

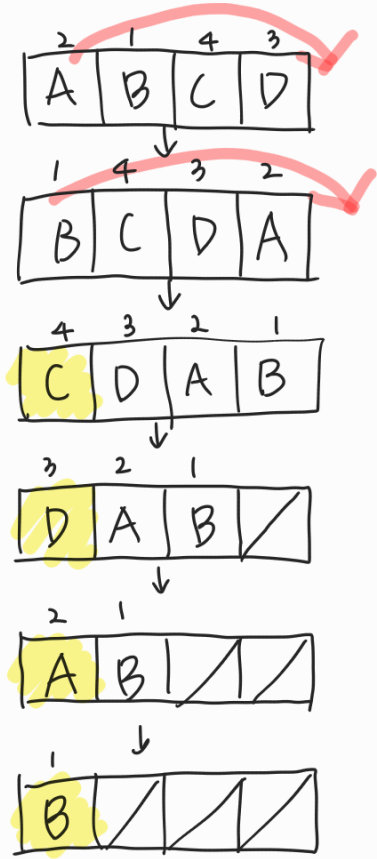
1966번. 프린터 큐

다음 조건에 따라 인쇄된다.

1. 현재 Queue의 가장 앞에 있는 문서의 중요도를 확인한다.
2. 나머지 문서들 중 현재 문서보다 중요도와 높은 문서가 하나라도 있으면 이 문서를 인쇄하지 않고 Queue의 가장 뒤에 재배치한다.
그렇지 않다면 바로 인쇄한다.

예) 4개의 문서 (A, B, C, D)의 중요도와

(2, 1, 4, 3) 이라면, C-D-A-B 순서대로 인쇄한다.



문서가 몇번재로 인쇄되었는지 알아내자.

[Input]

```

3
1 0
5
4 2
1 2 3 4
6 0
1 1 9 1 1 1
    
```

[Output]

```

1
2
5
    
```

예1) 1 0
5
답: 1

예2) 4 2
1 2 3 4
답: 2

예3) 6 0
1 1 9 1 1 1
1 9 1 1 1 1
9 1 1 1 1 1
1 1
[9, 1, 1, 1, 1]
[9, 1, 1, 1, 1]
답: 5



Solution code

```

public static void solution (LinkedList<int> q, int M){
    int count=0;
    while ( ! q.isEmpty()){
        int[] front = q.poll(); Queue의 첫 번째 원소
        boolean isMax = true;

        for (int i=0; i<q.size(); ++i){
            if (front[i] < q.get(i)[0]){ 더 중요한 원소가 있는 경우
                q.offer(front); Queue의 맨 뒤에 원소를 넣는다.
                for (int j=0; j<i; ++j){ 큰 원소를 발견했다. (=i)
                    q.offer(q.poll()); i 이전의 원소들도 뒤로 보낸다.
                }
                isMax = false; 가장 큰 원소가 아니었기 때문에 false로 변경
                break; 종료
            }
        }

        if ( ! isMax){
            continue;
        }

        count++;
        if (front[0] == M){ 가장 큰 원소라서 꺼내질 때
            break; 반복문 종료
        }
    }

    sb.append(count).append("\n");
}

```