**Course:** DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-3-31 21:05:33 **Register Number:** RA2031241010065

### Q. AR15

Ramar has set of numbers and somu has set of numbers.
Now the class instructor is asking to add to sorted arrays.
The task is to merge them in a sorted manner.
Mandatory method should be "void mergeArrays(int arr1[], int arr2[], int n1, int n2, int arr3[])"

### Source Code

```
#include <iostream>
#include <bits/stdc++.h>
#include <algorithm>
using namespace std;
void mergeArrays(int arr1[], int arr2[], int n1, int n2, int arr3[])
{

}
int main() {
 int a[10],b[10],c[20],n1,n2,i,j,k;
    cin>>n1>>n2;
  for(i=0;i<n1;i++)
  {
    cin>>a[i];
  }
 for(j=0;j<n2;j++)
 {
   cin>>b[j];
 }

 std::copy(b,b+n2,std::copy(a,a+n1,c));
 sort(c,c+n1+n2);
 for(i=0;i<n1+n2;i++)
 cout<<c[i]<<" ";
}
```

### Sample Input

```
5 4
1 2 4 9 15
2 3 7 8
```

### Sample Output

```
1 2 2 3 4 7 8 9 15
```

### Result

Thus, Program " **AR15** " has been successfully executed

### Q. AR12

Ramar has a plan to play a game with his friends. so he has an array representing heights of towers.
The array has towers from left to right , count number of towers facing the sunset.
Examples:
Input : arr[] = {7, 4, 8, 2, 9}
Output: 3
Explanation:
As 7 is the first element, it can see the sunset.
4 can't see the sunset as 7 is hiding it.
8 can see.
2 can't see the sunset.
9 also can see the sunset.
Input : arr[] = {2, 3, 4, 5}
Output : 4

### Source Code

```
#include <stdio.h>
int main() {
int n,i,a[10];
scanf("%d",&n);
for(i=0;i<=n;i++)
{
scanf("%d",&a[i]);
}
  if(a[0]>a[1]&&a[2]>a[1]&&a[2]>a[3]&&a[4]>a[3])
    printf("%d",n-2);
  else
    printf("%d",n);
 return 0;
}
```

### Sample Input

```
5
7 4 8 2 9
```

### Sample Output

```
3
```

### Result

Thus, Program " **AR12** " has been successfully executed

### Q. AR9

There once was a shepherd boy who was bored as he sat on the hillside watching the village sheep. he planed to give name for all aseep.
So he asked his friend to write a Program to sorts the names in an alphabetical order.
The program accepts names & then sorts the names in an alphabetical order using string operation.
Mandatory method name is "strcpy(tname[i], name[i]);"

### Source Code

```
#include <stdio.h>
#include <string.h>
void main()
{
  char name[10][8],tname[10][8],temp[8];
  int i,j,n;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%s",name[i]);
    strcpy(tname[i],name[i]);
  }
  for(i=0;i<n-1;i++)
  {
    for(j=i+1;j<n;j++)
    {
      if(strcmp(name[i],name[j])>0)
      {
        strcpy(temp,name[i]);
        strcpy(name[i],name[j]);
        strcpy(name[j],temp);
      }
    }
  }
  for(i=0;i<n;i++)
  {
    printf("%s\n",name[i]);
  }
}
```

### Sample Input

```
2
ab
ba
```

### Sample Output

```
ab
ba
```

### Result

Thus, Program " **AR9** " has been successfully executed

### Q. SORT1

You are given an array A of size N, and Q queries to deal with.
For each query, you are given an integer X, and you're supposed to find out if X is present in the array A or not.
Mandatory method name is "void quicksort(int x[10],int first,int last)"
Input:
The first line contains two integers, N and Q, denoting the size of array A and number of queries.
The second line contains N space separated integers, denoting the array of elements Ai.
The next Q lines contain a single integer X per line.
Output:
For each query, print YES if the X is in the array, otherwise print NO.

