PL-2020-Football Scores

locked

Problem Submissions Leaderboard Discussions

The number of goals achieved by two football teams in matches in a league is given in the form of two lists. Consider:

• Football team A, has played three matches, and has scored { 1 , 2 , 3 } goals in each match respectively. • Football team B, has played two matches, and has scored { 2, 4 } goals in each match respectively. • Your task is to compute, for each match of team B, the total number of matches of team A, where team A has scored less than or equal to the number of goals scored by team B in that match. • In the above case: • For 2 goals scored by team B in its first match, team A has 2 matches with scores 1 and 2. • For 4 goals scored by team B in its second match, team A has 3 matches with scores 1, 2 and 3.

Hence, the answer: {2, 3}.

Complete the code in the editor below. The program must return an array of m positive integers, one for each maxes[i] representing the total number of elements nums[j] satisfying nums[j] ≤ maxes[i] where 0 ≤ j < n and 0 ≤ i < m, in the given order.

It has the following: nums[nums[0],...nums[n-1]]: first array of positive integers maxes[maxes[0],...maxes[n-1]]: second array of positive integers

Input Format

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n, the number of elements in nums. The next n lines each contain an integer describing nums[j] where 0 ≤ j < n. The next line contains an integer m, the number of elements in maxes. The next m lines each contain an integer describing maxes[i] where 0 ≤ i < m.

Constraints

• 2 ≤ n, m ≤ 10^5 • 1 ≤ nums[j] ≤ 10^9, where 0 ≤ j < n. • 1 ≤ maxes[i] ≤ 10^9, where 0 ≤ i < m.

Output Format

-

Sample Input 0

```
4
1
4
2
2
4
2
3
5
```

Sample Output 0

```
2
4
```

Sample Input 1

```
4
1
4
2
4
2
3
5
```

Sample Output 1

```
2
4
```

Explanation 1

We are given n = 4, nums = [1, 4, 2, 4], m = 2, and maxes = [3, 5]. 1. For maxes[0] = 3, we have 2 elements in nums (nums[0] = 1 and nums[2] = 2) that are ≤ maxes[0]. 2. For maxes[1] = 5, we have 4 elements in nums (nums[0] = 1, nums[1] = 4, nums[2] = 2, and nums[3] = 4) that are ≤ maxes[1].

Thus, the function returns the array [2, 4] as the answer.

Sample Input 2

```
5
2
10
5
4
8
4
3
1
7
8
```

Sample Output 2

```
1
0
3
4
```

Explanation 2

We are given, n = 5, nums = [2, 10, 5, 4, 8], m = 4, and maxes = [3, 1, 7, 8]. 1. For maxes[0] = 3, we have 1 element in nums (nums[0] = 2) that is ≤ maxes[0]. 2. For maxes[1] = 1, there are 0 elements in nums that are ≤ maxes[1]. 3. For maxes[2] = 7, we have 3 elements in nums (nums[0] = 2, nums[2] = 5, and nums[3] = 4) that are ≤ maxes[2]. 4. For maxes[3] = 8, we have 4 elements in nums (nums[0] = 2, nums[2] = 5, nums[3] = 4, and nums[4] = 8) that are ≤ maxes[3].

Thus, the function returns the array [1, 0, 3, 4] as the answer.

```c
#include <string.h>
#include <math.h>
#include <stdlib.h>

int* solve(int n, int* nums, int m, int* maxes) {
  int* result = malloc(sizeof(int) * m);
  for (int i = 0; i < m; i++) {
    int count = 0;
    for (int j = 0; j < n; j++) {
      if (nums[j] <= maxes[i]) {
        count++;
      }
    }
    result[i] = count;
  }
  return result;
}

int main() {
  int n;
  scanf("%d", &n);
  int nums[n];
  for (int i = 0; i < n; i++) {
    scanf("%d", &nums[i]);
  }
  int m;
  scanf("%d", &m);
  int maxes[m];
  for (int i = 0; i < m; i++) {
    scanf("%d", &maxes[i]);
  }
  int* result = solve(n, nums, m, maxes);
  for (int i = 0; i < m; i++) {
    printf("%d\n", result[i]);
  }
  free(result);
  return 0;
}
```

Line: 1 Col: 1

Upload Code as File | Test against custom input | Run Code | Submit Code