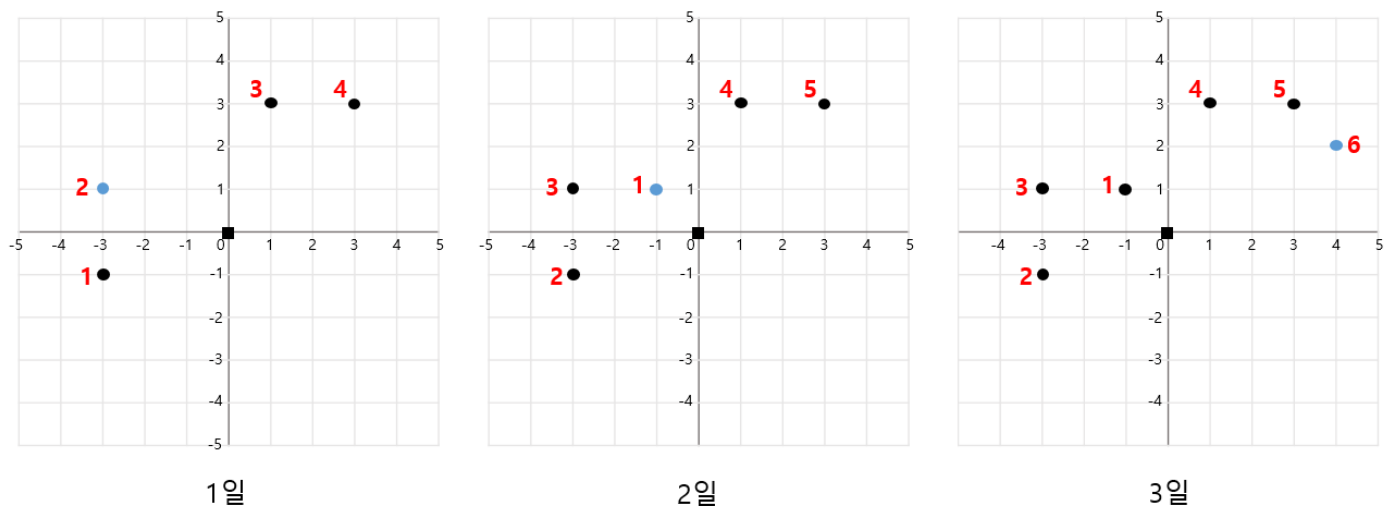


햄버거 가게

김인하는 햄버거를 좋아해서, 매일 점심식사로 햄버거를 먹는다. 인하대 근처에는 햄버거 가게가 M 개가 존재하고, 매일 아침 한 개의 새로운 햄버거 가게가 생성된다. 인하대학교에서 가까운 햄버거 가게는 늘 붐비기 때문에, 김인하는 매일 인하대로부터 K 번째로 가까운 햄버거 가게를 방문하기로 했다. N 일에 걸쳐 매일 아침에 새로운 햄버거 가게가 생성되므로, 1일에는 $M + 1$ 개의 햄버거 가게 중 K 번째로 가까운 햄버거 가게를 방문해야 하고, N 일에는 $M + N$ 개의 햄버거 가게 중 K 번째로 가까운 햄버거 가게를 방문해야 한다.

햄버거 가게들의 위치는 인하대학교를 원점(0,0)으로 하여 2차원 좌표 (x,y) 로 표현된다. 인하대학교에서 햄버거 가게까지의 거리는 $\sqrt{x^2 + y^2}$ 이며, 만약 거리가 같은 햄버거 가게가 여러 개 존재할 경우, x 좌표가 더 작은 가게가 가까운 가게이고, 거리와 x 좌표 모두 같은 두 가게는 y 좌표가 더 작은 가게가 더 가까운 가게라 하자. [그림 1]은 $M = 3, N = 3, K = 3$ 인 예시이다. 초기에는 $(-3,-1), (1,3), (3,3)$ 의 위치에 햄버거 가게가 존재했다. 1일에 $(-3,1)$ 의 위치에 가게가 생성되었고, 정의에 따라 $(1,3)$ 의 가게가 인하대학교로부터 K 번째로 가깝다. 2일에는 $(-1,1)$ 의 위치에 가게가 생성되었고, $(-3,1)$ 의 가게가 인하대학교로부터 K 번째로 가깝다. 3일에는 $(4,2)$ 의 위치에 가게가 생성되었고, $(-3,1)$ 의 가게가 인하대학교로부터 K 번째로 가깝다.

햄버거 가게들의 정보가 주어졌을 때, 1일부터 N 일까지 인하대학교로부터 K 번째로 가까운 햄버거 가게의 좌표를 출력하는 프로그램을 작성하시오.



[그림 1] $M = 3, N = 3, K = 3$ 일 때, 일별 햄버거 가게 예시

※ 프로그램의 실행 시간은 3 초, 메모리 사용량은 512MB 를 초과할 수 없다.

사용할 수 있는 언어는 C, C++로 제한한다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin, cout 의 입출력 속도를 개선할 수 있다.

```
ios_base::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);
```

단, 위의 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력을 혼용해서 사용하면 안된다. C++의 std::endl 의 경우 출력 속도가 느리므로, cout<<endl; 대신 cout<<"\n";을 사용하는 것을 권장한다.

입력

첫 번째 줄에 테스트 케이스 수 T ($1 \leq T \leq 30$)가 주어진다.

각 테스트 케이스의 구성은 다음과 같다.

- 첫 번째 줄에는 처음 존재하는 햄버거 가게의 수 M ($1 \leq M \leq 200,000$), 새롭게 생성되는 햄버거 가게의 수 N ($1 \leq N \leq 200,000$), 자연수 K ($1 \leq K \leq M$)가 주어진다.
- 이후 M 개의 줄에 걸쳐 기존에 존재하는 햄버거 가게의 좌표 정보 x 와 y ($-10,000 \leq x, y \leq 10,000$)가 공백으로 구분되어 주어진다.
- 이후 N 개의 줄에 걸쳐 새롭게 생성되는 햄버거 가게의 좌표 정보 x 와 y ($-10,000 \leq x, y \leq 10,000$)가 주어진다.

단, 모든 햄버거 가게의 좌표는 중복되지 않는다.

출력

각 테스트 케이스마다 N 개 줄에 걸쳐, 각 날짜마다 K 번째로 가까운 햄버거 가게의 좌표 x 와 y 를 공백으로 구분하여 출력한다.

예제 입출력

| 예제 입력 | 예제 출력 |
|-------|-------|
| 3 | 1 3 |
| 3 3 3 | -3 1 |
| -3 -1 | -3 1 |
| 1 3 | -3 -1 |
| 3 3 | -1 1 |
| -3 1 | -1 1 |
| -1 1 | 3 4 |
| 4 2 | 0 5 |
| 3 3 1 | -4 3 |
| -3 -1 | -4 3 |
| 1 3 | -5 0 |
| 3 3 | |
| -3 1 | |
| -1 1 | |
| 4 2 | |
| 3 5 3 | |
| 3 4 | |
| 0 5 | |

| | |
|------|--|
| -5 0 | |
| 6 -6 | |
| 2 3 | |
| -4 3 | |
| 7 8 | |
| 0 1 | |