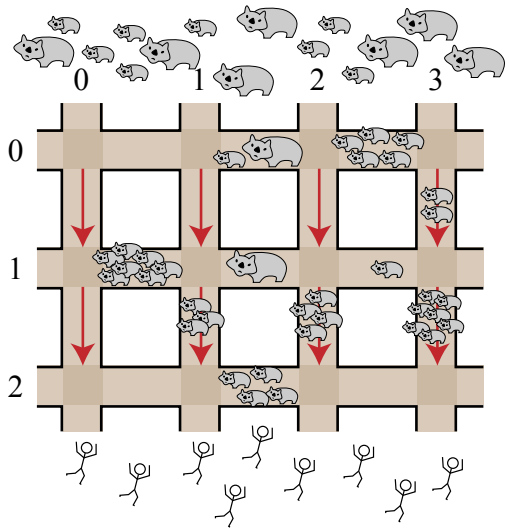
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia

wombats

Polish — 1.1

W Brisbane grasują wielkie zmutowane wombaty. Pomóż ocalić ludzi od tej plagi torbaczy.

Drogi w Brisbane tworzą regularną kratkę. Jest R poziomych dróg biegnących ze wschodu na zachód, ponumerowanych liczbami $0, \ldots, (R - 1)$ w kolejności z północy na południe, i C pionowych dróg biegnących z północy na południe, ponumerowanych liczbami $0, \ldots, (C - 1)$ w kolejności z zachodu na wschód (patrz rysunek poniżej).



Wombaty przybywają do miasta z północy, dlatego ludzie próbują uciekać na południe. Podczas ucieczki mogą oni biec ulicami poziomymi w dowolnym kierunku, jednak wzdłuż ulic pionowych biegną *zawsze w kierunku południowym*, bo tam czeka ocalenie.

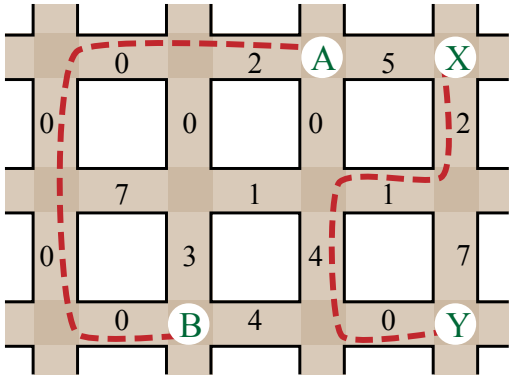
Przecięcie poziomej drogi o numerze P z drogą pionową o numerze Q będziemy oznaczać przez (P, Q). Na każdym fragmencie drogi położonym między dwoma sąsiednimi skrzyżowaniami znajduje się pewna liczba wombatów. Liczba ta może dodatkowo zmieniać się w czasie. Twoim zadaniem jest pomóc zaprowadzić każdą osobę z określonego skrzyżowania położonego na północy (tj. na poziomej drodze o numerze 0) na określone przez nią skrzyżowanie położone na południu (tj. na poziomej drodze o numerze $R - 1$), tak aby na swojej drodze spotkała jak najmniej wombatów.

Znasz rozmiary mapy oraz liczby wombatów znajdujących się początkowo na każdym fragmencie drogi. Ponadto dany jest opis E zdarzeń dwóch typów:

- *zdarzenie zmiany*, które odpowiada zmianie liczby wombatów na danym fragmencie drogi
- *zdarzenie ucieczki*, w którym dana osoba znajduje się przy skrzyżowaniu położonym na drodze poziomej o numerze `0`, a Twoim zadaniem jest znaleźć drogę do danego skrzyżowania położonego na drodze poziomej o numerze `R - 1`, na której osoba ta spotka najmniejszą możliwą liczbę wombatów.

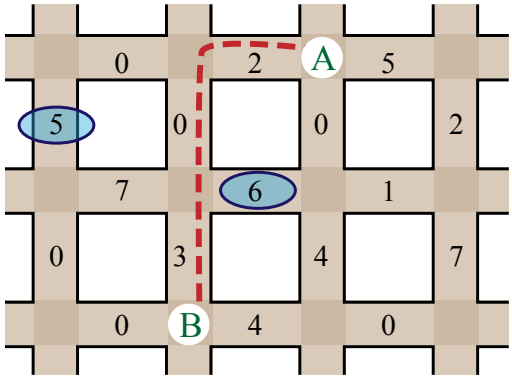
Twoim zadaniem jest obsługa podanych typów zdarzeń. Powinieneś zaimplementować funkcje `init()`, `changeH()`, `changeV()` i `escape()`, opisane poniżej.

Przykłady



Rysunek powyżej przedstawia mapę złożoną z `R = 3` dróg poziomych i `C = 4` dróg pionowych. Na każdym fragmencie drogi znajduje się pewna liczba wombatów. Rozważmy następującą sekwencję zdarzeń:

- Osoba znajduje się przy skrzyżowaniu `A = (0, 2)` i chce uciec do skrzyżowania `B = (2, 1)`. Najmniejsza liczba wombatów, jakie może napotkać w trakcie ucieczki, to `2`. Odpowiednią trasę ucieczki zaznaczono na rysunku linią przerywaną.
- Inna osoba znajduje się przy skrzyżowaniu `X = (0, 3)` i chce uciec do skrzyżowania `Y = (2, 3)`. Najmniejsza liczba wombatów, jakie może napotkać w trakcie ucieczki, to `7`. Odpowiednia trasa ucieczki jest oznaczona na rysunku.
- Dalej mają miejsce dwa zdarzenia zmiany: liczba wombatów przy górnym fragmencie pionowej drogi o numerze `0` zmienia się na `5`, a liczba wombatów na środkowym fragmencie poziomej drogi o numerze `1` zmienia się na `6`. Pozycje te zostały zakreślone na poniższym rysunku.



