GeeksMan

We code therefore
we are

CODE FOUNDATION

Passing array to functions:

C++ does not allow to pass an entire array as an argument to a function. However, You can pass a pointer to an array by specifying the array's name without an index. If you want to pass a single-dimension array as an argument in a function, you would have to declare the function’s formal parameter in one of the following three ways and all three declaration methods produce similar results because each tells the compiler that an integer pointer is going to be received.

Way 1:

```
void myFunction(int *param) {
    ..
    ..
    ..
}
```

Here the formal parameter is a pointer.

Way 2:

```
void myFunction(int param[10]) {
    ..
    ..
    ..
}
```

Here the formal parameter is as a sized array

Way 3:

```
void myFunction(int param[]) {
    ..
    ..
    ..
}
```

Here formal parameters are unsized array

Ways of passing a 2d array into functions:

One of the most important things for passing multidimensional arrays is, the first array dimension does not have to be specified. The second (and any subsequent) dimensions must be given. Means we can say that it’s not necessary to specify the row but the column specification must be given.

Way 1:

```
const int m=3;
const n=3;
void print(int arr[n][m])
{
```

```
. .
}
```

Here both dimensions are available globally (either as a macro or as a global constant).

Way 2:

```
const int n=3;
void print(int arr[][n])
{
..
}
```

Or

```
#define n 3
void  print(int arr[][n])
{
..
}
```

Here only the second dimension is available globally (either as a macro or as a global constant).

These are the ways of passing a fixed size array. To pass a variable sized array you can use pointers.

We will try to submit these 5 questions on HackerEarth platform so that you become comfortable using these platforms.

Problem1: HackerEarth link

Problem 2: HackerEarth link

Problem 3: HackerEarth link

Problem 4: HackerEarth link

Problem 5: Hacker earth link

Practice Problems:

Question 1

Micro purchased an array *A* having *N* integer values. After playing it for a while, he got bored of it and decided to update the value of its element. In one second he can increase the value of each array element by *1*. He wants each array element's value to become greater than or equal to *K*. Please help Micro to find out the minimum amount of time it will take for him to do so.

Input:

First line consists of a single integer, *T*, denoting the number of test cases.

First line of each test case consists of two space separated integers denoting *N* and *K*.

Second line of each test case consists of *N* space separated integers denoting the array *A*.

Output:

For each test case, print the minimum time in which all array elements will become greater than or equal to *K*. Print a new line after each test case.

Constraints:

$1 \leq T \leq 5$
$1 \leq N \leq 10^5$
$1 \leq A[i], K \leq 10^6$

SAMPLE INPUT

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

SAMPLE OUTPUT

```
3
```

0

Explanation

For first test case,

After *1* second, array will be {2 ,3 ,6 }

After *2* second, array will be {3 ,4 ,7 }

After *3* second, array will be {4 ,5 ,8 }

So it will take *3* second for all array elements to become greater than or equal to *4*.

Program Link :- HackerEarth

Solution : -

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
        int t;
        cin>>t;
        while(t--)
        {
int n,mini=INT_MAX,k;
cin>>n>>k;
int arr[100001];
for(int i=0;i<n;i++)
{
        cin>>arr[i];
        if(arr[i]<mini)mini=arr[i];
}
if(k-mini<0)cout<<0<<endl;
else
cout<<(k-mini)<<endl;
        }
}
```

Question 2

Students have become secret admirers of SEGP. They find the course exciting and the professors amusing. After a superb Mid Semester examination its now time for the results. The TAs have released the marks of students in the form of an array, where arr[i] represents the marks of the ith student.

Since you are a curious kid, you want to find all the marks that are not smaller than those on its right side in the array.

Input Format

The first line of input will contain a single integer n denoting the number of students.

The next line will contain n space separated integers representing the marks of students.

Output Format

Output all the integers separated in the array from left to right that are not smaller than those on its right side.

Constraints

1 <= n <= 1000000

0 <= arr[i] <= 10000

SAMPLE INPUT

```
6
16 17 4 3 5 2
```

SAMPLE OUTPUT

```
17 5 2
```

Program Link :- HackerEarth

Explanation: Traverse all the elements of the array and find whether it is greater than all the elements on the right side or not. If it is then print the element else do not. In the above test case there is no element greater than 17 on it's right side, hence 17 is included in the answer.

Solution : -

```
#include<iostream>
using namespace std;
int main()
{
        int n,i,j,flag=1;
        cin>>n;
        int arr[n];
        for(i=0;i<n;i++)
        {
        cin>>arr[i];
        }
        for(i=0;i<n-1;i++)
        {
                for(j=i+1;j<n;j++)
                {
                        if(arr[i]<arr[j])
                        {
                                flag=0;
                                break;
                        }
                }
                if(flag==1)
                cout<<arr[i]<<" ";
                flag=1;
        }
        cout<<arr[n-1]<<endl;
}
```