*void* printarray(*int* *arr*[],*int* *size*)

{

    for (*int* i=0;i<size;i++)

    {

        cout<<arr[i]<<" ";

     }

     cout<<"printingDone"<<endl;

}

*int* linearSearch(*int* *arr*[],*int* *size*, *int* *element*)

{

    for(*int* i=0;i<size;i++)

    {

        if (arr[i]==element)

        return i;

    }

    return -1 ;

}

*int* linearSearchRecursive(*int* *arr*[],*int* *size*,*int* *element*)

{

     if (size==0)

     return -1;

     else if (arr[size-1]==element)

     return size-1;

     else

     return linearSearchRecursive(arr,size-1,element);

}

*int* binarySearch(*int* *arr*[],*int* *size*, *int* *element*)

{

*int* start=0 , end = size-1 ,mid ;

    while(start<=end)

    {

         mid = (start+end)/2;

         if (arr[mid]==element)

         return mid;

         else if (arr[mid]>element)

         end = mid -1;

         else

         start=mid+1;

    }

    return -1;

}

*int* recursiveBinarySearch(*int* *arr*[], *int* *start* , *int* *end* , *int* *element*)

{

    if (start<=end)

    {

*int* mid = (start+end)/2 ;

        if (arr[mid]==element)

        {

            return mid;

        }

        else if (arr[mid]>element)

        return recursiveBinarySearch(arr,start,mid-1,element);

        else

        return recursiveBinarySearch(arr,mid+1,end,element);

    }

    return -1;

}

*void* leftRotate(*int* *arr*[],*int* *size*,*int* *k*)

{

    while(k--)

    {

*int* temp = arr[0];

        for (*int* i=0;i<=size-2;i++)

        {

            arr[i]=arr[i+1];

        }

        arr[size-1]=temp;

    }

}

*void* rightRotate(*int* *arr*[],*int* *size*, *int* *k*)

{

  while(k--)

  {

*int* temp=arr[size-1];

    for (*int* i =size-2 ;i>=0;i--)

    {

        arr[i+1]=arr[i];

    }

    arr[0]=temp;

  }

}

*int* firstoccurence(*int* *arr*[],*int* *size*, *int* *element*)

{

*int* s=0,e=size-1,mid;

    while(s<e)

    {

        mid = (s+e)/2;

        if (arr[mid]==element && arr[mid]>arr[mid-1])

        {

            return mid ;

        }

        else if(arr[mid]<element){

              s=mid+1;

        }

         else

        {

            e=mid-1;

        }

    }

}

*int* lastoccurence(*int* *arr*[],*int* *size*,  *int* *element*)

{

*int* s=0,e=size-1,mid;

    while(s<e)

    {

        mid = (s+e)/2;

        if (arr[mid]==element && arr[mid]<arr[mid+1])

        {

            return mid ;

        }

        else if(arr[mid]>element){

              e=mid-1;

        }

         else

        {

             s=mid+1;

        }

    }

}

*int* main(){

*int* n ;

    cin>>n;

*int* arr[n];

    for (*int* i = 0; i< n;i++)

    {

        cin>>arr[i];

    }

*int* element ;

    cin>>element;

*int* first = firstoccurence(arr,n,element);

*int* last = lastoccurence(arr,n,element);

*int* count  = (last-first+1);

    cout<<first<<endl;

    cout<<last<<endl;

    cout<<count<<endl;

}

*int* firstUniqiueOccurenceInString(string *str*)

{

  // return first unique occurence wala character in the string

*int* fa[26]={0};

*int* size=str.size();

for (*int* i = 0 ; i<size;i++)

  {

    if (str[i]==' ')

    {

      continue;

    }

    fa[str[i]-'a']++;

  }

//  for(int i = 0 ;i<26;i++)

//  {

//     cout<<fa[i]<<" ";

//  }

    for (*int* i = 0 ;i<size;i++)

    {

       if (fa[str[i]-'a']==1)

       {

        return i; // unique character frequency index

       }

    }

    return -1;

}

*int* main(){

    string str;

    cin>>str;

    if (firstoccurence(str)==-1){

   cout<<"no unique character"<<endl;

   return 0;

    }

  cout<<"unique character is "<<str[firstoccurence(str)]<<endl;

 }

*int* maxElement(*int* *arr*[],*int* *size*)

{

*int* max=arr[0];

    for (*int* i =0;i<size;i++)

    {

        if (arr[i]>max){

        max=arr[i];

        }

    }

    return max;

}

*int* minElement(*int* *arr*[],*int* *size*)

{

*int* min=arr[0];

    for (*int* i =0;i<size;i++)

    {

        if (arr[i]<min){

        min=arr[i];

        }

    }

    return min;

}

*void* swapAlternate(*int* *arr*[], *int* *size*) {

    for(*int* i = 0; i<size; i+=2 ) {

        if(i+1 < size) {

            swap(arr[i], arr[i+1]);

        }

    }

}

*void* sort012(*int* *arr*[],*int* *size*)

{

*int* count0=0 , count1=0 , count2=0;

    for(*int* i =  0 ; i < size ; i++)

    {

        if(arr[i]==0)

        {

            count0++;

        }

        if(arr[i]==1)

        {

            count1++;

        }

        if(arr[i]==2)

        {

            count2++;

        }

    }

*int* k = 0 ;

    while(count0--){

        arr[k]=0;

        k++;

    }

    while(count1--){

        arr[k]=1;

        k++;

    }

    while(count2--){

        arr[k]=2;

        k++;

    }

}

**Find the smallest missing number**

Given a **sorted**array of n distinct integers where each integer is in the range from 0 to m-1 and m > n. Find the smallest number that is missing from the array.

**Examples:**

**Input:** {0, 1, 2, 6, 9}, n = 5, m = 10

**Output:** 3

**Input:** {4, 5, 10, 11}, n = 4, m = 12

**Output:** 0

**Input:** {0, 1, 2, 3}, n = 4, m = 5

**Output:** 4

**Input:** {0, 1, 2, 3, 4, 5, 6, 7, 10}, n = 9, m = 11

**Output:** 8

#include <iostream>

using *namespace* std;

// Given a sorted array of n distinct integers where each integer is in the range from 0 to m-1 and m > n. Find the smallest number that is missing from the array.

*int* findNumber(*int* *arr*[], *int* *size*)

{

    // find max

*int* max=arr[0];

    for (*int*  i = 0 ; i<size;i++)

    {

        if (arr[i]>max)

        max=arr[i];

    }

*int* fa[size+1]={0};

    for (*int* i  = 0 ; i<size;i++)

    {

        fa[arr[i]]++;

    }

    for (*int* i=0;i<=max+1;i++)

    {

        if (fa[i]<1)

        return i ;

    }

    }

*int* main(){

*int* size;

    cin>>size ;

*int* arr[size];

    for (*int* i = 0 ; i< size ;i++)

    cin>>arr[i];

    cout<<findNumber(arr,size)<<endl;

}