

문제 1. 유괴2

입력 파일: standard input
출력 파일: standard output
시간 제한: 5초
메모리 제한: 512MB

어느 화창한 날 도에서 유괴사건이 일어났다. 범인은 Anna와 Bruno고 차를 통해 유괴 현장에서 도망쳤다고 추정하고 있다. 차는 아직 발견되지 않았다. 경찰은 아직도 차의 행방을 쫓고 있다.

유괴범은 H 개의 동서 방향 도로가 있고 W 개의 남북 방향 도로가 있는 격자 모양의 도에서 차를 운전하고 있다. 두 교차로의 사이는 1km이다.

각 도로는 **혼잡도**라고 불리는 정수가 붙어있다. 북쪽에서 i 번째 ($1 \leq i \leq H$) 동서 방향 도로의 혼잡도는 A_i 이고, 서쪽에서 j 번째 ($1 \leq j \leq W$) 남북 방향 도로의 혼잡도는 B_j 이다. 이 $H + W$ 개의 값은 서로 다르다. 각 도로에 대해 혼잡도는 어느 지점에 있든 같다.

조사는 유괴범이 다음과 같은 방법으로 이동했다는 것을 알아냈다.

- 도시 밖이나, 도로 밖으로 나가지는 않았다.
- 처음 유괴범은 유괴한 교차로으로부터 이동 가능한 방향 중 어떤 한 방향을 택해 움직였다.
- 어떤 교차로에 도착했을 때, 현재 달리는 방향의 도로보다 교차하는 다른 도로의 혼잡도가 더 크면 그 교차로에서 회전한다. 회전 할 수 있는 방향이 둘이면 양 쪽 모두 고를 가능성이 있다.
- 어떤 교차로에 도착했을 때, 현재 달리는 방향의 도로가 교차하는 다른 도로보다 혼잡도가 더 크면 계속 직진한다. 만약 도시의 경계에 도달하여 직진할 수 없을 때는 그 자리에서 멈춘다.

유괴사건이 일어났을 거라고 추정되는 Q 개의 후보지가 있다. 이 Q 개의 후보지는 서로 다르다. 조사팀의 사람을 정하기 위해서 경찰은 각 후보지에 대해서 유괴사건이 그 후보지에서 일어났을 경우에 범의자가 운전할 수 있는 최대 거리를 알고 싶다.

각 Q 개의 질의에 대해, 후보지에 대해서 범의자가 운전할 수 있는 최대 거리를 구하여라.

입력 형식

다음 정보가 표준 입력으로 주어진다.

- 첫째 줄에는 공백으로 구분된 두 정수 H, W, Q 가 주어진다. 이는 도시가 동서방향 도로가 H 개, 남북방향 도로가 W 개 있으며, 범죄 후보지가 Q 개 라는 의미이다.
- 둘째 줄에는 공백으로 구분된 H 개의 정수 A_1, A_2, \dots, A_H 가 주어진다. 이는 북쪽에서 i 번째 ($1 \leq i \leq H$) 동서 방향 도로의 혼잡도가 A_i 라는 의미이다.
- 셋째 줄에는 공백으로 구분된 W 개의 정수 B_1, B_2, \dots, B_W 가 주어진다. 이는 서쪽에서 j 번째 ($1 \leq j \leq W$) 남북 방향 도로의 혼잡도가 B_j 라는 의미이다.
- 다음 Q 개의 줄의 k 번째 ($1 \leq k \leq Q$) 줄에는 공백으로 구분된 두 정수 S_k, T_k 가 주어진다. k 번째 유괴사건 후보지가 북쪽에서 S_k 번째 동서 방향 도로와 서쪽에서 T_k 번째 남북 방향 도로의 교차로라는 의미이다.

출력 형식

표준 출력으로 Q 개의 줄을 출력하여라. k 번째 줄은 k 번째 후보지에 대해서 범의자가 운전할 수 있는 최대 거리를 (km단위로) 출력해야 한다.

