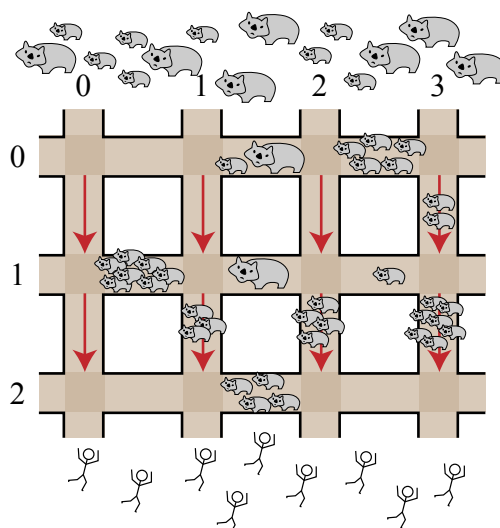


Grad Brizbejn je okupiran velikim mutiranim vombatima, pa vi morate pomoći građanima da se spasu.

Putevi u Brizbejnu su oblika velike rešetke. Postoji R horizontalnih puteva koji idu od zapada ka istoku, numerisani $0, \dots, (R - 1)$ redom od sjevera ka jugu, i C vertikalnih puteva koji idu od sjevera ka jugu, numerisani $0, \dots, (C - 1)$ 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 kom smjeru, ali vertikalnim se mogu kretati *samo ka jugu (na dolje, u sigurnom pravcu)*.

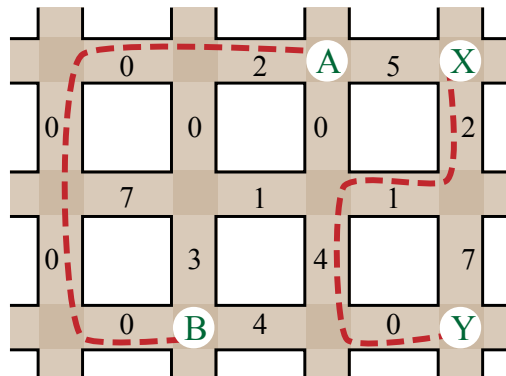
Raskrsnica horizontalnog puta P i vertikalnog puta Q je obilježena sa (P, Q) . Na svakom segmentu puta između dvije raskrsnice se nalazi neki broj vombata, i ti brojevi se mogu mijenjati vremenom. Vaš zadatak je da pomognete svakoj osobi da dođe od neke date raskrsnice koja se nalazi skroz sjeverno (na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

Na početku će vam biti data veličina rešetke i broj vombata na svakom segmentu puta. Nakon toga, biće vam dati redom E događaja, od kojih je svaki jedan od ove dvije vrste:

- *change*, označava da se broj vombata promijenio na nekom segmentu; ili
- *escape*, označava da neka osoba dolazi na datu raskrnicu na horizontalnom putu `0`, a vi morate pronaći rutu do date raskrsnice na horizontalnom putu `R - 1` koja prolazi najmanji mogući broj vombata.

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

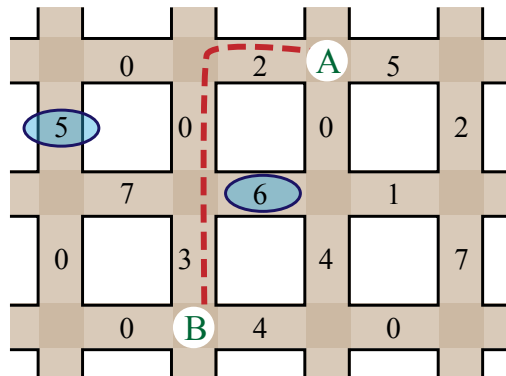
Primjer



Slika iznad pokazuje početnu mapu gdje je `R = 3` broj horizontalnih puteva i `C = 4` broj vertikalnih puteva, a broj vombata je obilježen na svakom segmentu.

Posmatrajmo sljedeći redosljed događaja:

- Osoba dolazi na raskrnicu `A = (0, 2)` i želi da pobjegne na raskrnicu `B = (2, 1)`. Najmanji mogući broj vombata koje usput može da prođe je `2`, što je obilježeno isprekidanom linijom.
- Druga osoba dolazi na raskrnicu `X = (0, 3)` i želi da pobjegne na raskrnicu `Y = (2, 3)`. Najmanji mogući broj vombata koje usput može da prođe je `7`, što je obilježeno 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 raskrsnicu $A = (0, 2)$ i želi da pobjegne na raskrsnicu $B = (2, 1)$. Sada je najmanji mogući broj vombata koji može usput da prođe 5, kao što je obilježeno isprekidanom linijom.

Implementacija

Potrebno je da snimate datoteku sa implementacijama funkcija `init()`, `changeH()` i `changeV()` i funkcije `escape()`, koje treba da se ponašaju kao što je opisano u nastavku:

Funkcija: `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);`

Napomena

Ova procedura Vam daje inicijalni izgled mape i dozvoljava da inicijalizujete globalne promjenljive i strukture podataka. Ova procedura se poziva samo jednom, prije bilo kog pozivanja procedura `changeH()`, `changeV()` ili funkcije `escape()`.

Parametri

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

Funkcija: `changeH()`

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

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

Napomena

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 ≤ P ≤ R - 1`).
- `Q` : govori između koja dva vertikalna puta dolazi do promjene (`0 ≤ Q ≤ C - 2`).
- `W` : Novi broj vobmata na tom dijelu puta (`0 ≤ W ≤ 1,000`).

Your Procedure: changeV()

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

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

Napomena

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

Parametri

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

Funkcija: escape()

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

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

Napomena

Ova funkcija određuje minimalni mogući broj vombata koji neka osoba mora sresti ako putuje od raskrsnice $(0, V1)$ do raskrsnice $(R-1, V2)$.

Parametri

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

Primjer

Sljedeći niz poziva opisuje prethodno opisani 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

- Vrijeme: 20 sekundi
- Memorija: 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 wombatova na bilo kom dijelu u bilo kom trenutku.

Podzadaci

Podzadatak	Poeni	Dodatna ograničenja vezeana 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

Program učitava ulazne podatke iz datoteke `wombats.in`. Datoteka ima sljedeći format:

- red 1: `R C`
- red 2: `H[0][0] ... H[0][C-2]`
- ...
- red `(R + 1)`: `H[R-1][0] ... H[R-1][C-2]`
- red `(R + 2)`: `V[0][0] ... V[0][C-1]`
- ...
- red `(2R)`: `V[R-2][0] ... V[R-2][C-1]`
- sljedeći red: `E`
- sljedećih `E` redova: jedan događaj u svakom redu, u redosljedu u kome se dešavaju

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

Svaki red za događaj ima jedan od sljedećih formata:

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

Na primjer, za prethodno opisani zadatak ulazna datoteka je 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 vezane za jezik implementacije

C/C++ Morate `#include "wombats.h"`.

Pascal Morate definisati `unit Wombats`. Indeksiranje nizova je od `0` (ne od `1`).

Pogledajte templejte sa rješenjima na vašem računaru kao dopunsku primjere.