

프로그래밍 과제 01

1. [Rank] 입력으로 $N < 1000$ 개의 정수 a_1, a_2, \dots, a_N 과 또 다른 하나의 정수 K 가 주어진다. 정수들은 정렬되어 있지 않고, 동일한 값이 중복될 수도 있다. 임의의 정수 x 의 rank란 N 개의 정수 a_1, a_2, \dots, a_N 중에서 x 보다 작은 것의 개수 + 1을 말한다. 정수 K 의 rank를 구해서 출력하는 프로그램을 작성하라. 단, 프로그램의 어떤 부분에서도 for문, while 문 등의 반복문을 사용해서는 안된다. 입력의 형식은 먼저 정수의 개수 N 이 주어지고, 이어서 N 개의 정수가 주어진다. 마지막으로 정수 K 가 주어진다. 시간복잡도가 $O(N)$ 을 초과해서는 안된다.

입력 예	출력
10 2 5 3 8 6 7 8 7 2 1	9
8 25 88 1 9 99 24 99 1 45 17 21 23 73 31 37 6 38 9 91 3 99 47 58 55 73 51	16
50 8 1 1 1 1 2 2 2 2 2	5

2. [Rank] 입력으로 $N < 1000$ 개의 정수 a_1, a_2, \dots, a_N 가 주어진다. 정수들은 정렬되어 있지 않고, 동일한 값이 중복될 수도 있다. 임의의 정수 x 의 rank란 N 개의 정수 a_1, a_2, \dots, a_N 중에서 x 보다 작은 것의 개수 + 1을 말한다. 입력으로 주어진 정수들 각각의 rank를 구하여 순서대로 출력하는 프로그램을 작성하라. 단, 프로그램의 어떤 부분에서도 for문, while 문 등의 반복문을 사용해서는 안된다. 입력의 형식은 먼저 정수의 개수 N 이 주어지고, 이어서 N 개의 정수가 주어진다. 시간복잡도가 $O(N^2)$ 을 초과해서는 안된다.

입력 예	출력
10 2 5 3 8 6 7 8 7 2 1	2 5 4 9 6 7 9 7 2 1
25 88 1 9 99 24 99 1 45 17 21 23 73 31 37 6 38 9 91 3 99 47 58 55 73 51	21 1 5 23 10 23 1 14 7 8 9 19 11 12 4 13 5 22 3 23 15 18 17 19 16
8 1 1 1 1 2 2 2 2	1 1 1 1 5 5 5 5

3. [3-SUM] 입력으로 $N < 1000$ 개의 서로 다른 양의 정수 a_1, a_2, \dots, a_N 과 또 다른 하나의 정수 K 가 주어진다. 정수들은 정렬되어 있지 않다. N 개의 정수들 중에서 어떤 세 정수를 선택하여 그 합이 K 가 되는 경우의 수를 카운트하는 프로그램을 작성하라. 하나의 정수를 2번 이상 중복해서 뽑는 것은 허용되지 않는다. 단, 프로그램의 어떤 부분에서도 for문, while 문 등의 반복문을 사용해서는 안된다. 시간복잡도가 $O(N^3)$ 을 초과해서는 안된다.

입력 예	출력
10 9 1 8 2 6 7 5 4 3 10	11
16 25 101 3 9 13 17 55 99 31 102 38 44 6 47 51 24 45 58 71 23 37 73 21 88 91 1	15
102 40 1 17 19 133 23 99 178 25 132 28 150 41 167 44 92 49 50 96 61 152 64 128 65 67 177 101 161 91 71 103 77 109 79 81 82 83 165 121 84 90 199	40

4. [Nearest] 오름차순으로 정렬된 $N < 1000$ 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 와 가장 가까운 정수를 찾는 함수 nearest를 구현하라. 가장 가깝다는 말을 차이의 절대값이 최소라는 의미이다. 답이 2개 이상인 경우에는 그 중 작은 값을 출력해야 한다. 함수 nearest

의 시간복잡도는 $O(\log N)$ 이어야 한다. 정수들을 입력받아서 배열에 저장하는 부분을 제외한 나머지 부분에 서는 for문이나 while문과 같은 반복문을 사용해서는 안된다.

입력 예	출력
10 1 2 3 4 5 6 7 8 9 10	10
11 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	71
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 99 101 103 109 121 128 132 133 150 152 161 165 167 177 178	109
111 2 2 2	2
1 2 2 2 3	2