



전국 대학생 프로그래밍 대회 동아리 연합  
여름 대회 2018

본선 대회

후원



BAGELCODE

HYPERCONNECT

시스템 및 플랫폼 지원



## Problem A. 더위 피하기

“오늘 서울 낮 최고기온은 38도를 기록하겠습니다. 현재 전국에 폭염경보가 내려져 있고...”

성원이는 이 더운 날씨에 집에서 에어컨을 켜며 누워있지 않고 밖에 나온 것을 후회하고 있다. 길을 잃었기 때문이다. 처음 보는 건물과 간판들로 둘러싸인 성원이는 생명의 위협을 느끼기 시작했다. 이대로 집에 도착하지 못하고 T초가 지나면 성원이는 더위를 먹어 쓰러지고 말 것이다. 역시 이불 밖은 위험해.

생명의 위협을 느끼고 있는 성원이는 집에 가기 위해 일단 아무 곳으로나 가고 있다. 구체적으로 성원이는 1초에 1m씩 동서남북 방향으로 걸어갈 수 있다. 이렇게 아무 방향으로 계속 가다 보면 언젠가는 집에 도착할 수도 있지 않을까?

길을 잃은 성원이를 집에서 지켜보고 있던 승현이는 성원이가 진짜로 더위를 먹어서 쓰러지면 구조하러 가기로 했다. 승현이는 지도와 성원이의 현재 위치를 알고 있고, 어느 곳에 건물이 있어 가지 못하는지 알고 있다. 승현이는 성원이가 무사히 집에 도착할 수 있는 경우의 수를 알고 싶어한다. 하지만 승현이는 무더운 날씨에 성원이를 구조하러 나가기 위한 만반의 준비를 해야 하기 때문에 여러분에게 계산을 부탁했다.

성원이의 현재 위치와 성원이가 더위를 버틸 수 있는 시간, 집의 위치, 그리고 장애물의 개수와 위치가 주어질 때, 성원이가 T초 이내에 집에 올 수 있는 경우의 수를 계산해 주자. 도착 시간이 같아도 중간에 거친 경로가 다르면 다른 경우로 센다. 성원이는 매초 방향을 바꿀 수 있으며(이 길이 아니다 싶으면 왔던 방향으로 돌아갈 수도 있다), 장애물이 있는 지점에는 가지 못한다. 그리고 성원이는 한 번 집에 도착하면 이동을 멈추고 다시는 밖에 나가지 않는다.

### 입력

첫 번째 줄에 성원이의 현재 위치 좌표  $(X_s, Y_s)$ 를 나타내는 정수 2개가 주어진다. 두 번째 줄에 성원이가 더위를 버틸 수 있는 시간  $T(1 \leq T \leq 200)$ 가 주어진다. 세 번째 줄에 집의 위치 좌표  $(X_h, Y_h)$ 를 나타내는 정수 2개가 주어진다.

네 번째 줄에 장애물의 개수  $N(0 \leq N \leq 100,000)$ 이 주어진다. 그 이후 N개의 줄에 걸쳐 각 줄에 장애물의 위치 좌표  $(X_i, Y_i)$ 를 나타내는 정수 2개가 주어진다.

주어지는 모든 좌표는  $-100,000$  이상  $100,000$  이하의 정수이며 미터 단위이다. 또한 주어지는 모든 좌표의 위치는 서로 다르다.

### 출력

첫 번째 줄에 성원이가 T초 이내에 집에 도착하는 경우의 수를  $10^9+7$ 로 나눈 나머지를 출력한다.

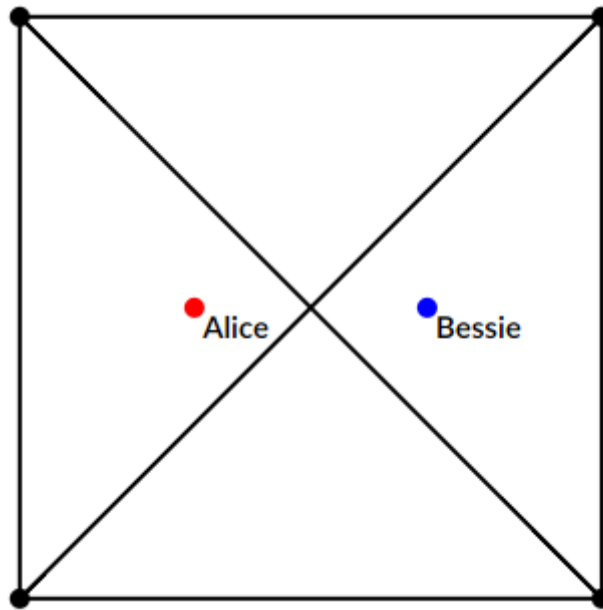
## 입출력 예제

Input	Output
1 2 4 0 0 1 0 2	2
1 2 4 0 0 2 0 2 1 1	0

## Problem B. 쉬운 최단경로 문제

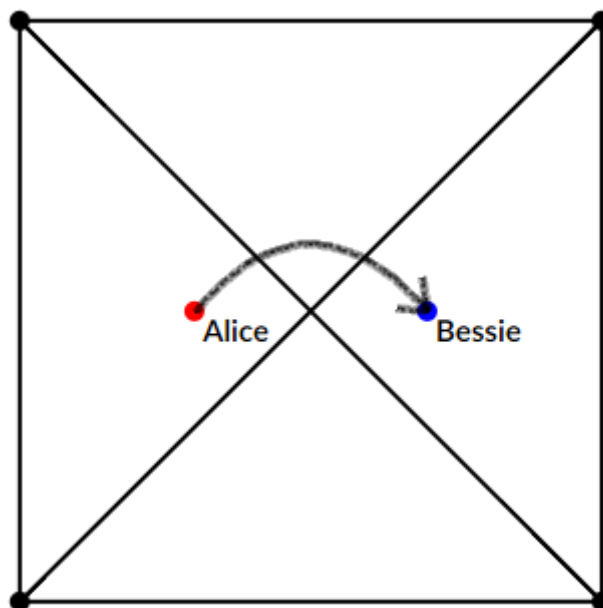
농부 John 의 목장은 특이하게 생겼다. 볼록 다각형을 이루는  $N$  개의 말뚝이 있고, 각 말뚝 쌍에 대해 두 말뚝을 선분으로 잇는 밧줄이 존재한다. 이 밧줄은 높이가相当해서 소들이 넘어 다니기 어렵다.