제한

- $2 \leq H \leq 50\,000$.
- $2 \leq W \leq 50\,000$.
- $2 \leq Q \leq 100$.
- $1 \leq A_i \leq 1\,000\,000\,000$ ($1 \leq i \leq H$).
- $1 \leq B_j \leq 1\,000\,000\,000$ ($1 \leq j \leq W$).
- $H + W$ 개의 정수 $A_1, A_2, \dots, A_H, B_1, B_2, \dots, B_W$ 는 서로 다르다.
- $1 \leq S_k \leq H$ ($1 \leq k \leq Q$).
- $1 \leq T_k \leq W$ ($1 \leq k \leq Q$).
- $(S_k, T_k) \neq (S_l, T_l)$ ($1 \leq k < l \leq Q$).

서브태스크 1 (13 점)

- $H \leq 8$
- $W \leq 8$
- $Q = 1$

서브태스크 2 (10 점)

- $H \leq 2\,000$
- $W \leq 2\,000$
- $Q = 1$

서브태스크 3 (17 점)

- $Q = 1$

서브태스크 4 (4 점)

- $H \leq 2\,000$
- $W \leq 2\,000$

서브태스크 5 (56 점)

추가 제한조건이 없다.

예제

standard input	standard output
3 3 5	4
3 2 6	5
1 4 5	4
1 1	4
1 2	2
2 2	
3 1	
3 3	

예를 들어, 세 번째 질의에 대해서 운전자가 이동한 거리는 다음 방법으로 최대가 된다.

- 북쪽에서 두 번째 동서 방향 도로와 서쪽에서 두 번째 남북 방향 도로의 교차로에서 동쪽으로 1km 움직였다.
- 북쪽에서 두 번째 동서 방향 도로와 서쪽에서 세 번째 남북 방향 도로의 교차로에서 남쪽 혹은 북쪽으로 움직일 수 있다. 남쪽을 골라서 1km 움직였다.
- 북쪽에서 세 번째 동서 방향 도로와 서쪽에서 세 번째 남북 방향 도로의 교차로에서 서쪽으로만 움직일 수 있다. 서쪽으로 1km 움직였다.
- 북쪽에서 세 번째 동서 방향 도로와 서쪽에서 두 번째 남북 방향 도로의 교차로에서 서쪽으로만 움직일 수 있다. 서쪽으로 1km 움직였다.
- 북쪽에서 세 번째 동서 방향 도로와 서쪽에서 첫 번째 남북 방향 도로의 교차로에서 더 움직일 수 없다. 그 장소에서 멈췄다.

위와 같이 움직인 경우에 이동한 거리는 4km 이다.

standard input	standard output
4 5 6	7
30 10 40 20	6
15 55 25 35 45	9
1 3	4
4 3	6
2 2	9
4 1	
2 5	
3 3	

문제 2. 도시

시간 제한: 3초
메모리 제한: 256MB

JOI 왕국에는 다양한 도시가 있다. 도로 체계는 다음 조건을 만족한다:

- (조건 1) 도시는 0번부터 $N - 1$ 번까지의 번호가 붙어있다. 여기서, N 은 JOI 왕국의 도시의 개수다.
- (조건 2) 도시는 $N - 1$ 개의 도로로 연결되어 있다. 도로는 양방향으로 통행할 수 있다. 어떤 도시에서 다른 도시까지 몇개의 도로를 거치면 통행할 수 있다.
- (조건 3) 0번 도시로부터 다른 도시까지 최대 18개의 도로를 거치면 갈 수 있다.

어느 날 JOI 왕국에서 많은 사람이 0번 도시에서 다른 도시로 옮겨갔다. 많은 JOI 왕국의 국민이 서로 다른 두 개의 도시를 목적지로 하고 있었기 때문에 다음과 같은 질문을 했다. 2개의 서로 다른 두 도시 X , Y 에 대해 (0), (1), (2) 중에 어느 것이 성립하는가?