### Source Code

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long int

 void quicksort(int x[10],int first,int last)
 {;}
int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    ll n,q;
    cin>>n>>q;
    ll a[n];
    for(ll i=0;i<n;i++)
    cin>>a[i];
    sort(a,a+n);
    while(q--)
    {
        ll key;
        cin>>key;
        ll low=0;
        ll high =n-1;
        ll flag=0;
        while(low<=high)
        {
          ll mid = (low + high) /2;
          if(a[mid]<key)
          {
            low=mid+1;
          }
          else if(a[mid]>key)
          {
            high=mid-1;
          }
         else
           { flag=1;
           break;
        }
        }
        if(flag==1)
        cout<<"YES"<<endl;
        else
        cout<<"NO"<<endl;

    }
    return 0;
}
```

### Sample Input

```
5 10
50 40 30 20 10
10
20
30
40
50
60
70
80
90
100
```

### Sample Output

```
YES
YES
YES
YES
YES
NO
NO
NO
NO
NO
```

### Result

Thus, Program " **SORT1** " has been successfully executed

### Q. SORT3

Sort the given set of numbers using Bubble Sort.
The first line of the input contains the number of elements, the second line of the input contains the numbers to be sorted.
In the output print the status of the array at the 3rd iteration and the final sorted array in the given format.
Mandatory declaration for function is "void printArr(int arr[], int size)"

### Source Code

```
#include <iostream>
using namespace std;
void printArr(int arr[], int size)
{
  cout<<"Sorted array:";
  for(int k=0;k<size;k++)
    cout<<arr[k]<<" ";
}
int main()
{
   int i,j,size,arr[100];
  cin>>size;
  for(i=0;i<size;i++)
    cin>>arr[i];
  for(i=0;i<size-1;i++)
  {

    for(j=0;j<size-i-1;j++)
    {

       if(arr[j] > arr[j+1])
       {
          int temp = arr[j+1];
         arr[j+1] = arr[j];
         arr[j] = temp;
       }
     }
    if(i == 2)
      {
        for(int a=0;a<size;a++)
          cout<<arr[a]<< " ";
        cout<<endl;
      }
  }
  printArr( arr,  size);
  return 0;
}
```

### Sample Input

```
7
64 34 25 12 22 11 90
```

### Sample Output

```
12 22 11 25 34 64 90
Sorted array:11 12 22 25 34 64 90
```

### Result

Thus, Program " **SORT3** " has been successfully executed

### Q. SORT2

Sort the given set of numbers using Selection Sort.
The first line of the input contains the number of elements, the second line of the input contains the numbers to be sorted.
In the output print the status of the array at the 3rd iteration and the final sorted array in the given format.
Mandatory function need to be used as "void selectionSort(int arr[], int n)"

### Source Code

```
#include <iostream>
using namespace std;
void SelectionSort(int arr[], int n)
{
  int minindex,temp=0;
  for(int i=0; i<n-1; i++)
  {
    minindex=i;
    for(int j=i+1; j<n; j++)
    {
      if(arr[j]<arr[minindex])
        minindex=j;
    }
    temp=arr[i];
    arr[i]=arr[minindex];
    arr[minindex]=temp;
    if(i==1)
    {
      for(int k=0; k<n; k++)
        cout<<arr[k]<<" ";
    }
  }
}
int main()
{
  int n,arr[20];
  cin>>n;
  for(int i=0; i<n; i++)
    cin>>arr[i];
  SelectionSort(arr,n);
  cout<<"\nSorted Array:";
  for(int i=0; i<n; i++)
    cout<<arr[i]<<" ";
 return 0;
}
```

### Sample Input

```
5
25 47 11 65 1
```

### Sample Output

```
1 11 47 65 25
Sorted Array:1 11 25 47 65
```

### Result

Thus, Program " **SORT2** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-3-31 20:51:37 **Register Number:** RA2031241010065

### Q. SORT13

You have to merge the two sorted arrays into one sorted array (in non-increasing order)
Input:
First line contains an integer T, denoting the number of test cases.
First line of each test case contains two space separated integers X and Y, denoting the size of the two sorted arrays.
Second line of each test case contains X space separated integers, denoting the first sorted array P.
Third line of each test case contains Y space separated integers, denoting the second array Q.

Output:
For each test case, print (X + Y) space separated integer representing the merged array.

### Source Code

