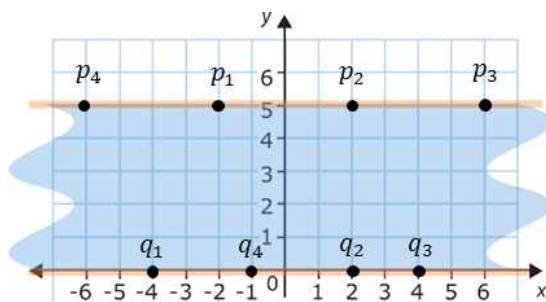


## 센서 네트워크

우리가 사는 지역을 가로 질러 흐르는 강의 생태 환경을 모니터링하기 위하여 무선 센서 네트워크(wireless sensor network)를 구축하고 있다. 무선 센서 네트워크는 무선 통신이 가능한 센서들로 구성된 네트워크를 말한다. 센서는 모두 강의 양쪽 강둑에 설치되어 있다. 그리고 두 센서 사이의 거리가  $r$  이하이면 서로 통신할 수 있고  $r$ 을 초과하면 통신할 수 없다. 여기서 거리는 유클리드 거리를 말한다.<sup>1)</sup> 각 센서가 강 건너에 있는 몇 개의 센서와 통신할 수 있는가를 알아보는 것은 센서 네트워크를 설계할 때 고려하는 흥미로운 문제이다.

오늘 여러분들이 할 일은 바로 이 문제에 대한 프로그램을 작성하는 것이다. 여기서 우리는, 아래 그림에 보인 것처럼, 지역을 가로 지르는 강을  $x, y$ -평면상의 두 직선  $y=0$ 과  $y=d$ 로 둘러싸인 띠(strip) 모양의 영역으로 모델한다. 물론  $d$ 는 강폭을 의미한다. 강둑에 위치한 센서는, 남쪽 강둑에 있는 것은 직선  $y=0$  위에 그리고 북쪽 강둑에 있는 것인  $y=d$  위에 위치한 점으로 나타낸다. 센서의 크기를 무시하고 있음에 유의한다.



위의 그림에서 강폭  $d=5$ 이다. 남쪽 강둑에 센서 넷  $q_1, q_2, q_3, q_4$ 가 그리고 북쪽 강둑에도 센서 넷  $p_1, p_2, p_3, p_4$ 가 그림처럼 위치하고, 통신 반경  $r=7$ 이라고 하자. 그러면  $q_1$ 과  $p_4$

사이의 거리  $d(q_1, p_4) = \sqrt{2^2 + 5^2} \leq 7$ 이므로  $q_1$ 과  $p_4$ 는 통신이 가능하다. 또한  $q_1$ 은  $p_1$ 과도 통신이 가능하지만,  $d(q_1, p_2) = \sqrt{6^2 + 5^2} > 7$ 이고  $d(q_1, p_3) = \sqrt{10^2 + 5^2} > 7$ 이므로  $q_1$ 은  $p_2, p_3$ 와는 통신할 수 없다.

강폭  $d$ , 통신 반경  $r$ , 그리고 북쪽 강둑에 설치된 센서  $n$ 개와 남쪽 강둑에 설치된 센서  $n$ 개의 위치가 주어질 때, 남쪽 강둑에 설치된 각 센서에 대하여 북쪽 강둑에 설치된 몇 개의 센서와 통신할 수 있는지를 계산하는 프로그램을 작성하시오. 프로그램 이름은 sensor.cpp(c), 설명 파일의 이름은 sensor.pdf로 하고, 프로그램의 실행시간은 1.0초를 초과할 수 없다. 부분 점수는 없다.

### 입력 형식

표준 입력을 통하여 입력한다. 첫째 줄에 강폭과 통신 반경을 나타내는 양의 정수  $d, r$ , 그리고 양쪽 강둑 각각에 설치된 센서의 개수를 나타내는 양의 정수  $n$ 이 입력으로 주어진다. 이때,  $d \leq 20,000$ ,  $r \leq 50,000$ 이고  $n \leq 10^5$ 이다. 둘째 줄에는 북쪽 강둑에 설치된 센서의  $x$ -좌표를 나타내는  $n$ 개의 정수가 주어지고, 마찬가지로 셋째 줄에는 남쪽 강둑에 설치된 센서의  $x$ -좌표를 나타내는  $n$ 개의 정수가 주어진다. 센서의  $x$ -좌표는  $-20,000$  이상  $20,000$  이하이다. 같은 곳에 둘 이상의 센서가 위치할 수 있음에 유의한다.

### 출력 형식

표준 출력을 통하여 출력한다. 남쪽 강둑에 설치된 센서 각각에 대하여 그것과 통신할 수 있으면서 북쪽 강둑에 설치되어 있는 센서의 개수를 모두 구한 다음, 이  $n$ 개의 정수를 비내림차순(nondecreasing)으로 정렬하여 한 줄에 출력한다.<sup>2)</sup> 출력하는 정수들 사이에 빈 칸을 정확히 하나 둔다.

1)  $x, y$ -평면 상의 두 점  $(x_0, y_0)$ 와  $(x_1, y_1)$  사이의 유클리드 거리는  $\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}$ 이다.

2) 비내림차순은 크기가 작은 것이 큰 것보다 앞서는 순서를 말한다.

## 입력과 출력의 예 (1)

입력

```
5 7 4
-2 2 6 -6
-4 2 4 -1
```

출력

```
2 2 2 3
```

## 입력과 출력의 예 (2)

입력

```
5 7 4
-2 2 6 -6
100 -100 0 -2
```

출력

```
0 0 2 3
```

## 입력과 출력의 예 (3)

입력

```
1 3 5
4 4 5 1 2
-5 1 2 0 3
```

출력

```
0 2 2 4 5
```

주의. 출력은 음이 아닌 정수  $n$ 개와 빈칸  $n-1$ 개로 구성된다.

프로그래밍은 즐거워 !!!