjeru, ali se vertikalnim mogu kretati *samo ka jugu* (na dole).

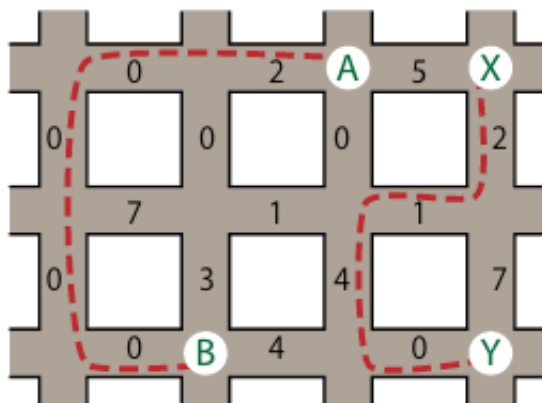
Raskrsnica horizontalnog puta  $P$  i vertikalnog puta  $Q$  obilježena je sa  $(P, Q)$ . Na svakom dijelu puta između dvije raskrsnice se nalazi određeni broj vombata, i ti brojevi se mogu mijenjati u vremenu. Vaš zadatak je da pomognete svakoj osobi da dođe od neke date raskrsnice koja se nalazi skroz sjeverno (na horizontalnom putu  $0$ ) do neke date raskrsnice koja je skroz južno (na horizontalnom putu  $R - 1$ ), tako što će ići rutom koja prolazi što je moguće manje vombata.

Na početku će Vam biti data veličina rešetke i broj vombata na svakom dijelu puta. Nakon toga, biće vam redom dato  $E$  događaja, od kojih je svaki jedne od sljedeće dvije vrste:

- promjena(*change*), označava da se broj vombata promijenio na nekom segmentu; ili
- bijeg(*escape*), označava da neka osoba dolazi na datu raskrsnicu na horizontalnom putu  $0$ , a vi trebate pronaći rutu do date raskrsnice na horizontalnom putu  $R - 1$  koja prolazi najmanji mogući broj vombata.

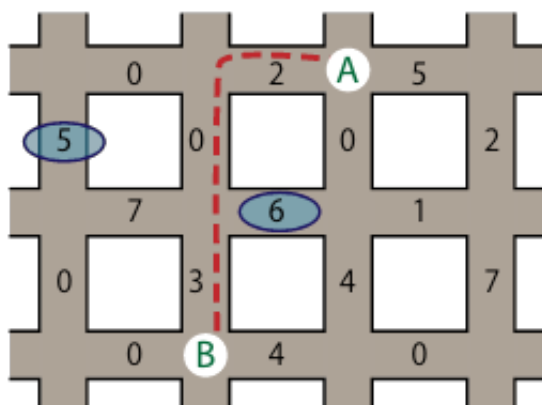
Trebate obraditi ove događaje implementirajući funkcije `init()`, `changeH()`, `changeV()` i `escape()`, kao što je opisano ispod.

## Primjeri



Slika iznad predstavlja početnu mapu gdje je  $R = 3$  broj horizontalnih puteva i  $C = 4$  broj vertikalnih puteva, a broj vombata je obilježen na svakom dijelu puta. Posmatrajmo sljedeći slijed događaja:

- Osoba dolazi na raskrnicu  $A = (0, 2)$  i želi da pobjegne do raskrsnice  $B = (2, 1)$ . Najmanji mogući broj vombata koje prilikom toga može da prođe je  $2$ . Ovaj put je obilježen isprekidanom linijom.
- Druga osoba dolazi na raskrnicu  $X = (0, 3)$  i želi da pobjegne na raskrnicu  $Y = (2, 3)$ . Najmanji mogući broj vombata koje prilikom toga može da prođe je  $7$ . Ovaj put je obilježen isprekidanom linijom.
- Dvije izmjene (događaj *change*) se dešavaju: broj vombata na najvišem (najsjevernijem) segmentu vertikalnog puta  $0$  se mijenja na  $5$ , i broj vombata na srednjem segmentu horizontalnog puta  $1$  se mijenja na  $6$ . Pogledajte zaokružene brojeve na slici ispod.



- Treća osoba dolazi na raskrnicu  $A = (0, 2)$  i želi da pobjegne do raskrsnice  $B = (2, 1)$ . Sada je najmanji mogući broj vombata koji prilikom toga može da prođe je  $7$ , i ovaj put je obilježen isprekidanom linijom.

---

## Implementacija

Treba da submitujete file sa implementacijom procedura `init()`, `changeH()` i `changeV()` i funkcije `escape()`, koje treba da se ponašaju kako slijedi:

### Vaša procedura: `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);`

### Opis

Ova procedura Vam daje početni izgled mape i dozvoljava da inicijalizirate globalne varijable strukture podataka. Ova procedura se poziva samo jednom, prije bilo kojeg poziva procedura `changeH()`, `changeV()` ili funkcije `escape()`.

