숫자 카드 2

https://www.acmicpc.net/problem/10816

- 입력 받은 숫자들을 정렬 한 후 binary search를 이용하여 찾는다.
- 이 문제의 시간 복잡도는 정렬에 필요한 O(NlogN)의 시간 복잡도를 가진다.
- 이 문제는 동일 값이 몇 개이었는지 카운팅 해야 하므로 lower_bound와 upper_bound 함수를 사용한다.
- lower_bound(시작 이터레이터, 끝 이터레이터, 찾을 값)
  : 찾을 값 / 찾을 값 보다 가장 가까운 큰값
- upper_bound(시작 이터레이터, 끝 이터레이터, 찾을 값)
  : 찾을 값 보다 가장 가까운 큰값

중복 값 존재 시 lower_bound는 값 중 가장 첫번째 원소를 가리키고
upper_bound는 찾을 값보다 큰 수 중 가장 작은 수를 가리킨다.
따라서 upper_bound – lower_bound를 통해 중복 값의 개수를 알 수 있다.

```
sort(a.begin(), a.end());
for (int i = 0; i < M; ++i) {
    printf("%d ", upper_bound(a.begin(), a.end(), b[i])
        - lower_bound(a.begin(), a.end(), b[i]));
}
```