- (0) 0번 도시부터 도시 X 까지 가는 도중에 도시 Y 를 반드시 거친다.
- (1) 0번 도시부터 도시 Y 까지 가는 도중에 도시 X 를 반드시 거친다.
- (2) (0)과 (1) 모두 아니다.

위의 조건에서 (0), (1), (2) 중에 정확히 하나가 성립한다. 단, X 가 0번 도시인 경우에 Y 에 상관없이 (1)이 성립하는 것으로 한다. 또한, Y 가 0번 도시인 경우에 X 에 상관없이 (0)이 성립하는 것으로 한다.

JOI 왕국 뿐만 아니라 다른 나라에 대해서도 도로 체계의 (조건 1)~(조건 3)이 성립하는 것이 잘 알려져 있다. 그렇기 때문에 다른 나라에서도 쓸 수 있게 JOI 왕국에는 다음과 같은 2개의 기계를 만들려고 한다.

- (기계 1) 도시의 개수 N 과 도로의 정보를 보고, 각각의 도시에 0 이상 $2^{60} - 1$ 이하의 정수를 붙인다.
- (기계 2) 도시 X , Y 에 대해 (기계 1)이 도시 X , Y 에 부여한 번호를 보고 질문에 대해 대답한다.

큰 번호가 붙을 경우 이를 관리하는 것은 힘들다. 그렇기 때문에 가능한 한 적은 번호가 붙게 기계를 개발하려고 한다.

기계 2가 사용될 때 도시의 수 N 이나 도로의 정보를 주지 않음에 유의하여야.

구현 명세

당신은 같은 프로그래밍 언어로 작성된 파일 두 개를 작성해야 한다.

첫 번째 파일의 이름은 `Encoder.c` 혹은 `Encoder.cpp`이다. 이 파일은 기계 1을 구현한 파일이며, 다음 함수를 구현해야 한다. 이 파일은 `Encoder.h`를 include해야 한다.

- `void Encode(int N, int A[], int B[])`

이 함수는 각 테스트 케이스마다 정확히 한 번 불린다.

- 인자 N 은 도시의 개수 N 을 의미한다.
- 인자 $A[]$, $B[]$ 는 길이 $N - 1$ 의 배열이고, 도로의 정보를 의미한다. 원소 $A[i]$, $B[i]$ ($0 \leq i \leq N - 2$)는 $A[i]$ 번 도시와 $B[i]$ 번 도시를 직접 연결하는 도로가 있다는 것을 의미한다.

당신의 프로그램은 다음의 함수를 호출해야 한다.

- `void Code(int city, long long code)`
이 함수는 도시에 정수를 붙이는 것을 의미한다.

- * 인자 `city`는 정수를 붙일 도시의 번호를 의미한다. `city`는 0 이상 $N - 1$ 이하이다. 만약에 이 범위를 벗어나서 함수를 호출 한 경우 **오답 [1]**이 된다. 같은 `city`를 인자로 하여 함수를 두 번 이상 호출한 경우 **오답 [2]**이 된다.
- * 인자 `code`는 `city`번 도시에 붙일 정수이다. `code`는 값은 0 이상 $2^{60} - 1$ 이어야 한다. 만약에 이 범위를 벗어나서 함수를 호출한 경우 **오답 [3]**이 된다.

함수 `Code`는 프로그램에서 정확히 N 번 호출되어야 한다. `Encode` 함수가 종료되었을 때, `Code`이 호출된 횟수가 N 과 다르면 **오답 [4]**이 된다.

만약 `Encode`가 함수를 올바르게 호출될 경우 프로그램이 종료된다.

두 번째 파일의 이름은 `Device.c` 혹은 `Device.cpp`이다. 이 파일은 기계 2를 구현한 파일이며, 다음 함수를 구현해야 한다. 이 파일은 `Device.h`를 `include`해야 한다.