소 Alice 와 Bessie 는 서로의 가장 친한 친구이다. 하지만 농부 John 이 그들을 떨어뜨려 놓았기 때문에 현재는 멀리 위치해 있다.



위 그림은  $N=4$  일때의 한 예를 그림으로 나타낸 것이다. 총 6 개의 밧줄이 존재한다.

Alice 는 Bessie 가 있는 곳으로 놀러 가려고 한다. 하지만 밧줄을 넘는 것은 쉽지 않기 때문에, 넘는 밧줄의 개수를 최소로 하여 가려고 한다.



위 그림의 경우, 밧줄을 2 번 넘어서 가는 것이 최적임을 알 수 있다.

농부 John 이 두 소의 위치를 자주 바꾸기 때문에,  $Q$  개의 위치 쌍에 대해 각각의 경우 넘어야 하는 밧줄의 최소 개수를 구하고자 한다. Alice 와 Bessie 를 도와 문제를 해결해주자.

두 개 이상의 밧줄이 교차하는 점을 통해 넘어가는 것은 한꺼번에 넘는 것으로 인정되지 않는다. 예를 들어 3 개 밧줄이 교차하는 점을 통해 한꺼번에 넘어도 3 번으로 센다. 또한, 같은 밧줄을 여러 번 넘는 것은 여러 번 센다.

## 입력

첫 번째 줄에 말뚝의 수  $N(3 \leq N \leq 5,000)$ 이 주어진다. 두 번째 줄부터  $N$  개의 줄에는 2 개의 정수  $x$  와  $y$  가 주어진다. 이는 말뚝들의 좌표를 나타낸다. 말뚝의 정보는 반시계방향 순서대로 주어진다.

$N+2$  번째 줄에 쿼리의 개수  $Q(1 \leq Q \leq 10,000)$ 가 주어진다.  $N+3$  번째 줄부터  $Q$  개의 줄에는 4 개의 정수  $x_1, y_1, x_2, y_2$  가 주어진다.  $(x_1, y_1)$ 은 Alice 의 좌표를,  $(x_2, y_2)$ 은 Bessie 의 좌표를 나타낸다.

주어지는 모든 좌표의 절댓값은  $10^8$  이하이다.

$N$  개의 말뚝은 내각이 180 도 미만인 볼록 다각형을 이루고 있으며, 주어지는 소들의 위치는 밧줄 위에 있지 않다. 또한, 각 쿼리에 대해  $N$  개의 말뚝 및 두 소의 좌표는 서로 다르다.

## 출력

쿼리가 주어진 순서대로  $Q$  개의 줄에 걸쳐 각 줄에 그 쿼리에 대한 답을 출력한다.

## 입출력 예제

Input	Output
4	2
-5 -5	2
5 -5	0
5 5	
-5 5	
3	
0 2 0 -2	
2 0 -2 0	
6 0 0 -6	

## Problem C. 네트워크 해킹

1 번부터  $N$  번까지 번호가 붙어있는  $N$  개의 컴퓨터로 구성된 컴퓨터 네트워크가 있다. 그 사이에는 두 컴퓨터를 직접 연결하는 회선이  $N-1$  개 설치되어있다. 이 네트워크는 모든 컴퓨터 사이에 통신이 가능하다. 즉, 임의의 컴퓨터에서 시작하여 직접 연결된 회선들을 계속 따라가면 다른 모든 컴퓨터와 연결되어 통신할 수 있다.

각 회선마다 그 회선에 직접 연결되어있는 한쪽 컴퓨터에서 다른쪽 컴퓨터로 데이터를 보내는데 걸리는 시간이 고유하게 존재한다. 따라서 데이터가 여러 회선을 통해 전송되는데 걸리는 총 전송시간은 그 데이터가 지나간 회선들의 전송시간 각각을 더한 값이 된다.

임의의  $u$  번 컴퓨터에서 다른  $v$  번 컴퓨터로 데이터를 전송할 때는,  $u$  번 컴퓨터에서 출발해  $v$  번 컴퓨터로 도착하는 경로 중 총 전송시간이 가장 작은 경로로 데이터가 움직인다. 이 때의 총 전송시간이  $u, v$  사이의 데이터 전송시간이 된다.

‘최대 전송시간’은 네트워크 안의 임의의 두 컴퓨터 사이의 데이터 전송시간 중 가장 큰 값으로 정의된다. 이 ‘최대 전송시간’의 값으로 컴퓨터 네트워크의 가치가 결정된다. 즉, ‘최대 전송시간’이 작을 수록 가치가 높은 네트워크인 것이다.

세계 최고의 해커를 꿈꾸는 성원이는 해킹 실력을 기르려면 우선 물리적 해킹부터 연습해봐야겠다고 생각했다. 마침 평소에 이 네트워크에 대해 불만이 많았으므로, 이 네트워크를 타겟으로 삼아 물리적 해킹을 해보기로 했다. 구체적으로 다음과 같은 과정을 통해 이 네트워크를 공격하려고 한다.

1. 네트워크에 존재하는 회선 중 하나를 골라 그 회선을 끊는다.
2. 서로 다른 두 컴퓨터를 고른 후, 아까 끊은 그 회선과 동일한 전송시간을 갖는 회선을 그 사이에 설치한다.
3. 사람들이 네트워크가 공격당했다는 것을 눈치채면 안되기 때문에, 회선을 끊고 다시 설치한 후에도 모든 컴퓨터 사이에 통신이 가능해야 한다.