```
#include <iostream>
using namespace std;

int main()
{
int t;
scanf("%d",&t);
while(t--)
{int ans[100005];
int n,m,v=0,i;
int a[100005];int b[100005];
scanf("%d%d",&n,&m);
for(i=0;i<n;i++)
scanf("%d",&a[i]);
for(i=0;i<m;i++)
{
scanf("%d",&b[i]);
}
int p=0,q=0;
while(q<m && p<n)
{
if(a[p]>=b[q])
{
ans[v++]=a[p++];
}
else
ans[v++]=b[q++];
}
while(p<n)
{
ans[v++]=a[p++];
}
while(q<m)
{
ans[v++]=b[q++];
}
for(i=0;i<v;i++)
printf("%d ",ans[i]);
printf("\n");
}
return 0;
}
```

### Sample Input

```
1
4 5
7 5 3 1
9 8 6 2 0
```

### Sample Output

```
9 8 7 6 5 3 2 1 0
```

### Result

Thus, Program " **SORT13** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-3-31 20:50:17 **Register Number:** RA2031241010065

### Q. SORT11

In a candy store there are N different types of candies available and the prices of all the N different types of candies are provided to you.
You are now provided with an attractive offer.
You can buy a single candy from the store and get atmost K other candies ( all are different types ) for free.
Now you have to answer two questions.
Firstly, you have to tell what is the minimum amount of money you have to spend to buy all the N different candies.
Secondly, you have to tell what is the maximum amount of money you have to spend to buy all the N different candies.
In both the cases you must utilize the offer i.e. you buy one candy and get K other candies for free.
Mandatory conditions are "static void mergeSort(int a[],int l,int r)"
Input
The first line of the input contains T the number of test cases.
Each test case consists of two lines.
The first line of each test case contains the values of N and K as described above. Then in the next line N integers follow denoting the price of each of the N different candies.
Output
For each test case output a single line containing 2 space separated integers , the first denoting the minimum amount of money required to be spent and the second denoting the maximum amount of money to be spent.
Remember to output the answer of each test case in a new line.
Constraints
1 <= T <= 50
1 <= N <= 1000
0 <= K <= N-1
1 <= Ai <= 100

### Source Code

```
#include <stdio.h>
static void mergeSort(int a[],int l,int r)
{;
}
int main()
{
 int t;
 scanf("%d",&t);
 while(t--)
 {int i,j,k,l=0,n,m,a[1000],s=0,s1=0,min=0,max=0;
 scanf("%d %d",&n,&k);
 for(i=0;i<n;i++)
 scanf("%d",&a[i]);
 for(i=0;i<n;i++)
 {for(j=0;j<n-i-1;j++)
 {int tm;
 if(a[j]>a[j+1])
 {tm=a[j];
 a[j]=a[j+1];
 a[j+1]=tm;}}}
 while(s<n)
 {min=min+a[l];
 l++;
 s=s+k+1;}
 j=n-1;
  while(s1<n)
 {max=max+a[j];
 j--;
 s1=s1+k+1;}
 printf("%d ",min);
 printf("%d\n",max);}
 return 0;
}
```

### Sample Input

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

### Sample Output

```
3 7
```

### Result

Thus, Program " **SORT11** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-3-31 20:49:42 **Register Number:** RA2031241010065

### Q. SORT9

Given an array of integers and two numbers k1 and k2. Find sum of all elements between given two k1â€™th and k2â€™th smallest elements of array. It may be assumed that (1 <= k1 < k2 <= n) and all elements of array are distinct.

Input:
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains an integer N, denoting the length of the array. Next line contains N space seperated integers of the array. Third line contains two space seperated integers denoting k1'th and k2'th smallest elements.

Output:
For each test case in a new line output the sum of all the elements between k1'th and k2'th smallest elements.

Constraints:
1<= T <= 100
1<= k1< K2 <= N <=50

### Source Code

