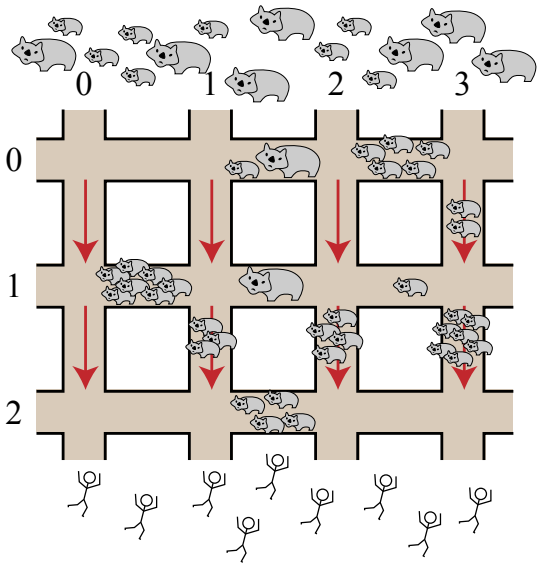
International Olympiad in Informatics 2013
6-13 July 2013
Brisbane, Australia

wombats

Korean — 1.1

브리즈번 시는 돌연변이로 거대해진 웜뱃(호주에 사는 너구리 비슷한 동물)에 장악당했다. 당신의 임무는 사람들을 구조하는 것이다.

브리즈번 시의 도로는 커다란 격자 모양으로 되어 있다. 동서 방향으로 놓인 수평 도로는 R개가 있고, 북쪽에서 남쪽 방향으로 차례로 번호가 $0, \cdots, (R-1)$로 매겨져 있다. 또한, 남북 방향으로 놓인 수직 도로는 C개가 있고, 서쪽에서 동쪽 방향으로 차례로 번호가 $0, \cdots, (C-1)$로 매겨져 있다. 다음 그림은 이렇게 번호가 매겨진 도로의 예이다.



웜뱃은 북쪽에서부터 쳐들어오고 있고, 사람들은 남쪽으로 도망친다. 사람들은 가로 방향으로는 어느 쪽이든 움직일 수 있지만, 세로 방향으로는 안전한 방향인 남쪽으로만 움직일 수 있다.

수평 도로 P와 수직 도로 Q의 교차로는 (P, Q)로 표현한다. 두 교차로 사이 세그먼트에는 웜뱃들이 있을 수 있고, 몇 마리의 웜뱃이 있는지는 시간이 흐르면서 변할 수 있다. 여러분의 임무는 북쪽(수평 도로 0번)의 주어진 교차로에 도착한 사람을 남쪽(수평 도로 $R-1$번)의 주어진 교차로로 가는 길을 알려주는 것인데, 이 경로를 지날 때 가능한 한 적은 수의 웜뱃을 만나야 한다.

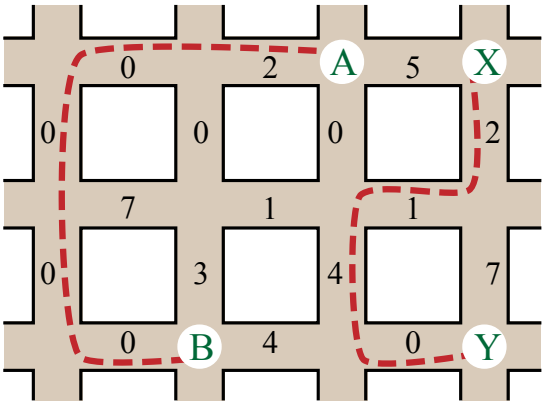
먼저, 격자의 크기와 각 도로 세그먼트에 웜뱃이 몇 마리가 있는지가 주어진다. 다음에는 E개의 이벤트가 차례로 주어지는데, 각 이벤트는 아래 두 가지 중 하나의 형태이다.

- change : 어떤 도로 세그먼트에 있는 웜뱃의 마릿수가 바뀐다.