성원이는 위 과정을 딱 한 번만 수행해 이 네트워크의 가치를 최대한 떨어뜨리려고 한다. 즉, 해킹을 마친 상태에서 계산한 ‘최대 전송시간’이 가장 커지는 선택을 단 한 번만 할 것이다.

네트워크 안의 컴퓨터 수가 매우 많으므로 프로그램을 만들어 계산해야 하는데 성원이는 프로그래밍을 할 줄 모른다. 4 차산업혁명 시대에 프로그래밍도 모르는 불쌍한 성원이를 도와주자.

### 입력

첫 번째 줄에 네트워크 상의 컴퓨터 수  $N(2 \leq N \leq 200,000)$ 이 주어진다.

두 번째 줄부터  $N-1$  줄에 걸쳐 네트워크의 회선 정보가 주어진다. 각 줄마다 세개의 자연수  $a, b, t(1 \leq a, b \leq N, a \neq b, 1 \leq t \leq 10^7)$ 가 주어지는데, 이는  $a$  번 컴퓨터와  $b$  번 컴퓨터 사이를 직접 연결하는 회선이 존재하며, 그 회선의 전송시간이  $t$ 초 라는 뜻이다.

### 출력

첫 번째 줄에 최선의 해킹을 했을 때 가능한 ‘최대 전송시간’의 최댓값을 출력한다.

## 입출력 예제

Input	Output
5 3 5 2 3 1 5 1 2 1 4 1 3	11
3 1 2 3 1 3 4	7
6 1 2 3 1 3 1 1 4 5 1 5 6 1 6 2	14

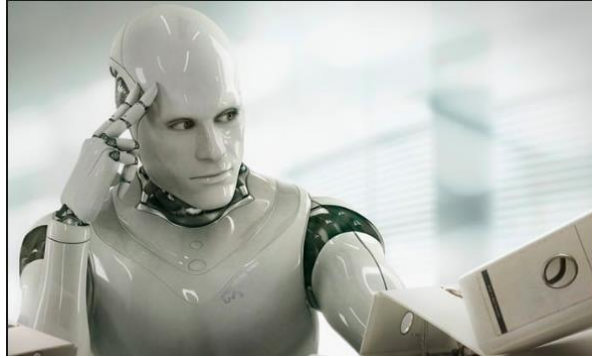
## Problem D. TV 동물 농장

ntopia는  $n$ 마리의 개와  $m$ 마리의 고양이가 참여하는 유명한 동물 예능 프로그램을 진행하고 있다. 모든 개는 1번부터  $n$ 번까지 번호가 붙어있고, 모든 고양이도 1번부터  $m$ 번까지 번호가 붙어있다. 어른들의 사정 때문에 ntopia는 높은 시청률을 만들어야만 했고, 때문에 시청률을 올리기 위한 수많은 편법들을 시도하고 있다.

개와 고양이 사이의 호감도에는 관심 없음, 싫음, 좋음의 3가지 종류가 있다. 신기하게도 호감도에는 대칭성이 있어서, 어떤 고양이가 어떤 개에게 느끼는 호감도는 그 개가 그 고양이에게 느끼는 호감도와 항상 같으며 반대의 경우도 마찬가지이다.

시청률을 올리기 위한 편법으로 아래와 같은 두 가지 종류의 작업을 수행할 수 있다.

- 1마리의 개, 그리고 번호가 인접한 2마리의 고양이를 고른다. 고른 동물들 사이에서 2개의 개-고양이 관계가 모두 좋으면 싫음으로, 모두 싫음이면 좋음으로 바꿀 수 있다.
- 1마리의 고양이, 그리고 번호가 인접한 2마리의 개를 고른다. 고른 동물들 사이에서 2개의 개-고양이 관계가 모두 좋으면 싫음으로, 모두 싫음이면 좋음으로 바꿀 수 있다.



동물들의 관심과 흥미를 빅데이터로 분석하여 가장 좋은 상태를 계산하고 있는 ntopia의 모습

ntopia는 모든 동물들을 분석하여 가장 시청률이 많이 나오는 호감도 상태를 계산하였다. 하지만 동물들 사이의 호감도를 바꾸는 것은 매우 어려운 작업이기 때문에, ntopia가 직접 작업하려던 계획은 실패했다. 결국 ntopia는 한국에서 가장 뛰어난 프로그래머인 여러분에게 도움을 요청했다.

초기 호감도 상태와 ntopia가 만들어야 하는 호감도 상태가 주어졌을 때, 만들 수 있는지의 여부와, 만들 수 있다면 작업의 최소 수행 횟수, 그리고 그 방법을 구하자.

### 입력

첫 번째 줄에  $n, m$ 이 주어진다. ( $1 \leq n, m \leq 100$ )

두 번째 줄에 초기 상태의 좋음 관계 수  $u$ 와 싫음 관계 수  $v$ 가 주어진다. ( $1 \leq u + v \leq 500$ )

세 번째 줄부터  $u$  개의 줄에 좋음 관계의 개 번호  $d_i$ 와 고양이 번호  $c_i$ 가 주어진다.  $u+3$  번째 줄부터  $v$  개의 줄에 싫음 관계의 개 번호  $d_j$ 와 고양이 번호  $c_j$ 가 주어진다( $1 \leq d_i, d_j \leq n, 1 \leq c_i, c_j \leq m$ ).

$u+v+3$  번째 줄부터 만들어야 하는 상태의 정보가 초기 상태의 정보와 같은 방식으로 주어지며, 제한 또한 동일하다.

입력으로 주어지지 않은 관계의 호감도는 “관심없음”임이 보장된다. 한 상태에서 같은  $(d_i, c_i)$  쌍이 2 번 이상 주어지는 경우는 없다.



## 출력

최종 상태를 만들 수 없다면 첫 번째 줄에  $-1$  을 출력한다.