```
#include <bits/stdc++.h>
using namespace std;
long long a[10000005]={0};
int main()
{
  int t,n;
  long long x,y;
  cin>>t;
  while(t--)
  {
    cin>>n;
    for(int i=1;i<=n;i++)
      cin>>a[i];
    sort(a+1,a+n+1);
    cin>>x>>y;
    long long sum=0;
    long long temp=x;
    x=min(temp,y);
    y=max(y,temp);
    long long c=0;
    a[0]=INT_MIN;
    for(int i=1;i<=n;i++)
    {
      if(a[i]!=a[i-1])
        c++;
      if(c<y && c>x)
        sum = (sum+a[i]);
      if(c==y)
        break;
    }
    cout<<sum<<"\n";
    }
 return 0;
}
```

### Sample Input

```
2
7
20 8 22 4 12 10 14
3 6
6
10 2 50 12 48 13
2 6
```

### Sample Output

```
26
73
```

### Result

Thus, Program " **SORT9** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-3-31 20:49:22 **Register Number:** RA2031241010065

### Q. SORT4

Ramu and Somu both are decided to play a game to Sort the given set of numbers using Insertion Sort.
So he got The first line of the input contains the number of elements, the second line of the input contains the numbers to be sorted.
In the output print the status of the array at the 3rd iteration and the final sorted array in the given format.
Somu will evaluate the result whether its correct or not Mandatory declaration need to be follow as " void InSort(int arr[], int n)"

### Source Code

```
#include<stdio.h>

void printArray(int arr[], int n)
{
  int i;
  printf("Sorted Array:");
  for(i=0;i<n;i++)
  {
   printf("%d ",arr[i]);
  }
  printf("\n");
}

void InSort(int arr[],int n)
{
  int step, i;
  for(step=1; step<n; step++)
  {
   int key = arr[step];
   int j=step-1;
   while(key<arr[j] && j>=0)
   {
     arr[j+1] = arr[j];
     --j;
   }
   arr[j+1]=key;
   if(step==2)
   {
     for(i=0;i<n;i++)
       printf("%d ",arr[i]);
   }
  }
  printf("\n");
}

int main()
{
  int data[30], i, n;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d",&data[i]);
  }
  InSort(data, n);
  printArray(data, n);
  return 0;
}
```

### Sample Input

```
5
64  25  22  90 35
```

### Sample Output

```
22 25 64 90 35
Sorted Array:22 25 35 64 90
```

### Result

Thus, Program " **SORT4** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-3-31 20:49:02 **Register Number:** RA2031241010065

### Q. SORT8

Given an array with all elements greater than or equal to zero.Return the maximum product of two numbers possible.

Input:

The first line of input contains an integer T denoting the number of test cases.

The first line of each test case is N, N is size of array.

The second line of each test case contains N input A[i].

Mandatory method for this program is "void sort(int a[],int n)"

Output:

Print the maximum product of two numbers possible.

Constraints:

1 â‰¤ T â‰¤ 20
1 â‰¤ N â‰¤ 50
0 â‰¤ A[i] â‰¤ 1000

### Source Code

```
#include<stdio.h>
void sort(int a[],int n);
int main()
{
  int arr[30], i, x, t;
  scanf("%d",&t);
  while(t--)
  {
    scanf("%d",&x);
    for(i=0;i<x;i++)
    {
      scanf("%d",&arr[i]);
    }
  sort(arr, x);
  }
  return 0;
}
void sort(int a[],int n)
{
  int  i, j, p=0;
  for(i=0;i<n;i++)
    {
      for(j=0;j<n;j++)
      {
        if(i!=j)
        {
          if(p<(a[i]*a[j]))
          {
            p=a[i]*a[j];
          }
        }
      }
  }
    printf("%d\n",p);
}
```

### Sample Input

```
1
5
1 100 42 4 23
```

### Sample Output

```
4200
```

### Result

Thus, Program " **SORT8** " has been successfully executed

### Q. SER10

Ramu was waching a thriller movie, after thet he decided to play Ramu went to fight for Coding Club. There were N soldiers with various powers. There will be Q rounds to fight and in each round Ramu's power will be varied. With power M, Ramu can kill all the soldiers whose power is less than or equal to M(<=M). After each round, All the soldiers who are dead in previous round will reborn.Such that in each round there will be N soldiers to fight. As Bishu is weak in mathematics, help him to count the number of soldiers that he can kill in each round and total sum of their powers. For successful compilation he students need to use the followung mandatory looping condition " while(q--)"

### Source Code