- escape : 사람 한 명이 수평 도로 `0`번의 주어진 교차로로 도착하고, 이 사람을 가장 적은 수의 웜뱃을 만나면서 수평 도로 `R-1`의 주어진 교차로로 보내는 경로를 구해야 한다.

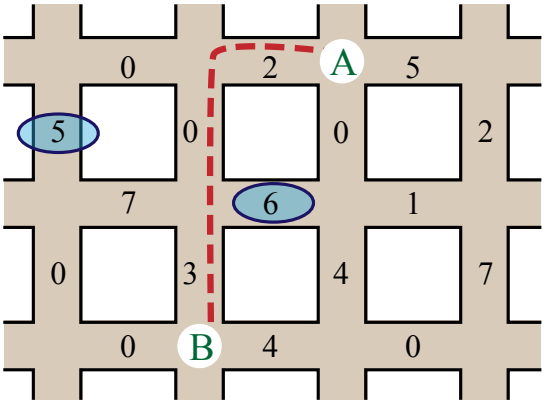
이러한 이벤트는 다음과 같이 정의되는 함수 `init()`, `changeH()`, `changeV()`, `escape()`을 사용하여 처리해야 한다.

예시



위 그림은 수평 도로가 `R = 3`개, 수직 도로가 `C = 4` 개, 그리고 각 도로 세그먼트에 웜뱃이 몇 마리가 있는지 주어진 지도의 처음 상황이다. 다음 순서대로 이벤트가 발생한다고 하자.

- 한 사람이 교차로 `A = (0, 2)`에 도착해서 교차로 `B = (2, 1)`로 도망치고 싶어한다. 이 때 만나는 웜뱃 마릿수의 최소값은 `2`인데, 점선을 따라가면 이를 확인할 수 있다.
- 또 한 사람이 교차로 `X = (0, 3)`에 도착해서 교차로 `Y = (2, 3)`로 도망치고 싶어한다. 이 때 만나는 웜뱃 마릿수의 최소값은 `7`인데, 또 점선을 따라가면 이를 확인할 수 있다.
- change 이벤트가 두 번 발생한다. 수직 도로 `0`의 가장 위 세그먼트에 있는 웜뱃의 수가 `5`마리로 바뀌고, 수평 도로 `1`의 가운데 세그먼트에 있는 웜뱃의 수가 `6`마리로 바뀐다. 아래 그림의 동그라미를 친 숫자를 확인해보자.



- 세 번째 사람이 교차로 `A = (0, 2)`에 도착해서 교차로 `B = (2, 1)`로 도망치고 싶어한다. 이번에는 만나게 되는 웜뱃의 최소 숫자가 `5`마리이고, 다시 점선을 따라가면 이를 확인할 수 있다.

구현

다음 조건을 만족하는 함수 `init()`, `changeH()`, `changeV()`, `escape()`을 구현한 파일을 제출하시오.

