

Ahsania Mission University of Science & Technology

Lab Report

Lab No: 04

Course Code: CSE 2202

Course Title: Computer Algorithm Sessional.

Submitted By:

Mt.Sumaiya Ajmeri Borsha
ID: 1012320005101020
1st Batch, 2nd Year, 2nd Semester
Department of Computer science and Engineering,
Ahsania Mission University of Science & Technology

Submitted To:

Md. Fahim Faisal
Lecturer,
Department of Computer science and Engineering,
Ahsania Mission University of Science & Technology

Task No.: 01

Problem Statement: Sam Height <HSBC>

Sam among his friends wants to go to watch a movie in Armstord Cinema. There is something special about Armstord cinema whenever people come in the group here. They will get seats accordingly their heights. Sam as a curious guy always wants to sit in the middle as cinema has the best view from the middle. Now, Sam as the leader of his group decides who will join him for the movie. Initially, he has $N-1$ friends with him (N including him).

You are given $N-1$ numbers that represent the heights of Sam's friends. You are given the height of Sam as well. Now, Sam can do two operations:

1. He can call a new friend of height H .
2. He can cancel any of his friend invitations.

Each operation will cost him a unit time.

He wants to do this as soon as possible.

Input format

- The first line contains T , where T is the test case.
- Each test case contains two lines,
 - The first line contains two space-separated integer N, S where N is the total number of Sam's friend and S is Sam height.
 - The second line contains N space-separated integers that represent the height of Sam's friend.

Output format

Print the required answer for each test case in a new line.

Constraints

$$1 \leq T \leq 100$$

$$1 \leq N \leq 105$$

$$1 \leq A_r[i] \leq 109$$

Note:

- Sam should always sit in middle and there's an equal number of persons in his left and right.
- Height of Sam is always unique.

Sample Input

2
3 2
4 3 1
1 5
6

Sample Output

1
1

In first test case :

We can cancel invitation of person of height 4 (Cost = 1)

In second Test Case:

We can invite person with height 4 (Cost =1)

Source Code:

```
#include<iostream>
using namespace std;
void merge(int arr[], int left, int mid, int right)
{
    int n1 = mid - left + 1;
    int n2 = right - mid;
    int L[n1], R[n2];
    for (int i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (int j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];
    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2)
    {
        if (L[i] <= R[j])
        {
            arr[k++] = L[i++];
        }
        else
        {
            arr[k++] = R[j++];
        }
    }
}
```

```

    }
}
while (i < n1)
    arr[k++] = L[i++];

while (j < n2)
    arr[k++] = R[j++];
}

void mergeSort(int arr[], int left, int right)
{
    if (left < right)
    {
        int mid = left + (right - left) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
        merge(arr, left, mid, right);
    }
}

int In_Out(int arr[],int L, int H, int Sh)
{
    int i;
    for(i=L; i<H; i++)
    {
        if(arr[i]==Sh)
            break;
    }
    if(i==(H/2))
        return 0;
    else if(i<(H/2))
    {
        return ((H/2)-i);
    }
    else if(i>(H/2))
    {
        return (i-(H/2));
    }
}

int main()
{
    int t;

```

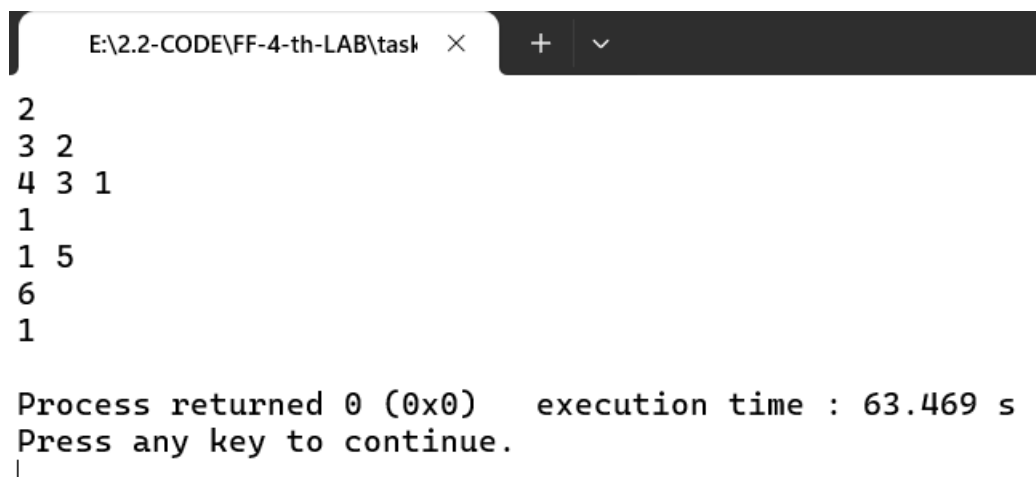
```

cin>>t;
while(t--)
{
    int Fn,Sh;
    cin>>Fn>>Sh;
    if(Fn==1)
    {
        int M;
        cin>>M;
        cout<<Fn<<endl;
        continue;
    }
    int arr[Fn];
    for(int i=0; i<Fn; i++)
    {
        cin>>arr[i];
    }
    arr[Fn]=Sh;
    mergeSort(arr,0,Fn);
    cout<<In_Out(arr, 0, Fn+1, Sh)<<endl;

}
}

```

Output:



```

E:\2.2-CODE\FF-4-th-LAB\task × + ▾
2
3 2
4 3 1
1
1 5
6
1

Process returned 0 (0x0)   execution time : 63.469 s
Press any key to continue.
|

```