

프로그래밍 과제 02

1. [2-SUM] 오름차순으로 정렬된 N 개의 서로 다른 정수 $a_0 < a_1 < \dots < a_{N-1}$ 과 또 다른 하나의 정수 K 가 입력으로 주어진다. 이때 N 개의 정수들 중에서 합이 K 가 되는 정수 쌍의 개수를 카운트하라. 하나의 정수를 2번 중복해서 뽑는 것은 허용되지 않는다. 이 문제는 다음과 같은 알고리즘으로 풀 수 있다.

```
int data[MAX];
int N, K;

/* N개의 정수들과 K값을 입력받는다. 정수들은 정렬된 상태로 입력된다.*/

int start = 0, end = N-1;
int count = 0;
while(start < end) {
    int sum = data[start] + data[end];
    if (sum > K)
        end--;
    else if (sum < K)
        start++;
    else {
        count++;
        start++;
        end--;
    }
}
printf("%d\n", count);
```

이 문제를 푸는 프로그램을 순환함수(recursion)로 작성하라. 정수들을 입력받는 부분을 제외하고는 **for**문이나 **while**문과 같은 반복문을 사용해서는 안되며 시간복잡도는 $O(N)$ 이어야 한다. 입력 형식은 다음과 같다. 먼저 N 의 값이 주어지고 이어서 N 개의 정수들이 이미 오름차순으로 정렬된 순서대로 주어진다. 마지막으로 정수 K 의 값이 주어진다.

입력 예	출력
10 1 2 3 4 5 6 7 8 9 10	4
13 25 1 3 6 9 13 17 21 23 24 31 37 38 44 45 47 51 55 58 71 73 88 91 99 101 102	5
102 40 1 17 19 23 25 28 41 44 49 50 61 64 65 67 71 77 79 81 82 83 84 90 91 92 96 99 101 103 109 121 128 132 133 150 152 161 165 167 177 178 199	5

2. 오름차순으로 정렬된 N 개의 정수 $a_0 \leq a_1 \leq \dots \leq a_{N-1}$ 과 또 다른 하나의 정수 K 가 입력으로 주어진다. 정수들 중에서 K 보다 작거나 같으면서 가장 큰 정수를 찾는 함수 **floor**와 K 보다 크거나 같으면서 가장 작은 정수를 찾는 함수 **ceiling**을 각각 순환함수로 구현하라. 두 함수 모두 그러한 정수가 없을 경우 -1 을 반환하라. 1번과 동일한 형식으로 입력을 받아서 배열 **data**에 저장한 후 정수 K 의 **floor**와 **ceiling**을 찾아서 출력하는 프로그램을 작성하라. 두 함수는 별개로 구현해야하며 각각의 시간복잡도는 $O(\log N)$ 이어야 한다.

입력 예	출력
10	10
1 2 3 4 5 6 7 8 9 10	-1
11	
25	
1 3 6 9 13 17 21 23 24 31 37 38 44 45 47	71
51 55 58 71 73 88 91 99 101 102	73
72	
40	
1 17 19 23 25 28 41 44 49 50 61 64 65 67	
71 77 79 81 82 83 84 90 91 92 96 99 101	109
103 109 121 128 132 133 150 152 161 165	121
167 177 178	
111	
2	
2 2	-1
1	2
2	
2 2	2
3	-1