만들 수 있다면 첫 번째 줄에 최소 횟수  $k$  를 출력한 뒤, 그 이후로  $k$  개의 줄에 호감도를 바꾸는 방법을 출력한다.  $d$  번 개와  $c$ ,  $c+1$  번 고양이를 선택했다면 "0 d c"를,  $c$  번 고양이와  $d$ ,  $d+1$  번 개를 선택했다면 "1 c d"를 출력한다(쌍따옴표는 생략).

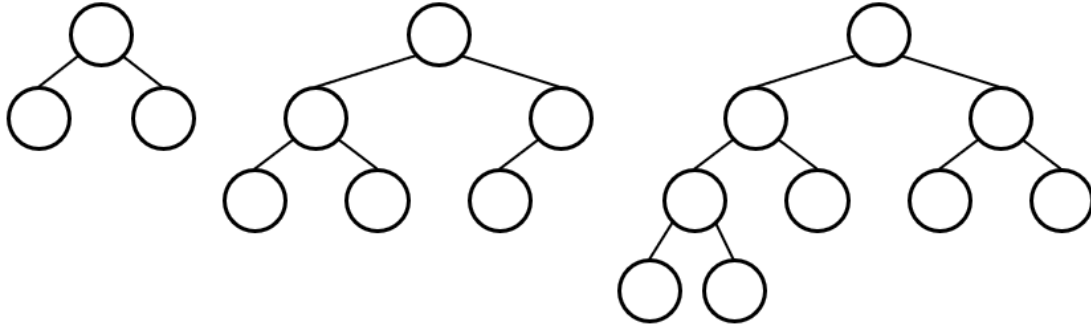
최소 수행 횟수를 달성하는 방법이 여러가지라면 그 중 아무거나 출력해도 된다.

## 입출력 예제

Input	Output
3 4	3
3 4	0 1 2
1 2	1 3 1
1 3	0 1 3
1 4	
2 3	
2 4	
3 3	
3 4	
1 6	
2 3	
1 4	
1 3	
1 2	
2 4	
3 3	
3 4	

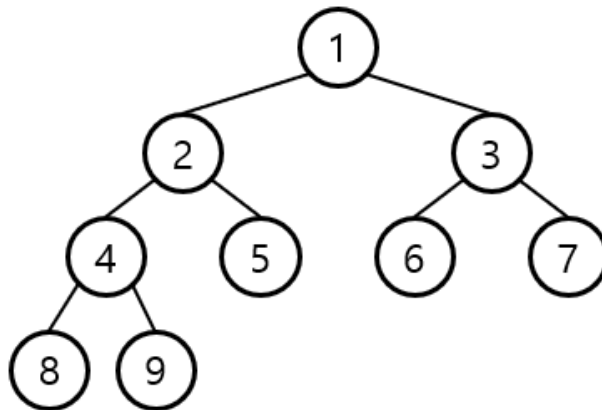
## Problem E. Thinking Heap

Binary heap 은 Heap 을 구현하는 방법의 하나이며 Complete binary tree 형태로 만들어진다. Complete binary tree 는 Binary tree 의 종류 중 하나로, 마지막 레벨을 제외한 나머지 레벨에는 노드가 꽉 차 있고 마지막 레벨에는 노드들이 왼쪽으로 쏠려있는 모습을 하고 있다.



〈그림〉 Complete binary tree 의 예

Complete binary tree 는 1 차원 배열을 이용하면 쉽게 구현할 수 있다. Complete binary tree 의 각 노드에 아래 그림과 같은 식으로 번호를 붙이고 이를 1 차원 배열에서의 index 로 삼으면 자연스럽게 구현할 수 있다.



〈그림〉 Complete binary tree 에 번호를 붙인 모습

이를 이용하면 Binary min-heap 에 원소를 삽입하는 알고리즘을 간단하게 구현할 수 있다. 아래 코드는 삽입 알고리즘을 C++로 구현한 코드이다(코드의 자잘한 문제들은 신경 쓰지 않기로 한다). 코드의 insert\_heap() 함수를 호출하면 우리가 만든 Binary min-heap 에 원소가 적절히 삽입된다.

```
const int MAX = 1000000;
int n = 0;
int heap[MAX];
void upheap(int id) {
    while (id > 1) {
        if (heap[id / 2] <= heap[id]) break;
        std::swap(heap[id], heap[id / 2]);
        id = id / 2;
    }
}
void insert_heap(int value) {
    heap[++n] = value;
    upheap(n);
}
```

<그림> Binary min-heap 에 원소를 삽입하는 알고리즘을 구현한 코드

비어있는 Binary min-heap 에 1 이상 N 이하의 서로 다른 자연수 N 개를 insert\_heap() 함수를 이용해 삽입할 것이다. N 개의 자연수를 전부 다 삽입한 후에, 자연수 k 가 heap 배열의 p 번째(배열의 맨 처음 공간을 0 번째로 생각한다)에 위치하도록 하고 싶다. 이렇게 만드는 삽입 순서를 찾는 프로그램을 작성하시오.

## 입력

첫 번째 줄에 자연수 N( $1 \leq N \leq 200,000$ )이 주어진다. 두 번째 줄에 자연수 k 와 p( $1 \leq k, p \leq N$ )가 공백으로 구분되어 주어진다.

## 출력

자연수 k 가 heap 배열의 p 번째에 위치하도록 하는 삽입 순서가 존재한다면 i 번째 줄에 i 번째로 삽입할 수를 출력한다. 가능한 삽입 순서가 여러 가지라면 그중 아무거나 하나를 출력해도 된다. 만약 그렇게 만드는 삽입 순서가 존재하지 않는다면 첫 번째 줄에 -1 을 출력한다.

## 입출력 예제

Input	Output
5	5
3 5	2
	4
	1
	3
6	-1
2 4	

## Problem F. parentheses recover

여는 괄호 '('와 닫는 괄호 ')'로 이루어진 문자열  $S$ 가 있다. 이 때, 다음 조건을 만족하는 문자열  $T$ 의 개수를 구하려고 한다.