### Parametri

- `R`: Broj horizontalnih puteva.
- `C`: Broj vertikalnih puteva.
- `H`: Dvodimenzionalni niz dimenzija  $R \times (C - 1)$ , pri čemu je `H[P][Q]` broj vombata na dijelu horizontalnog puta između raskrsnica `(P, Q)` i `(P, Q + 1)`.
- `V`: Dvodimenzionalni niz dimenzija  $(R - 1) \times C$ , pri čemu je `V[P][Q]` broj vombata na dijelu vertikalnog puta između raskrsnica `(P, Q)` i `(P + 1, Q)`.

### Vaša procedura: `changeH()`

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

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

### Opis

Ova procedura se poziva pri promjeni broja vombata na dijelu horizontalnog puta između raskrsnica `(P, Q)` i `(P, Q + 1)`.

### Parametri

- `P`: Redni broj horizontalnog puta na kome se mijenja broj vombata ( $0 \leq P \leq R - 1$ ).

- $Q$  : Označava između koja dva vertikalna puta dolazi do promjene ( $0 \leq Q \leq C - 2$ ).
- $W$  : Novi broj vombata na tom dijelu puta ( $0 \leq W \leq 1,000$ ).

### Vaša procedura: `changeV()`

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

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

#### Opis

Ova procedura se poziva pri promjeni broja vombata na dijelu vertikalnog puta između raskrsnica  $(P, Q)$  i  $(P + 1, Q)$ .

#### Parametri

- $P$  : Označava između kojih vertikalnih puteva dolazi do promjene ( $0 \leq P \leq R - 2$ ).
- $Q$  : Redni broj vertikalnog puta na kome se mijenja broj vombata ( $0 \leq Q \leq C - 1$ ).
- $W$  : Novi broj vombata na tom dijelu puta ( $0 \leq W \leq 1,000$ ).

### Vaša funkcija: `escape()`

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

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

#### Opis

Ova funkcija treba da odredi najmanji mogući broj vombata koje osoba mora sresti ako putuje od raskrsnice  $(0, V1)$  do raskrsnice  $(R-1, V2)$ .

#### Parametri

- $V1$  : Označava raskrsnicu na horizontalnom putu 0 od koje osoba kreće ( $0 \leq V1 \leq C - 1$ ).
- $V2$  : Označava raskrsnicu na horizontalnom putu  $R-1$  ( $0 \leq V2 \leq C-1$ ) na koju osoba želi da stigne.
- *Vraća*: Najmanji broj vombata koje osoba mora sresti na tom putu.

---

## Primjer izvršavanja

Sljedeći primjer izvršavanja odnosi se na ranije navedeni primjer:

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

## Ograničenja

- Vremensko ograničenje: 20 sekundi
- Memorijsko ograničenje: 256 MB
- $2 \leq R \leq 5,000$
- $1 \leq C \leq 200$
- Najviše 500 promjena (poziva procedura `changeH()` ili `changeV()`)
- Najviše 200,000 poziva funkcije `escape()`
- Najviše 1,000 vombata na bilo kom dijelu u bilo kom trenutku.

## Podproblemi

Podproblem	Bodovi	Dodatna ograničenja vezana za ulaz
1	9	$C = 1$
2	12	$R, C \leq 20$ , i nema poziva procedura <code>changeH()</code> ili <code>changeV()</code>
3	16	$R, C \leq 100$ , i ima najviše 100 poziva funkcije <code>escape()</code>
4	18	$C = 2$
5	21	$C \leq 100$
6	24	(Nema)

## Testiranje

Grader na Vašem računaru treba da pročita ulazne podatke iz file-a `wombats.in`. File ima sljedeći format:

- linija 1: `R C`
- linija 2: `H[0][0] ... H[0][C-2]`
- ...
- linija `(R + 1)`: `H[R-1][0] ... H[R-1][C-2]`
- linija `(R + 2)`: `V[0][0] ... V[0][C-1]`
- ...
- linija `(2R)`: `V[R-2][0] ... V[R-2][C-1]`
- sljedeća linija: `E`
- sljedećih `E` linija: jedan događaj u svakom redu u redoslijedu u kojem se dešavaju

Ako je `C = 1`, prazni redovi koji sadrže broj vombata na horizontalnim putevima (redovi od `2` do `R + 1`) nisu neophodni.

Svaki red za događaj ima jedan od sljedećih formata:

- Da biste opisali poziv procedure `changeH(P, Q, W)`: `1 P Q W`;
- Da biste opisali poziv procedure `changeV(P, Q, W)`: `2 P Q W`;
- Da bi opisali poziv funkcije `escape(V1, V2)`: `3 V1 V2`.

Ilustracije radi, ranije opisani primjer je zapisan u sljedećem formatu:

```

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

```

---

## Napomene za programske jezike

C/C++ Treba uraditi `#include "wombats.h"`.

Pascal Trebate definisati `unit Wombats`. Svi nizovi su numerisani sa početkom u `0` (ne `1`).

Pogledati primjer rješenja na Vašem računaru za predložene primjere.