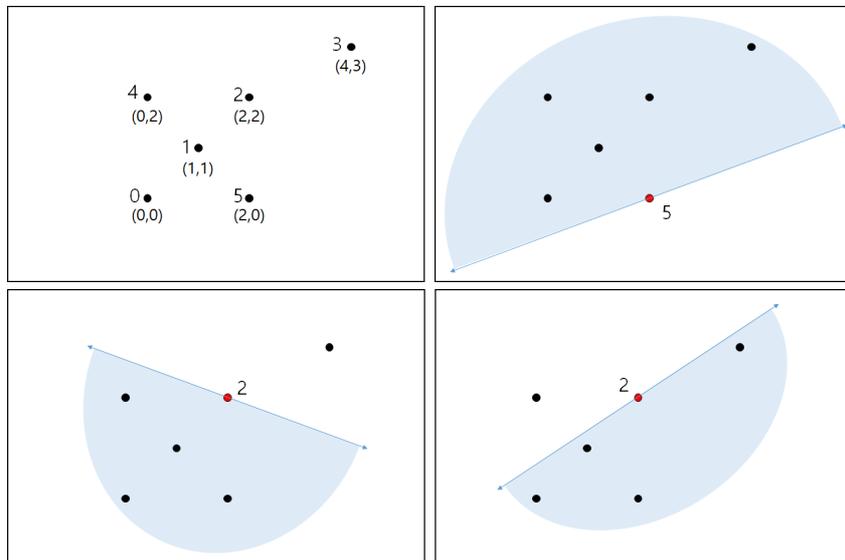


문제2 : 염소

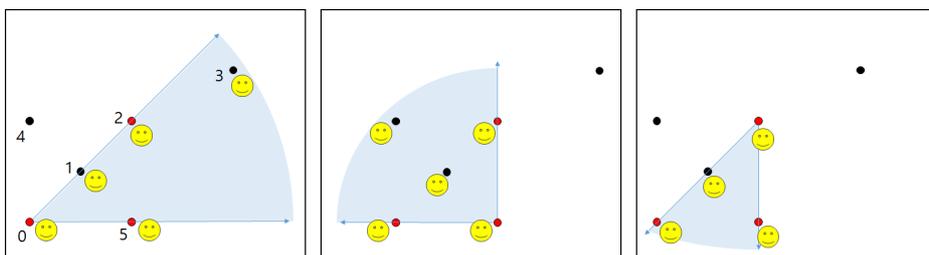
넓은 평원에 N 마리의 염소들이 흩어져서 한가로이 풀을 뜯어먹고 있다. 한 자리에 두 마리 이상의 염소가 있는 경우는 없다. 염소들은 게을러서 자리를 옮기지는 않고 가끔 제자리에서 돌기만 한다. 사실 이 염소들은 모두 로봇으로 걷지는 못하고 제자리에서 회전만 가능한 것이었다. 또한 염소들의 코는 LED등으로 원격에서 점등이 가능하며, 염소들의 눈은 고성능 카메라로 카메라에 담기는 영상은 실시간으로 원격에서 확인할 수 있고 원하는 장면을 저장할 수도 있다. 염소의 눈에는 자신의 코도 보이며, 투시 기능도 있어 염소들이 겹쳐서 보이더라도 몇 마리가 있는지 파악이 가능하다. 또한 눈의 좌우 시야각은 180° 로 180° 경계 상에 있는 것까지 눈에 보인다.

염소들의 위치에 따라 한번에 N 마리를 볼 수 있는 염소도 있고, 경우에 따라 어떤 방향을 보더라도 한번에 N 마리를 볼 수 없는 염소도 있을 수 있다. 예를 들어, 아래 그림과 같이 0번부터 5번까지 여섯 마리의 염소가 주어질 때, 5번 염소는 한번에 여섯 마리를 볼 수 있지만 2번 염소는 어떤 방향을 보더라도 한번에 여섯 마리를 모두 볼 수는 없다.