- `void InitDevice()`

이 함수는 기계 2의 초기화에 대응한다. 다음 `Answer` 함수가 불리기 전에 `InitDevice`는 정확히 한 번만 불린다.

- `int Answer(long long S, long long T)`

이 함수는 각 질문에 대응한다. 각 질문에 대응하여 `Answer`가 한 번 불린다.

- 인자 `S`, `T`는 두 개의 서로 다른 도시 X , Y 에 붙은 정수이다.
- 함수 `Answer`는 질문에 답하기 위해, 다음의 조건을 만족하는 값을 반환해야 한다.
 - * 0번 도시부터 도시 X 까지 가는 도중에 도시 Y 를 반드시 거치는 경우, 0을 반환한다.
 - * 0번 도시부터 도시 Y 까지 가는 도중에 도시 X 를 반드시 거치는 경우, 1을 반환한다.
 - * 위 두 경우 모두 아닌 경우 2를 반환한다.

즉, `Answer`의 반환값은 0 이상 2 이하의 정수여야 한다. 이 범위 밖의 수를 반환한 경우 **오답 [5]**이 된다. 또한, 범위 내의 정수여도, 위 조건을 만족하지 않는 값을 반환한 경우 **오답 [6]**이 된다.

채점은 다음과 같은 방식으로 진행된다. 만약 프로그램이 오답으로 판단된 경우, 즉시 채점은 종료된다.

1. 함수 `Encode`를 1회 호출한다.
2. 함수 `InitDevice`를 1회 호출한다.
3. 각 테스트케이스에 대해, 기계 2에게 Q 개의 질문을 준다. j 번째 ($1 \leq j \leq Q$) 질문에 대해, 함수 `Answer`는 인자 `S`에 S_j 를, 인자 `T`에 T_j 를 넣어서 호출되고, S_j 와 T_j 는 함수 `Encode`가 X_j 번과 Y_j 번 도시에 설정한 값이다.
4. 당신의 프로그램은 정답이 된다.

참고 사항

- 실행 시간과 메모리 사용량은 채점 방식의 1, 2, 3에서 계산된다.
- 당신의 프로그램은 채점 방식 1.의 `Encode`, 채점 방식 2.의 `InitDevice` 혹은 에서 채점 방식 3.의 `Answer`에서 오답으로 판단되면 안된다. 당신의 프로그램은 런타임 에러 없이 실행되어야 한다.
- 당신의 프로그램은 내부에서 사용할 목적으로 함수나 전역변수를 사용할 수 있다. 제출한 프로그램은 그레이더와 함께 컴파일 되어 하나의 실행파일이 된다. 모든 전역변수나 내부 함수는 다른 파일과의 충돌을 피하기 위해 `static`으로 선언되어야 한다. 기계 1과 기계 2는 2개의 별개의 프로세스로 실행되기 때문에 채점될 때 전역변수를 공유하지 않는다.

- 당신의 프로그램은 표준 입출력을 사용해서는 안 된다. 당신의 프로그램은 어떠한 방법으로도 다른 파일에 접근해서는 안 된다.

당신은 대회 홈페이지의 아카이브에서 프로그램을 테스트하기 위한 목적의 샘플 그레이더를 받을 수 있다. 아카이브는 당신의 프로그램의 예제 소스 또한 첨부되어 있다. 샘플 그레이더는 파일 `grader.c` 혹은 `grader.cpp`이다. 당신의 프로그램이 `Encoder.c`와 `Device.c` 혹은, `Encoder.cpp`와 `Device.cpp` 인 경우 다음 커맨드로 컴파일할 수 있다.

- `C g++ -std=c11 -O2 -o grader grader.c Encoder.c Device.c -lm`
- `C++ g++ -std=c++14 -O2 -o grader grader.cpp Encoder.cpp Device.cpp`

컴파일이 성공적이면, 파일 `grader`가 생성된다.