구현해야 하는 함수: **`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);
```

설명

도로의 초기 상태가 이 함수를 통해 전달되며, 당신이 필요로 하는 전역변수 및 자료구조를 이 함수 내에서 초기화할 수 있다. 이 함수는 단 한 번만 호출되며, `changeH()`, `changeV()`, 또는 `escape()`가 입력되기 전에 호출될 것이다.

파라미터

- `R`: 수평 도로들의 개수
- `C`: 수직 도로들의 개수
- `H`: 크기가 `R × (C − 1)`인 이차원 배열, 여기서 `H[P][Q]`는 교차로 `(P, Q)`와 `(P, Q + 1)` 사이의 수평 도로 세그먼트상의 웜뱃의 마릿수를 나타낸다.
- `V`: 크기가 `(R − 1) × C`인 이차원 배열, 여기서 `V[P][Q]`는 교차로 `(P, Q)`와 `(P + 1, Q)` 사이의 수직 도로 세그먼트상의 웜뱃의 마릿수를 나타낸다.

구현해야 하는 함수: **changeH()**

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

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

설명

이 함수는 교차로 (P, Q) 와 $(P, Q+1)$ 사이의 수평 도로 세그먼트 위의 웜뱃의 마릿수를 바꿀 때 호출되어진다.

파라미터

- P: 영향을 받는 수평 도로를 나타낸다 ($0 \le P \le R-1$).
- Q: 세그먼트가 어떤 두 수직 도로 사이에 놓여있는지를 나타낸다 ($0 \le Q \le C-2$).
- W: 이 도로 세그먼트 위의 새로운 웜뱃의 마릿수를 나타낸다 ($0 \le W \le 1,000$).

구현해야 하는 함수: **changeV()**

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

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

설명

이 함수는 교차로 (P, Q) 와 $(P+1, Q)$ 사이의 수직 도로 세그먼트 위의 웜뱃의 마릿수를 바꿀 때 호출된다.

파라미터

- P: 세그먼트가 어떤 두 수평 도로 사이에 놓여있는지를 나타낸다 ($0 \le P \le R-2$).
- Q: 영향을 받는 수직 도로를 나타낸다 ($0 \le Q \le C-1$).
- W: 이 도로 세그먼트 위의 새로운 웜뱃의 마릿수를 나타낸다 ($0 \le W \le 1,000$).

구현해야 하는 함수: **escape()**

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

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

설명

이 함수는 사람이 교차로 `(0, V1)`에서 `(R - 1, V2)`로 지나갈 때 만나는 웜뱃 마릿수의 최소값을 계산하여야 한다.

파라미터

- `V1`: 수평 도로 0 위의 어느 교차로에서 사람이 출발하는지를 나타낸다. (`0 ≤ V1 ≤ C-1`).
- `V2`: 수평 도로 `R-1` 위의 어느 교차로로 사람이 도달하려고 하는지 나타낸다. (`0 ≤ V2 ≤ C-1`).
- 리턴값: 이 경로를 지나가는 과정에서 만나는 웜뱃 마릿수의 최소값

예제 세션

다음 예제 세션은 위의 예시를 나타낸 것이다.

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`

제약 조건

- 시간 제한 : 20초
- 메모리 제한 : 256MB
- `2 ≤ R ≤ 5,000`
- `1 ≤ C ≤ 200`
- change는 최대 500번 (`changeH()` 또는 `changeV()` 호출)

- `escape()`는 최대 200,000번 호출
- 한 세그먼트에 있는 웜뱃의 마릿수는 항상 1,000 이하

서브태스크

서브태스크	점수	추가적인 입력 제한 사항
1	9	`C = 1`
2	12	`R,C ≤ 20`이고 `changeH()` 또는 `changeV()` 호출은 없다.
3	16	`R,C ≤ 100`이고 `escape()` 호출은 최대 100번
4	18	`C = 2`
5	21	`C ≤ 100`
6	24	(없음)

테스트용 입력 형식

당신의 컴퓨터에 있는 샘플 그레이더는 입력을 파일 `wombats.in`에서 읽어들이는데, 포맷은 다음과 같아야 한다.

- 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: 이벤트가 발생하는 순서대로 한 줄마다 이벤트 하나씩

`C = 1`인 경우, 수평 도로에 존재하는 웜뱃의 마릿수를 나타내기 위해서 (줄 `2`부터 `R + 1`까지) 빈 줄들을 사용해야 할 필요는 없다.

각 이벤트는 다음 중 하나의 형식을 따라야 한다.

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

예를 들어, 위의 예시는 아래와 같이 입력되어야 한다.

```
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
```

언어 유의사항

C/C++	당신의 프로그램은 `#include "wombats.h"` 명령어를 통해 헤더 파일을 추가시켜야 한다.
Pascal	당신의 프로그램은 `unit Wombats`을 정의해야 한다. 모든 배열의 인덱스는 `1`이 아닌 `0`부터 시작한다.

예시를 위해서 컴퓨터에 있는 솔루션 템플릿을 참조하시오.