- Trzecia osoba znajduje się przy skrzyżowaniu `A = (0, 2)` i chce uciec do skrzyżowania `B = (2, 1)`. Teraz najmniejsza liczba napotkanych wombatów to `5`, patrz linia przerywana powyżej.

Implementacja

Powinieneś zgłosić plik z implementacją procedur `init()`, `changeH()` i `changeV()` oraz funkcji `escape()`:

Twoja 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

Procedura przekazuje Ci początkowy wygląd mapy. W tej procedurze możesz zainicjować wszystkie swoje zmienne globalne i struktury danych. Zostanie ona wywołana tylko raz, przed wszystkimi wywołaniami funkcji `changeH()`, `changeV()` i `escape()`.

Parametry

- `R`: liczba dróg poziomych.
- `C`: liczba dróg pionowych.
- `H`: dwuwymiarowa tablica rozmiaru `R × (C - 1)`, przy czym `H[P][Q]` oznacza liczbę wombatów na fragmencie drogi poziomej ograniczonym skrzyżowaniami `(P, Q)` i `(P, Q + 1)`.
- `V`: dwuwymiarowa tablica rozmiaru `(R - 1) × C`, przy czym `V[P][Q]` oznacza liczbę wombatów na fragmencie drogi pionowej ograniczonym skrzyżowaniami `(P, Q)` i `(P + 1, Q)`.

Twoja procedura: `changeH()`

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

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

Opis

Wywołanie tej procedury oznacza zmianę liczby wombatów znajdujących na fragmencie drogi poziomej ograniczonym skrzyżowaniami (P, Q) i $(P, Q + 1)$.

Parametry

- `P`: oznacza numer drogi poziomej ($0 \le P \le R - 1$).
- `Q`: opisuje parę skrzyżowań wyznaczających fragment drogi poziomej ($0 \le Q \le C - 2$).
- `W`: podaje nową liczbę wombatów na rozważanym fragmencie drogi ($0 \le W \le 1{,}000$).

Twoja procedura: `changeV()`

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

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

Opis

Wywołanie tej procedury oznacza zmianę liczby wombatów znajdujących na fragmencie drogi pionowej ograniczonym skrzyżowaniami `(P, Q)` i `(P + 1, Q)`.

Parametry

- `P`: opisuje parę skrzyżowań wyznaczających fragment drogi pionowej (`0 ≤ P ≤ R - 2`).
- `Q`: oznacza numer drogi pionowej (`0 ≤ Q ≤ C - 1`).
- `W`: podaje nową liczbę wombatów na rozważanym fragmencie drogi (`0 ≤ W ≤ 1,000`).

Twoja funkcja: `escape()`

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

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

Opis

Ta funkcja ma wyznaczyć najmniejszą możliwą liczbę wombatów, jakie musi napotkać osoba w trakcie ucieczki ze skrzyżowania `(0, V1)` do skrzyżowania `(R-1, V2)`.

Parametry

- `V1`: wskazuje położenie skrzyżowania początkowego na drodze poziomej numer 0 (`0 ≤ V1 ≤ C-1`).
- `V2`: wskazuje położenie skrzyżowania końcowego na drodze poziomej numer `R-1` (`0 ≤ V2 ≤ C-1`).
- *Wynik funkcji*: najmniejsza liczba wombatów na trasie ucieczki.

Przykład

Oto przykładowe parametry funkcji oraz prawidłowy wynik. Sorry że tabelka po angielsku, nie dało się przetłumaczyć:

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`

Ograniczenia

- Maksymalny czas działania: 20 sekund
- Limit pamięci: 256 MiB
- $2 \le R \le 5\,000$
- $1 \le C \le 200$
- Liczba zdarzeń zmiany (tj. wywołań funkcji `changeH()` i `changeV()`) nie przekracza 500
- Liczba wywołań funkcji `escape()` nie przekracza 200 000
- Na każdym fragmencie drogi będzie co najwyżej 1 000 wombatów

Podzadania

Podzadanie	Punkty	Dodatkowe ograniczenia
1	9	$C = 1$
2	12	$R,C \le 20$ i brak wywołań funkcji `changeH()` i `changeV()`
3	16	$R,C \le 100$; ponadto będzie co najwyżej 100 wywołań funkcji `escape()`
4	18	$C = 2$
5	21	$C \le 100$
6	24	*(brak)*

Uruchamianie lokalne

Przykładowy moduł oceniający na Twoim komputerze czyta wejście z pliku `wombats.in` w następującym formacie:

- wiersz 1: `R C`
- wiersz 2: `H[0][0] … H[0][C-2]`
- …
- wiersz $(R + 1)$: `H[R-1][0] … H[R-1][C-2]`
- wiersz $(R + 2)$: `V[0][0] … V[0][C-1]`
- …
- wiersz $(2R)$: `V[R-2][0] … V[R-2][C-1]`
- kolejny wiersz: `E`
- kolejne `E` wierszy: opis zdarzeń w kolejności następowania, po jednym zdarzeniu w wierszu

Jeśli $C = 1$, nie jest konieczne wypisanie wierszy opisujących liczby wombatów na drogach poziomych (tj. wierszy o numerach od 2 do $R + 1$).

Wiersz opisujący zdarzenie musi być w następującym formacie:

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

Dane z powyższego przykładu powinny być więc podane w następującym formacie:

```
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
```

Uwagi natury językowej

C/C++	Użyj dyrektywy `#include "wombats.h"`.
Pascal	Musisz zdefiniować `unit Wombats`. Wszystkie tablice są indeksowane od `0` (a nie od `1`).

Przykłady znajdziesz w przykładowych rozwiązaniach na Twoim komputerze.