- 조건 1.  $T$ 는 여는 괄호와 닫는 괄호로 이루어진 길이  $L$ 인 문자열이다.
- 조건 2.  $S$ 의 문자들과  $T$ 의 문자들을 각각 내부에서의 순서를 유지하면서 합쳐서 올바른 괄호 문자열이 되게 할 수 있다.

예를 들어,  $S = "(()"$ 이고  $L = 3$ 일 때  $T = ")("$ 이면 1, 2, 6번째 문자를  $S$ 의 문자로 하고 3, 4, 5번째 문자를  $T$ 의 문자로 하여  $"(())"$ 를 만들 수 있으므로  $T$ 는 조건을 만족한다.

그러나  $S = ")("$ 이고  $L = 3$ 일 때  $T = ")("$ 이면  $S$ 와  $T$ 를 합쳐 올바른 괄호 문자열을 만들 수 없으므로  $T$ 는 조건을 만족하지 않는다.

조건을 만족하는 문자열  $T$ 의 개수가 많아질 수 있으므로,  $10^9+7$ 로 나눈 나머지를 계산하도록 한다.

### 입력

첫 번째 줄에 문자열  $S$ 가 주어진다.  $S$ 는 길이가 1 이상 3,000 이하이며 '('와 ')'로만 구성되어있다.

두 번째 줄에 자연수  $L(1 \leq L \leq 3,000)$ 이 주어진다.

### 출력

첫 번째 줄에 문제의 정답을 출력한다.

### 입출력 예제

Input	Output
( 3	2

### 참고 사항

위 입력 예제의 경우, 조건을 만족하는  $T$ 는  $()$ ,  $)()$  의 2가지가 가능하다.

## Problem G. 성공

당연한 이야기지만, 성공으로 가는 길이 항상 평탄하지만은 않다. 온갖 장애물이 가득하고, 장애물에 막혀서 주저앉을 수도 있다. 그래서 그 장애물을 폭파하려고 한다.

성공으로 가는 길은  $N \times M$  격자 위에 놓여 있다. 성공으로 가려면 맨 왼쪽 위 칸에서 시작하여 장애물이 없는 상하좌우로 인접한 칸을 밟으면서 맨 오른쪽 아래 칸에 도착해야 한다. 한 번의 폭파 작업으로  $D \times D$  격자 내에 있는 모든 장애물을 없앨 수 있다. 하지만 세상에 공짜는 없는 법. 폭파 작업에도 큰 힘이 들기 때문에, 성공으로 가려면 최소 몇 번의 폭파 작업이 필요한지 알고 싶다.

### 입력

첫 번째 줄에 격자의 행의 개수  $N$ , 열의 개수  $M$ , 폭파의 범위  $D$ 가 주어진다( $D \leq N, M \leq 500, 1 \leq D \leq 100$ ).

그 다음  $N$  개의 줄에 격자의 각 행을 나타내는 길이  $M$ 의 문자열이 주어진다. "."은 장애물이 없는 칸, "#"은 장애물이 있는 칸이다. 출발 지점과 성공에는 장애물이 없다.

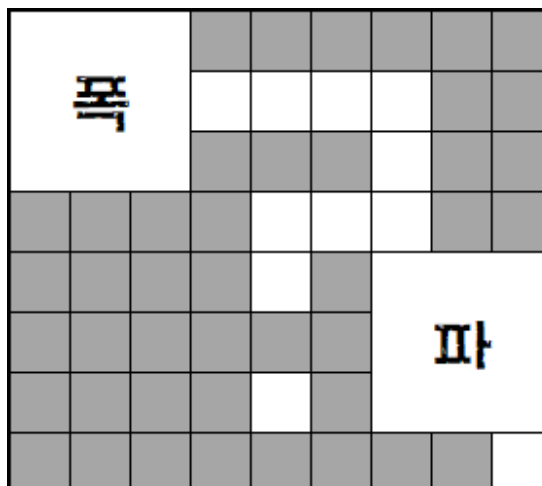
### 출력

첫 번째 줄에 폭파 작업의 최소 횟수를 출력한다.

### 입출력 예제

Input	Output
<pre>8 9 3 .##### ##.....## #####.## ####...## ####.#### ##### ####.#### #####. #####.</pre>	2

### 참고 사항



## Problem H. Make Similar

$n$ 개의 정수로 이루어진 수열  $A_1, A_2, \dots, A_n$ 이 있다. 당신은 이 수들을 최대한 비슷하게 만들고자 한다.

당신은 이 수열에서 두 수를 골라 둘 중 하나를 다른 하나에 더하는 작업을 여러 번 수행할 수 있다. 작업을 하고 싶은 만큼 수행하여  $n$ 개의 수 중 최댓값과 최솟값의 차이를 최소로 만들려고 한다. 이 때, 그 차이를 구하여라.

### 입력

첫 번째 줄에 수열의 길이  $n$  ( $1 \leq n \leq 10^5$ )이 주어진다.

두 번째 줄에 공백을 사이에 주고  $n$ 개의 정수  $A_1, \dots, A_n$  ( $-10^9 \leq A_i \leq 10^9$ )이 주어진다.

### 출력

첫 번째 줄에  $\max(A_1, \dots, A_n) - \min(A_1, \dots, A_n)$ 의 가능한 최솟값을 출력한다.

### 입출력 예제

Input	Output
3 1 2 3	1

### 참고 사항

$A_1 += A_2$  를 수행하면  $[3, 2, 3]$ 이 되어 최댓값 - 최솟값이 1이 된다.

## Problem I. 피아의 아틀리에 ~신비한 생명의 연금술사~

피아는 “새로운 생명을 창조하는 연금술”에 대해 연구하는 유명한 연금술사이다. 피아는 이 연금술로 수많은 생명을 창조하여 인류의 발전에 기여하고 있다. 하지만 피아는 빛의 속도로 달린다는 전설속의 동물 “달토끼”만큼은 만들 수 없었다.