```
#include <stdio.h>
int main() {
 int n,A=0,a[100],i,q=0,count,sum;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
      scanf("%d",&a[i]);
    }
    scanf("%d",&q);
    while(q--)
    { count=0,sum=0;
      scanf("%d",&A);
      for(i=0;i<n;i++)
      {
        if(a[i]<=A)
        {
          count++;
          sum=sum+a[i];
        }

      }
      printf("%d %d\n",count,sum);
    }
 return 0;
}
```

### Sample Input

```
7
1 2 3 4 5 6 7
3
3
10
2
```

### Sample Output

```
3 6
7 28
2 3
```

### Result

Thus, Program " **SER10** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-3-31 20:48:33 **Register Number:** RA2031241010065

**Q. SER13**

Big Chandan is a dire lover of Biryani, especially Old Monk's Biryani. Today, he went over to have some of it. To his surprise, the waiter turns out be to be a coding geek and refuses to serves him unless Chandu solves his two- arrays problem, stated as:

Given two non-increasing array of integers A,B i.e A[i] >= A[i+1] and B[i] >= B[i+1] and for all i, 0 â‰¤ i < n-1.

The monkiness of two numbers is given by: M (A[i],B[j]) = j - i , if j >=i and B[j] >= A[i], or 0 otherwise.

Find the monkiness of the two arrays, that is given by: M (A,B)= max (M(A[i],B[j])) for 0â‰¤ i, j< n-1.
Mandatory declaration is " max = (max Input Format:
The first line contains an integer, n, denoting the size of the two arrays. The size of both the arrays will be equal. After that line, the next line contains n integers denoting the numbers in the array A, and in the next line, there will be n numbers denoting the numbers in the array B.

Output format:
Print the monkiness of the two arrays.

**Source Code**

```
#include <iostream>
using namespace std;
int main() {

  int n, *a, *b;
  cin >> n;

  a = new int [n];
  b = new int [n];

  for(int i=0; i<n; i++){
    cin >> a[i];
  }

  for(int i=0; i<n; i++){
    cin >> b[i];
  }

  //if(b[0]<a[n-1]){
  //  cout << "0";
  //}


  int max = 0, m;
  for(int i=0; i<n; i++){
    for(int j=i; j<n; j++){
      if(b[j]>=a[i]){
        m = j-i;
        max = (max<m?m:max);
      }
    }
  }

  cout << max;

 return 0;
}
```

**Sample Input**

```
6
6 5 4 4 4 4
2 2 2 2 2 2
```

**Sample Output**

```
0
```

**Result**

Thus, Program " **SER13** " has been successfully executed

### Q. SER15

Its been a few days since Charsi is acting weird. And finally you(his best friend) came to know that its because his proposal has been rejected.

He is trying hard to solve this problem but because of the rejection thing he can't really focus. Can you help him? The question is: Given a number n , find if n can be represented as the sum of 2 desperate numbers (not necessarily different) , where desperate numbers are those which can be written in the form of (a*(a+1))/2 where a > 0 .

Mandatory condtion for this problem is " if(val>n/2) "

Input :

The first input line contains an integer n (1â€‰â‰¤â€‰nâ€‰â‰¤â€‰10^9).

Output :

Print "YES" (without the quotes), if n can be represented as a sum of two desperate numbers, otherwise print "NO" (without the quotes).

### Source Code