실제 그레이더와 샘플 그레이더는 다름에 주의하여라. 샘플 그레이더는 하나의 프로세스에서 실행되며, 입력을 표준 입력으로 부터받고, 출력을 표준 출력에 출력한다.

입력 형식

샘플 그레이더는 다음 형식으로 표준 입력으로 부터 데이터를 입력받는다.

- 첫째 줄에 두 개의 정수 N, Q 가 공백으로 구분되어 주어진다. 이는 N 개의 도시가 있고, 질문이 Q 개라는 의미이다.
- 다음 $N - 1$ 개의 줄의 $i + 1$ 번째 ($0 \leq i \leq N - 2$) 줄에는, 두 개의 정수 A_i 와 B_i 가 공백으로 구분되어 주어진다. 이는 A_i 번 도시와 B_i 번 도시가 도로로 직접 연결되어 있다는 의미이다.
- 다음 Q 개의 줄의 j 번째 ($1 \leq j \leq Q$) 줄에는, 3개의 정수 X_j, Y_j, E_j 가 공백으로 구분되어 주어진다. 이는 j 번째 질문에 대해, $X = X_j$ 이고 $Y = Y_j$ 라는 것을 의미하고, 당신의 프로그램이 질문에 대해 E_j 이외의 답을 반환한 경우에는 샘플 그레이더가 오답으로 판단한다는 의미이다.

출력 형식

프로그램이 정상적으로 종료되었다면, 샘플 그레이더는 다음과 같은 정보를 표준 출력에 출력한다. (따옴표는 출력하지 않는다.)

- 정답인 경우, 도시에 붙은 정수의 최댓값을 “Accepted max_code=123456”과 같은 형식으로 출력한다.
- 오답으로 판단 된 경우, 오답의 종류를 “Wrong Answer [1]”과 같은 형식으로 출력한다.

프로그램이 다양한 오답의 종류에 속해 있으면 샘플 그레이더는 그중 하나만 출력할 것이다.

제한

N, Q, A_i, B_i, X_j, Y_j 의 의미에 대해서는 입력 형식을 참고하여라.

- $2 \leq N \leq 250\,000$.
- $1 \leq Q \leq 250\,000$.
- $0 \leq A_i \leq N - 1$ ($0 \leq i \leq N - 2$).
- $0 \leq B_i \leq N - 1$ ($0 \leq i \leq N - 2$).
- $A_i \neq B_i$. ($0 \leq i \leq N - 2$)
- 도시 0부터 어떤 다른 도시에 대해서도, 18개 이하의 도로를 사용하여 오가는 것이 가능하다.

- $0 \leq X_j \leq N-1$ ($1 \leq j \leq Q$).
- $0 \leq Y_j \leq N-1$ ($1 \leq j \leq Q$).
- $X_j \neq Y_j$ ($1 \leq j \leq Q$).

서브태스크 1 (8 점)

- $N \leq 10$.

서브태스크 2 (92 점)

추가 제한조건이 없다. 이 서브태스크에 대해서는 다음과 같이 점수가 정해진다.

- 이 서브태스크의 모든 테스트케이스에 대해 도시에 할당된 정수의 최댓값을 L 이라고 하자.
- 이 때, 이 서브태스크의 점수는
 - $2^{38} \leq L$ 인 경우, 0점.
 - $2^{36} \leq L \leq 2^{38} - 1$ 인 경우, 10점.
 - $2^{35} \leq L \leq 2^{36} - 1$ 인 경우, 14점.
 - $2^{34} \leq L \leq 2^{35} - 1$ 인 경우, 22점.
 - $2^{28} \leq L \leq 2^{34} - 1$ 인 경우, $\lfloor 372 - 10 \log_2(L+1) \rfloor$ 점 ($\lfloor x \rfloor$ 는 x 를 넘지 않는 최대의 정수)
 - $L \leq 2^{28} - 1$ 인 경우, 92점

채점 시스템에 대해서 $2^{38} \leq L$ 인 경우에, 이 서브태스크의 정보가 “정답: 0점”과 같이 표시되는 것이 아니라, “오답”으로 표시됨에 주의하여라.

