DSC-102

GE23131-Programming Using C/2024

Lakshmi Engineer

Exam Mode: Strict Mode (Copy/Paste Disabled)

Exam Mode

Exam 2
Attempt 1
Attempt(s) Left: 1/10
In Progress

Status: Active
Start Time: Sunday, 01 December 2024, 10:00 AM
Completed: Monday, 09 December 2024, 10:00 AM
Duration: 168 days 10 hours

Write a program to print odd numbers in ascending order and even numbers in descending order. Input of the first number (n) from user and then input n numbers (a1, a2, ..., an).

Input Format

1. First line is a number of test cases T. Following T lines contain:
2. N followed by N integers of the array.
3. The last line contains the target S.

Output Format

Print T lines as per the output and NO EXAMPLE.

Sample Input

Input

1
5 5 5 5 5
4

Output

1

Input

1
5 5 5 5 5
2

Output

1

Answer (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{
    int T;
    scanf("%d", &T);
    while (T--)
    {
        int n, i, j, temp, a[100];
        int count = 0, flag = 0;
        int *ptr;
        scanf("%d", &n);
        ptr = a;
        for (i = 0; i < n; i++)
        {
            scanf("%d", *(ptr + i));
        }
        for (i = 0; i < n; i++)
        {
            for (j = i + 1; j < n; j++)
            {
                if (*(ptr + i) == *(ptr + j))
                {
                    count++;
                    *(ptr + j) = -1;
                }
            }
            if (*(ptr + i) == -1)
                continue;
            if (count == 0)
            {
                flag = 1;
                break;
            }
            else
            {
                count = 0;
            }
        }
        if (flag == 1)
            printf("1\n");
        else
            printf("0\n");
    }
}
```

Input Expected Got

Input	Expected	Got
1	1	1
5 5 5 5 5		
4		
1	1	1
5 5 5 5 5		
2		
1	0	0
5 5 5 5 5		
5		
1	1	1
1 2 3 4 5		
5		
1	1	1
1 2 3 4 5		
4		
1	1	1
1 2 3 4 5		
3		
1	1	1
1 2 3 4 5		
1		
1	1	1
1 2 3 4 5		
2		
1	1	1
1 2 3 4 5		
6		
1	1	1
1 2 3 4 5		
7		
1	1	1
1 2 3 4 5		
8		
1	1	1
1 2 3 4 5		
9		
1	1	1
1 2 3 4 5		
10		

Passed all tests! ✓

Exam 2
Attempt 1
Attempt(s) Left: 1/10
In Progress

Given a sorted array and a target value, return the index if the target is found. If not, return the index at which it would be if it were inserted in order.

Input Format

The single line contains an array of integers as a space separated.

The second line contains the value that should be inserted in the array. Note: Assume that the array contains no duplicates. Note: Assume array is sorted. Note: Indices are 0 based.

Constraints

1 <= n <= 100
1 <= a[i] <= 100
1 <= x <= 100

Output Format

For each test case: Print the index at which the element should be inserted in the array. Print a space between the elements.

Sample Input 0

1
1
3
5
6

Sample Output 0

1
3

Explanation

Test Case 0: N = 1
Array: 1 Elements in array: 1 (only single element)
Target: 3 Since, target > array[0], return index 1.

Test Case 1: N = 3
Array: 1 3 5 Elements in array: 1 3 5 (in ascending order)
Target: 6 Since, target > array[2], return index 3.

Sample Input 1

1
3
1 3 5
4

Sample Output 1

2

Explanation

Test Case 0: N = 3
Array: 1 3 5 Elements in array: 1 3 5 (in ascending order)
Target: 4 Since, array[1] < target < array[2], return index 2.

Answer (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{
    int n, i, j, a[100], x, temp;
    int *ptr;
    scanf("%d", &n);
    ptr = a;
    for (i = 0; i < n; i++)
    {
        scanf("%d", *(ptr + i));
    }
    scanf("%d", &x);
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (*(ptr + i) > *(ptr + j))
            {
                temp = *(ptr + i);
                *(ptr + i) = *(ptr + j);
                *(ptr + j) = temp;
            }
        }
    }
    for (i = 0; i < n; i++)
    {
        if (*(ptr + i) >= x)
            break;
    }
    printf("%d", i);
}
```

Input Expected Got

Input	Expected	Got
1	1	1
3		
1 3 5		
6		
1	1	1
3		
1 3 5		
4		
1	2	2
3		
1 3 5		
2		
1	0	0
3		
1 3 5		
0		
1	3	3
3		
1 3 5		
7		
1	1	1
3		
1 3 5		
3		
1	1	1
3		
1 3 5		
1		
1	1	1
3		
1 3 5		
8		
1	1	1
3		
1 3 5		
9		
1	1	1
3		
1 3 5		
10		

Passed all tests! ✓

Exam 2
Attempt 1
Attempt(s) Left: 1/10
In Progress

The number of spaces, delimited by a line feed, that follow a number in the input are equal to the greatest common factor of that number and 100.

Input Format

Each line of input, except the last one, contains a number (1 <= n <= 10^9). Input ends with a line containing 0.

Output Format

Output the number of spaces that follow each number in the input. Output ends with a line containing 0. Note: Input ends with a line containing 0. Output ends with a line containing 0.

Sample Input 0

2
3
0

Sample Output 0

2
1
0

Explanation 0

Input gives inputs -1, 4, 0, 2, 3, 0. Output gives 1, 2, 1, 0.

1. For input 2 (2 > 0), output is 2 (the only common factor of 2 and 100).
2. For input 3 (3 > 0), output is 1 (the only common factor of 3 and 100).
3. For input 0, 0 < 0 for input 0, so the output is 0.

Thus, the function returns the values 2, 1, 0 as the output.

Sample Input 1

2
3
0

Sample Output 1

2
1
0

Explanation 1

Input gives inputs -1, 4, 0, 2, 3, 0. Output gives 1, 2, 1, 0.

1. For input 2 (2 > 0), output is 2 (the only common factor of 2 and 100).
2. For input 3 (3 > 0), output is 1 (the only common factor of 3 and 100).
3. For input 0, 0 < 0 for input 0, so the output is 0.

Thus, the function returns the values 2, 1, 0 as the output.

Answer (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{
    int n, i, j, k, a[100], temp;
    int *ptr;
    int count = 0, flag = 0;
    scanf("%d", &n);
    ptr = a;
    for (i = 0; i < n; i++)
    {
        scanf("%d", *(ptr + i));
    }
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (*(ptr + i) == *(ptr + j))
            {
                count++;
                *(ptr + j) = -1;
                flag = 1;
                break;
            }
            else
            {
                count = 0;
                flag = 0;
                break;
            }
        }
        if (flag == 1)
            printf("%d", count);
        else
            printf("%d", count);
        count = 0;
        flag = 0;
        if (*(ptr + i) == -1)
            continue;
        if (count == 0
```