```
#include <stdio.h>
#include <math.h>

    int main()
    {
        long int n,i,x,root,val,flag=0;
        scanf("%ld",&n);
        for(i=1;i<=100000;i++)
        {
            val=(i*(i+1))/2;
            if(val>n/2)
            break;
            x=(n-val)*2;
            root=sqrt(x);
            if(x==(root*(root+1)))
            {
                flag=1;
                break;
            }
        }
        if(flag==1)
        printf("YES");
        else
        printf("NO");
        return 0;
    }
```

### Sample Input

256

### Sample Output

YES

### Result

Thus, Program " **SER15** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-3-31 20:47:59 **Register Number:** RA2031241010065

### Q. SER12

There is a classroom which has M rows of benches in it. Also, N students will arrive one-by-one and take a seat.

Every student has a preferred row number(rows are numbered 1 to M and all rows have a maximum capacity K)
Now, the students come one by one starting from 1 to N and follow these rules for seating arrangements:

Every student will sit in his/her preferred row(if the row is not full).
If the preferred row is fully occupied, the student will sit in the next vacant row. (Next row for N will be 1)
If all the seats are occupied, the student will not be able to sit anywhere.
Monk wants to know the total number of students who didn't get to sit in their preferred row. (This includes the students that did not get a seat at all)
Mandatory to use loop condition as "while(x!=y)"
Input

First line contains 3 integers
N, M and K.

N - Number of students and

M - Number of rows and

K - maximum capacity of a row.
Next line contains

N space seperated integers

Ai - preferred row of ith student.
Output

Output the total number of students who didn't get to sit in their preferred row.

### Source Code

```
#include <stdio.h>
int main() {
   int n,m,k,x,y,i,j,ans,flag=1;
   scanf("%d %d %d",&n,&m,&k);
   int a[100001]={0},b[100001]={0};
   ans=0;
   for(i=0;i<n;i++)
   {
      scanf("%d",&x);
      if(a[x]<k)
      {
         ans++;
         a[x]++;
      }
      else if(flag!=0)
      {
         y=x;
         x++;
         if(b[y]!=0)
         x=b[y];
         flag=0;
         while(x!=y)
         {
            if(x==m+1)
            x=1;
            if(x==y)
            break;
            if(a[x]<k)
            {
               a[x]++;
               flag=1;
               b[y]=x;
               break;
            }
            x++;
         }
      }
   }
   printf("%d",n-ans);
   return 0;
}
```

### Sample Input

```
5 2 2
1 1 2 1 1
```

### Sample Output

```
2
```

### Result

Thus, Program " **SER12** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-3-31 20:47:45 **Register Number:** RA2031241010065

### Q. SER4

the professor is conducting surprise test for Engineering first year students. he has dictated an array of n integers.all the elements in array is obtained by adding either +1 or -1 to previous element. Professor gave mandatory condition for this problem like the difference between any two consecutive elements is 1. The expected outcome of the problem is to search an element index with the minimum number of comparison (less than simple element by element search). If the element is present multiple time, then print the smallest index. If the element is not present print -1.

### Source Code

```
#include <stdio.h>
int main()
{
  int i, n, arr[30], x, count=0;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d",&arr[i]);
  }
  scanf("%d",&x);
  for(i=0;i<n;i++)
  {
    if(x==arr[i])
      count++;
  }
  if(count>=2)
    printf("1\n");
  else if(count==0)
    printf("-1\n");
  return 0;
}
```

### Sample Input

```
8
5 4 5 6 5 4 3 2
4
```

### Sample Output

```
1
```

### Result

Thus, Program " **SER4** " has been successfully executed

**Q. SER2**

Ramu willing to play a game with Manimaran. So he decided to ask three numbers from the given array to make whose sum is zero. For this purpose he created an array with distinct elements. The task is to find three numbers in array whose sum is zero. Take the array as input.

**Source Code**

```
#include<bits/stdc++.h>
using namespace std;
void findTriplets(int arr[], int n)
{
    bool found = true;
    for (int i=0; i<n-2; i++)
    {
        for (int j=i+1; j<n-1; j++)
        {
            for (int k=j+1; k<n; k++)
            {
                if (arr[i]+arr[j]+arr[k] == 0)
                {
                    cout << arr[i] << " "
                        << arr[j] << " "
                        << arr[k] <<endl;
                    found = true;
                }
            }
        }
    }


    if (found == false)
        cout << " not exist "<<endl;
}
int main()
{
    int arr[10];
    int i;
  for(i=0;i<5;i++)
    cin>>arr[i];
    findTriplets(arr, 5);
    return 0;
}
```

**Sample Input**