예제

예제 입력과 이에 해당하는 함수 호출을 보여준다.

예제 입력	예제 함수 호출	
	기계 1	기계 2
	Encode(...)	
6 5	Code(0,0)	
4 1	Code(2,4)	
0 3	Code(4,16)	
4 5	Code(1,1)	
3 2	Code(3,9)	
3 4	Code(5,25)	
2 4 2		InitDevice()
1 0 0		Answer(4,16)
5 1 2		Answer(1,0)
5 3 0		Answer(25,1)
4 1 1		Answer(25,9)
		Answer(16,1)

여기서 Encode(...) 호출의 인자들은 다음과 같다.

인자	Encode(...)
N	6
A	{4, 0, 4, 3, 3}
B	{1, 3, 5, 2, 4}

문제 3. 용2

입력 파일: standard input
출력 파일: standard output
시간 제한: 3초
메모리 제한: 256MB

JOI 평원에서 사람들은 용과 함께 살고 있다.

JOI 평원은 광활한 좌표평면이고, 직각으로 교차하는 X축과 Y축이 설정되어 있다. X 좌표가 x , Y 좌표가 y 인 점을 (x, y) 로 표시한다.

JOI 평원은 N 마리의 용이 생활하고 있고 1번부터 N 번까지 번호가 붙어있다. 또한 용은 M 종류의 종족이 있어서 1번부터 M 번까지 번호가 붙어있다. i 번 ($1 \leq i \leq N$) 용은 평소에는 JOI 평원의 (A_i, B_i) 에 살고 있고, C_i 번 종족이다. JOI 평원에는 모든 종족의 용이 생활하고 있지 않을 수도 있다.

JOI 평원에 사람이 사는 두 마을은 (D_1, E_1) 과 (D_2, E_2) 에 있다. 두 도시는 도로로 연결되어 있으며, 이는 두 점을 잇는 선분이다.

점 $(A_1, B_1), \dots, (A_N, B_N)$ 과 $(D_1, E_1), (D_2, E_2)$ 는 서로 다르며, 어떠한 세 점도 일직선 위에 있지 않다.

가끔 용의 종족 사이에서 대립이 벌어진다. a 번 ($1 \leq a \leq M$) 종족이 b 번 ($1 \leq b \leq M, a \neq b$) 종족에 대해 적의를 가지면 a 번 종족의 모든 용이 b 번 종족의 모든 용을 향해 화염구를 뿜는다. 화염구는 일직선으로 날아가고 맞은 이후에도 계속 날아간다. 즉, 화염구의 궤적은 반직선이다.

종족 사이의 대립이 일어났을 때 도로와 화염구의 궤적이 교차하면 도로는 손상을 입을 것이다. 일어날 수 있는 Q 개의 대립이 주어졌을 때, 각각의 대립에 대해서 도로와 교차하는 화염구의 개수를 알고 싶다.

입력 형식

다음 정보가 표준 입력으로 주어진다.

- 첫째 줄에는 공백으로 구분된 두 정수 N, M 이 주어진다. 이는 N 마리의 용이 JOI 평원에 살고 있고, M 종류의 종족이 존재한다는 의미이다.
- 다음 N 개의 줄의 i 번째 ($1 \leq i \leq N$) 줄에는 공백으로 구분된 세 정수 A_i, B_i, C_i 가 주어진다. 이는 i 번 ($1 \leq i \leq N$) 용이 (A_i, B_i) 에 살고 있고, C_i 번 종족이라는 의미이다.
- 다음 줄에는 공백으로 구분된 네 정수 D_1, E_1, D_2, E_2 가 존재한다. 이는 사람이 사는 두 마을이 $(D_1, E_1), (D_2, E_2)$ 라는 의미이다.
- 다음 줄에는 정수 Q 가 주어진다. 이는 일어날 수 있는 대립의 개수가 Q 개라는 의미이다.
- 다음 Q 개의 줄의 j 번째 ($1 \leq j \leq Q$) 줄에는 공백으로 구분된 두 정수 F_j, G_j 가 주어진다. 이는 가능한 j 번째 대립이 F_j 번 종족이 G_j 번 종족에게 적의를 품는다는 의미이다.