연금술은 미생물이 서식하고 있는 가마를 이용해 이루어진다. 이 미생물들은 특정한 규칙을 만족해야만 “신비한 생명의 힘”이 발휘되어 새로운 생명을 창조할 수 있다(만족하지 않는다면 가마의 뜨거운 온도 때문에 미생물의 기분이 나빠져 연금술을 할 수 없다). 그 규칙은 다음과 같다.

- 가마는 가로, 세로 모두  $n$  칸인 격자 모양이다.
- 가마의 각 칸은 미생물 서식지이거나, 미생물 서식지가 아니다.
- $(i, j)$ ,  $(i+1, j)$ ,  $(i, j+1)$ ,  $(i+1, j+1)$  4 개의 칸에 있는 미생물 서식지 수의 총합을 2 로 나눈 나머지는  $c_{ij}$  와 같아야 한다. ( $1 \leq i, j \leq n-1$ ;  $0 \leq c_{ij} \leq 1$ )

특별한 생명을 창조하기 위해서는 특별한 조건을 만족해야 하며, 이를 “레시피”라고 한다. 레시피를 만족하지 않는다면 좀비같은 괴생명체가 만들어질 수 있기 때문에 매우 조심해야 한다. 피아가 만든 “달토끼 레시피”는 아래와 같은  $m$  개의 조건을 만족해야 한다.

- 달토끼 레시피의  $i$  번째 조건은 다음과 같다: " $s_i$  일부터  $e_i$  일까지 가마의  $(x_i, y_i)$ 칸은, 항상 서식지이거나, 항상 서식지가 아니어야 한다." ( $1 \leq i \leq m$ )

피아는 자신이 만든 레시피대로 연금술을 시도했지만, 특정 시간에는 위의 조건을 모두 만족하는 서식지 배치가 존재하지 않아 실패하고 말았다. 피아는 레시피를 수정하기 전에 여러분에게 다음과 같은 내용을 의뢰했다: “연금술이 이루어지는  $T$  일 각각에 대해, 달토끼 레시피를 포함한 모든 조건을 만족하는 미생물 서식지가 존재하는지 여부를 알려주세요.”

피아의 달토끼 레시피가 성공적으로 완성될 수 있도록 피아를 도와주자.

### 입력

첫 번째 줄에 가마의 크기  $n$  과 조건의 수  $m$ , 일 수  $T$ 가 주어진다. ( $2 \leq n \leq 3,000$ ,  $1 \leq m \leq 10^5$ ,  $1 \leq T \leq 10^5$ )

두 번째 줄부터  $n-1$  개의 줄에  $n-1$  개의 정수  $c_{ij}$ 가 공백 없이 주어진다. ( $0 \leq c_{ij} \leq 1$ )

$n$  번째 줄부터  $m$  개의 줄에 걸쳐 조건의 정보가 5 개의 수  $s_i, e_i, x_i, y_i, p_i$  로 주어진다( $1 \leq s_i \leq e_i \leq T$ ;  $1 \leq x_i, y_i \leq n$ ;  $0 \leq p_i \leq 1$ ). 이는  $x_i$  번째 행  $y_i$  번째 열에 위치한 칸이,  $s_i$  일부터  $e_i$  일까지,  $p_i$ 가 0 인 경우 미생물 서식지가 될 수 없는 경우를,  $p_i$ 가 1 인 경우 미생물 서식지가 되어야 하는 경우를 의미한다.

### 출력

첫 번째 줄에  $T$  개의 글자를 출력한다.  $k$  번째 글자로  $k$  번째 날에 미생물이 열기를 이겨낼 수 있는 배치가 존재한다면 1, 아니면 0 을 출력한다.

## 입출력 예제

Input	Output
3 5 9 01 10 1 5 1 1 1 2 6 1 3 0 3 7 2 2 1 4 8 3 1 0 5 9 3 3 0	111101111



## Problem J. 아기 석환 뚜루루 뚜루

석환이는 오늘 낮에 커피를 마셔서 잠에 들지 못하고 있다. 이불 속에서 너무 심심한 나머지 “아기 석환” 노래를 잠에 들 때까지 부르려고 한다. 석환이는 UCPC 2018 출제진 중의 한 명인 석환(seokhwan)이라는 달리, 자신의 이름을 sukhwan 이라고 쓴다. 노래가 끝나면 다시 처음부터 부른다. 아기 석환 노래는 아래와 같다.

baby sukhwan tururu turu  
very cute tururu turu  
in bed tururu turu  
baby sukhwan

하지만 석환이는 계속 노래를 똑같이 부르기는 심심해서, 노래가 한 번 끝날 때마다 “tururu”와 “turu”에 “ru”를 한 번씩 더 추가해서 다음과 같이 부르려고 한다.

baby sukhwan tururu turu  
very cute tururu turu  
in bed tururu turu  
baby sukhwan

baby sukhwan turururu tururu  
very cute turururu tururu  
in bed turururu tururu  
baby sukhwan

baby sukhwan tururururu turururu  
very cute tururururu turururu  
in bed tururururu turururu  
baby sukhwan

...

이 때, 석환이가 부르는 노래의 N번째 단어는 무엇일까?

### 입력

첫 번째 줄에 자연수  $N(1 \leq N \leq 1,000,000)$ 이 주어진다.

### 출력

첫 번째 줄에 석환이가 N번째로 부를 단어를 출력한다. 여기서 단어란 가사 중 공백으로 구분되는 연속된 알파벳 소문자열을 뜻한다. 단, 출력할 단어가 “tururu...ru”일 때, “ru”가  $k(k \geq 5)$ 번 반복되면 “tu+ru\*k”와 같이 출력한다.