특별한 실험을 위해 한번에 N 마리를 볼 수 있는 서로 다른 두 염소와 함께 나머지 모든 염소들 중 임의로 한 염소를 선택한다. 선택된 세 염소는 코를 점등하여 어디서나 위치를 확인할 수 있도록 한다. 선택된 세 염소의 눈에 담기는 영상을 지켜보다가 점등된 세 염소가 한 화면에 보이는 장면을 각 염소마다 하나씩 저장한다. 저장된 세 장면을 하나씩 보며, 점등된 세 염소와 함께 점등된 다른 두 염소 사이(경계 포함)에 있는 모든 염소에 특별한 표식을 한다. 이 표식은 특별한 실험을 위한 것으로, N 마리 염소 중 세 번 모두 표식이 된 염소들의 마릿수를 구하는 것이 이 실험의 목표이다.

예를 들어, 전체 여섯 마리 염소 중 선택된 세 염소를 붉은 점으로 표시한 경우, 아래 그림은 각 염소를 기준으로 특별한 표식이 된 염소를 나타낸다.(물론 각 그림은 저장된 장면 자체는 아니다.) 이 경우 세 번 모두 표식이 된 염소들의 마릿수는 4이다.



여러분은 특별한 실험을 위해 다음 2가지 함수를 구현해야만 한다.

- `void init(int x[], int y[])` ; 최초에 호출되며 오직 한번만 호출되는 함수이다. x 와 y 는 크기 N 인 배열(vector)이다. 요소 위치의 x 좌표를 나타내는 $x[0..N-1]$, y 좌표를 나타내는 $y[0..N-1]$ 이 인자로 주어진다. i ($0 \leq i \leq N-1$)번 요소의 위치는 $(x[i], y[i])$ 이다.
- `int count(int a, int b, int c)` ; 주어진 세 요소의 좌표 $(x[a], y[a])$, $(x[b], y[b])$, $(x[c], y[c])$ 를 이용하여 세 번 표식이 되는 요소들의 마릿수를 구하여 return 한다.

구현 세부사항

여러분은 `goat.cpp`라는 이름을 가진 하나의 파일을 제출해야만 한다. 이 파일에는 다음의 함수들이 구현되어 있어야 한다.

- `void init(int x[], int y[])` ;
- `int count(int a, int b, int c)` ;

이 함수들은 위에서 설명한 것과 같이 동작하여야 한다. 물론, 다른 함수들을 만들어서 내부적으로 사용할 수 있다. 제출한 코드는 입출력을 수행하거나 다른 파일에 접근하여서는 안된다.

grader 예시

주어지는 grader는 다음과 같은 형식으로 입력을 읽는다.

- line 1: $N Q$
(N : 요소의 수, Q : 질의(Query)의 개수)
- 다음 N 개의 줄 각각: $x y$
(요소 위치의 x 좌표와 y 좌표)
- 다음 Q 개의 줄 각각: $a b c$
(세 요소의 번호 a, b, c , $0 \leq a, b, c \leq N-1$)

주어지는 grader는 매 질의마다 조건을 만족하는 요소들의 마릿수를 줄로 구분해서 출력한다.

제약 조건

- $3 \leq N \leq 3,000$
- $1 \leq Q \leq 5 \times 10^6$
- 모든 요소의 x 좌표와 y 좌표는 0 이상 10^9 이하

서브태스크 1 [5 points]

- $N \leq 500$, $Q \leq 10^5$
- 어떤 세 점도 한 직선 위에 있지 않다.

서브태스크 2 [17 points]

- $N \leq 500$, $Q \leq 10^5$

서브태스크 3 [10 points]

- $N \leq 500$
- 어떤 세 점도 한 직선 위에 있지 않다.

서브태스크 4 [8 points]

- $N \leq 500$

서브태스크 5 [29 points]

- 어떤 세 점도 한 직선 위에 있지 않다.

서브태스크 6 [31 points]

- 추가 제한이 없다.

[입력 예 1]

```
6 3
0 0
1 1
2 2
4 3
0 2
2 0
0 5 2
1 3 4
4 1 5
```

[출력 예 1]

```
4
4
3
```

아래는 예 1에 대해 함수 호출 및 그 결과를 차례대로 보여준다.

함수호출	결과 값
<code>init({0, 1, 2, 4, 0, 2}, {0, 1, 2, 3, 2, 0})</code>	
<code>count(0, 5, 2)</code>	4
<code>count(1, 3, 4)</code>	4
<code>count(4, 1, 5)</code>	3