출력 형식

표준 출력으로 Q 개의 줄을 출력하여라. j 번째 ($1 \leq j \leq Q$) 줄은 j 번째 대립이 일어났을 때 도로와 교차하는 화염구의 개수여야 한다.

제한

- $2 \leq N \leq 30\,000$.
- $2 \leq M \leq N$.
- $-1\,000\,000\,000 \leq A_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).
- $-1\,000\,000\,000 \leq B_i \leq 1\,000\,000\,000$ ($1 \leq i \leq N$).

- $1 \leq C_i \leq M$ ($1 \leq i \leq N$).
- $-1\,000\,000\,000 \leq D_1 \leq 1\,000\,000\,000$.
- $-1\,000\,000\,000 \leq E_1 \leq 1\,000\,000\,000$.
- $-1\,000\,000\,000 \leq D_2 \leq 1\,000\,000\,000$.
- $-1\,000\,000\,000 \leq E_2 \leq 1\,000\,000\,000$.
- $N+2$ 개의 점 $(A_1, B_1), \dots, (A_N, B_N), (D_1, E_1), (D_2, E_2)$ 는 서로 다르며, 어떠한 세 점도 일직선 위에 있지 않다.
- $1 \leq Q \leq 100\,000$.
- $1 \leq F_j \leq M$ ($1 \leq j \leq Q$).
- $1 \leq G_j \leq M$ ($1 \leq j \leq Q$).
- $F_j \neq G_j$ ($1 \leq j \leq Q$).
- $(F_j, G_j) \neq (F_k, G_k)$ ($1 \leq j < k \leq Q$).

서브태스크 1 (15 점)

- $N \leq 3\,000$

서브태스크 2 (45 점)

- $Q \leq 100$

서브태스크 3 (40 점)

추가 제한조건이 없다.

예제

standard input	standard output
4 2	1
0 1 1	2
0 -1 1	
1 2 2	
-6 1 2	
-2 0 2 0	
2	
1 2	
2 1	

첫 번째 종족간의 대립에서, 다음을 만족한다.

- 1번 용이 3번 용에게 발사한 화염구는 도로와 교차하지 않는다.
- 1번 용이 4번 용에게 발사한 화염구는 도로와 교차하지 않는다.
- 2번 용이 3번 용에게 발사한 화염구는 도로와 교차한다.
- 1번 용이 4번 용에게 발사한 화염구는 도로와 교차하지 않는다.

그러므로 하나의 화염구가 도로를 교차한다.

두 번째 종족간의 대립에서, 다음을 만족한다.

- 3번 용이 1번 용에게 발사한 화염구는 도로와 교차한다.
- 3번 용이 2번 용에게 발사한 화염구는 도로와 교차한다.
- 4번 용이 1번 용에게 발사한 화염구는 도로와 교차하지 않는다.
- 4번 용이 2번 용에게 발사한 화염구는 도로와 교차하지 않는다.

그러므로 두 개의 화염구가 도로를 교차한다.

standard input	standard output
3 2 -1000000000 -1 1 -999999998 -1 1 0 0 2 999999997 1 999999999 1 1 1 2	1
6 3 2 -1 1 1 0 1 0 3 2 2 4 2 5 4 3 3 9 3 0 0 3 3 6 1 2 1 3 2 1 2 3 3 1 3 2	4 2 4 0 2 1