### 입출력 예제

Input	Output
1	baby
17	turururu
45	tu+ru*5

## Problem K. 간단한 문제

자연수  $n$ ,  $m$  과 자연수 수열  $A_1, A_2, \dots, A_m$  이 주어졌을 때, 다음 등식을 만족하는 자연수 수열  $B_1, B_2, \dots, B_m$  을 구하라.

$$1 + \frac{2^m - 1}{n} = \frac{(A_1 + B_1)(A_2 + B_2) \cdots (A_m + B_m)}{B_1 B_2 \cdots B_m}$$

### 입력

첫 번째 줄에 자연수  $n$  과  $m$  이 공백으로 구분되어 주어진다. ( $1 \leq n \leq 10^{15}, 1 \leq m \leq 50$ )

두 번째 줄에 수열  $A_m$  을 나타내는 정수  $m$  개가 공백으로 구분되어 주어진다. ( $1 \leq A_i \leq 1,000$ )

### 출력

첫 번째 줄에 등식을 만족하는 수열  $B_m$  을 공백으로 구분하여 출력한다. 각  $B_i$  는 1 이상  $3 \times 10^{18}$  이하여야 한다. 등식을 만족하는 수열이 여러가지라면 그 중 아무거나 출력해도 된다. 만약 등식을 만족하는 수열이 존재하지 않는다면 첫 번째 줄에  $-1$  을 출력한다.

### 입출력 예제

Input	Output
3 4 1 2 2 1	1 2 16 3
10 3 27 283 42	81 1415 672

## Problem L. Piet

Piet 은 프로그래밍 언어의 하나로, 코드가  $N \times M$  의 2 차원 이미지로 되어 있는 것이 특징이다. 이름의 유래는 추상화로 유명한 네덜란드의 화가인 피트 몬드리안(Piet Mondrian)이다. 다음의 이미지는 "Hello, world!"를 출력하는 Piet 코드이다.

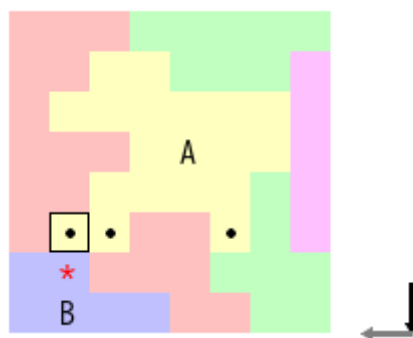


Piet 프로그램에서 단위 정사각형을 코델(Codel)이라고 하며 각 코델마다 특정 색깔이 칠해져있다. 이는 비트맵 이미지의 픽셀에 대응되는 개념이다. 같은 색깔의 코델이 상하좌우로 연결된 블록이 프로그램의 최소 실행 단위가 된다. 맨 처음 실행되는 블록은 가장 왼쪽 위 코델이 포함된 블록이며, 규칙에 따라 다음에 실행될 블록으로 이동하는 과정을 반복한다.

현재 블록에서 어떤 블록으로 이동할지를 결정하기 위해 DP(Direction Pointer)와 CC(Codel Chooser)라는 두 가지 값이 존재한다. DP의 값은 왼쪽, 오른쪽, 위쪽, 아래쪽 중 하나이고, CC의 값은 왼쪽 혹은 오른쪽 중 하나이다. 프로그램이 처음 실행될 때 DP의 값은 오른쪽, CC의 값은 왼쪽이다. 어떤 블록으로 이동할지를 선택하는 기준은 다음과 같다.

- 현재 블록의 코델들 중에서 DP의 방향으로 가장 멀리 위치한 코델들을 찾는다. 블록이 불록하지 않은 경우 이 코델들은 이어지지 않은 형태일 수 있다.
- 위에서 찾은 코델들 중 DP 방향을 향했을 때 CC의 방향으로 가장 끝에 있는 코델을 선택한다. 이 때 CC의 방향은 상대적인 방향임에 유의하라. 예를 들어, DP가 아래쪽을 가리키고 CC가 오른쪽을 가리키는 경우, 선택되는 코델은 가장 왼쪽에 있는 코델이 된다.
- 위에서 선택한 코델에서 DP의 방향으로 맞닿은 코델이 포함된 블록이 다음 블록이 된다.

이에 대한 예시로, 다음과 같은 그림에서 어떤 블록으로 이동할지를 선택하는 과정을 살펴보자.

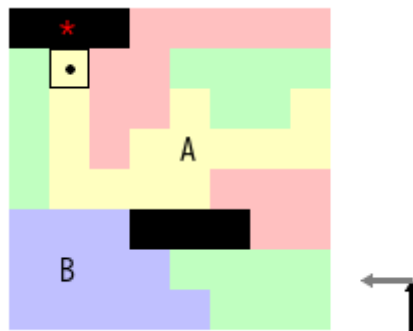


각 칸에 적힌 글자는 색깔을 나타내는 값이다. 현재 블록은 A 블록이고, DP의 값(검은 화살표)은 아래쪽, CC의 값(회색 화살표)은 오른쪽인 상태이다. 이 때 A 블록의 코델들 중에서 DP의 방향으로 가장 멀리 위치한 코델은 A 블록에서 가장 아래쪽에 위치한 코델로, 그림에서 점으로 표시된 코델이다. 이 블록들 중 CC의 방향으로 가장 끝에 있는 코델은 그림에서 테두리로 표시된 블록이다. 이 코델의 아래쪽(DP의 방향)으로 맞닿은 코델(빨간색 별표)이 속한 블록인 B 블록이 다음 블록이 된다.