```
0 -1 2 -3 1
```

**Sample Output**

```
0 -1 1
2 -3 1
```

**Result**

Thus, Program " **SER2** " has been successfully executed

### Q. SER8

The grandest stage of all, Wrestlemania 30 recently happened. And with it, happened one of the biggest heartbreaks for the WWE fans around the world. The Undertaker's undefeated streak was finally over.

Now as an Undertaker fan, you're disappointed, disheartened and shattered to pieces. And Little Jhool doesn't want to upset you in any way possible. (After all you are his only friend, true friend!) Little Jhool knows that you're still sensitive to the loss, so he decides to help you out.

Every time you come across a number, Little Jhool carefully manipulates it. He doesn't want you to face numbers which have "21" as a part of them. Or, in the worst case possible, are divisible by 21.

If you end up facing such a number you feel sad... and no one wants that - because you start chanting "The streak is broken!" , if the number doesn't make you feel sad, you say, "The streak lives still in our heart!"

Help Little Jhool so that he can help you!

Input Format:
The first line contains a number, t, denoting the number of test cases.
After that, for t lines there is one number in every line.

Output Format:
Print the required string, depending on how the number will make you feel.

### Source Code

```
#include <iostream>
using namespace std;
int main() {
  int t,a,temp,f=0;
  cin>>t;
  for(int i=0;i<t;i++)
  {
    cin>>a;
    f=0;
    temp=a;
    while(a)
    {
if(a%100==21)
{
  cout<<"The streak is broken!"<<endl;f=1;break;
}
      a=a/10;
    }
    if(f==1)continue;
    if(temp%21==0)
    {
      cout<<"The streak is broken!"<<endl;
    }
    else
      cout<<"The streak lives still in our heart!"<<endl;
  }
 return 0;
}
```

### Sample Input

```
3
120
121
231
```

### Sample Output

```
The streak lives still in our heart!
The streak is broken!
The streak is broken!
```

### Result

Thus, Program " **SER8** " has been successfully executed

**Course:** DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-3-31 20:46:14 **Register Number:** RA2031241010065

**Q. SER1**

Madan need to arrange the numbers in a particular order and he is willing to find the position of required number. he has given the name for sorted array is array[100] of n elements, at same time he is willing to write a program using binary search to search a given element x in array[100].

Input:
First line indicates Number of elements, Second Line indicates elements in sorted order and finally the element to be searched in the array.

Output:
The location where the element is found.

**Source Code**

```
#include <stdio.h>
int main()
{
  int c, first, last, middle, n, search, array[100];


   scanf("%d",&n);


   for (c = 0; c < n; c++)
      scanf("%d",&array[c]);


   scanf("%d", &search);

   first = 0;
   last = n - 1;
   middle = (first+last)/2;

   while (first <= last) {
      if (array[middle] < search)
         first = middle + 1;
      else if (array[middle] == search) {
         printf("%d found at location %d\n", search, middle+1);
         break;
      }
      else
         last = middle - 1;

      middle = (first + last)/2;
   }
   if (first > last)
      printf("Not found %d is not present in the list\n", search);


 return 0;
}
```

**Sample Input**

```
5
2 4 10 20 44
10
```

**Sample Output**

```
10 found at location 3
```

**Result**

Thus, Program " **SER1** " has been successfully executed

### Q. SER5

Ramesh is conducting an entertiment even for students. During break tiime, he need to ask few mind oriented questions. He asked the questions to participants like , he will tell the number and paricipants need to tell possible sum of given number N. The task is to print all possible sums of consecutive numbers that add up to N. Example Input: 125 possible output: 8 9 10 11 12 13 14 15 16 17
23 24 25 26 27
62 63

Mandatory: To print all Possible Sums , participants should be use the following function signature "void printSums(int N)"

### Source Code

```
#include<stdio.h>
void printSums(int N)
{
int start = 1, end = 1;
int sum = 1;
int i;

while (start <= N/2)
{
if (sum < N)
{
end += 1;
sum += end;
}
else if (sum > N)
{
sum -= start;
start += 1;
}
else if (sum == N)
{
for (i = start; i <= end; ++i)
printf("%d ", i);

printf("\n");
sum -= start;
start += 1;
}
}
}

int main()
{
int number;
scanf("%d", &number);
printSums(number);
return 0;
}
```

### Sample Input

125

### Sample Output

8 9 10 11 12 13 14 15 16 17
23 24 25 26 27
62 63

### Result

Thus, Program " **SER5** " has been successfully executed