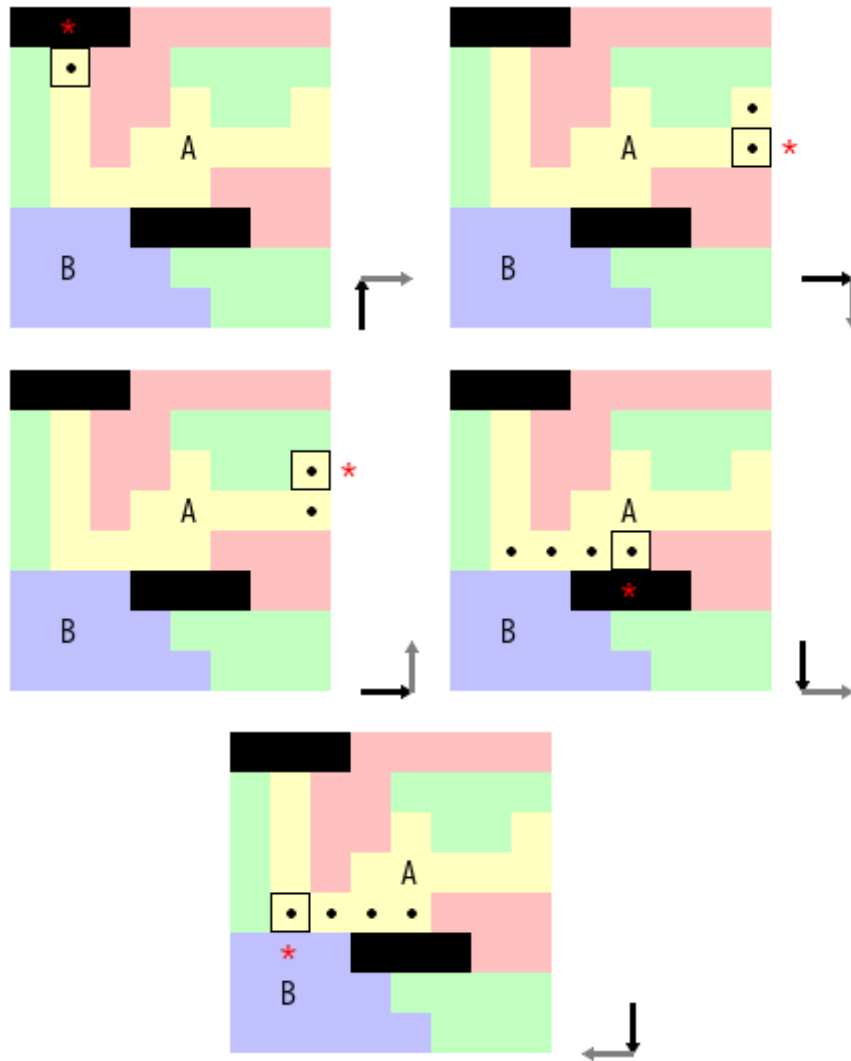
검은색 블록 혹은 이미지의 경계 바깥은 이동할 수 없는 구역이다. 만약 다음으로 이동하려는 블록이 검은색이거나 이미지의 경계를 벗어나는 경우는 다음의 방법으로 진행한다.

1. CC의 값이 왼쪽인 경우 오른쪽으로, 오른쪽인 경우 왼쪽으로 바꾼 후 다시 이동을 시도한다.
2. CC의 값을 바꾼 후에도 이동할 수 없는 경우, CC의 값을 유지하며 DP를 시계방향으로 회전한 후 다시 이동을 시도한다.
3. 시계방향으로 회전한 후에도 이동할 수 없는 경우, 1번으로 되돌아간다.
4. 위 과정을 계속 반복하여 총 8가지의 경우를 모두 시도했는데도 이동할 수 있는 블록을 찾지 못한 경우, 프로그램이 종료된다.

예를 들어, 다음과 같은 그림에서 어떤 블록으로 이동할지를 선택하는 과정을 살펴보자.



현재 블록은 A 블록이고, DP의 방향은 위, CC의 방향은 왼쪽인 상태이다. 이번에도 경계에 위치한 코델을 점으로, CC에 의해 선택된 코델을 테두리로 표시하였다. 현재 상태에서 선택된 다음 블록은 검은색 블록이므로 이동할 수 없다. 따라서 CC의 값을 오른쪽으로 바꾼 뒤 다시 다음 블록을 찾는데, 역시 검은색 블록이므로 이동할 수 없다. 이제 DP의 방향을 시계방향으로 회전하여 오른쪽으로 바꾼 뒤 블록을 찾는다(이 때 CC의 값은 바뀌지 않는다). 선택된 코델과 맞닿은 블록은 이미지의 경계 밖이므로 CC의 값을 왼쪽으로 바꾼 뒤 다시 시도한다. 이번에는 다른 코델이 선택되지만 역시 맞닿은 블록이 이미지의 경계 밖이므로 이동할 수 없다. 다시 DP의 방향을 시계방향으로 회전하여 아래로 바꾼 뒤 블록을 찾는다. 이 때 선택된 블록은 검은색 블록으로 역시 이동할 수 없고, CC의 값을 다시 오른쪽으로 바꾸면 B 블록이 선택되고 이는 이동할 수 있는 블록이다. 따라서 다음 블록은 B 블록이 된다. 이상의 과정을 그림으로 나타내면 다음과 같다.



Piet 으로 작성된 프로그램이 주어졌을 때, 이 프로그램이 어떤 블록을 거치며 실행되는지 알고 싶어졌다. 입력된 Piet 프로그램에 대해, 프로그램이 종료할 때까지 거치는 블록의 색깔을 출력하는 프로그램을 작성하시오.

\* 실제 Piet 의 연산 중에는 픽셀의 색깔에 따라 DP 혹은 CC 가 바뀌는 연산이 있지만, 이 문제에 한해서는 위에서 설명된 규칙에 의해서만 DP 와 CC 가 변경되는 것으로 간주한다.

## 입력

첫 번째 줄에 자연수  $N$  과  $M$  이 주어진다. ( $1 \leq N, M \leq 100$ )

두 번째 줄부터  $N$  줄에 걸쳐  $M$  개의 글자로 이루어진 문자열이 입력되며, 이는 Piet 프로그램 이미지의 각 코델을 의미한다. 각각의 글자는 알파벳 대문자로 색깔을 나타내는 값이며, X 는 검은색을 의미한다. 종료되지 않는 프로그램은 입력되지 않는다.

## 출력

첫 번째 줄에 프로그램이 종료될 때까지 거치는 블록들의 색깔을 순서대로 출력한다.

## 입출력 예제

Input	Output
3 3 ABB XBX CCC	ABC
5 5 ILOVE ZZZYE WXXYU XCCXC XCPPP	ILOVEUCPC