Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-3-31 18:38:02 **Register Number:** RA2031241010094

Q. SER7

Kanna is extremely disappointed to find out that no one in his school knows his first name. Even his classmates call him only by his last name. Frustrated, he decides to make his fellow college students know his first name by forcing them to solve this question.
Kanna has given a long string as input in each testcase, containing any ASCII character. Your task is to find out the number of times SUVO and SUVOJIT appears in it.
Input Format
The first line contains the number of testcases, T. Next, T lines follow each containing a long string S.
Output Format
For each long string S, display the no. of times SUVO and SUVOJIT appears in it.

Source Code

```
#include <stdio.h>
#include<string.h>
int main()
{
  int i,j,n,l,p,c1,c2;
  char s[200];
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%s",s);
    c1=0;
    c2=0;
    l=strlen(s)-4;
    for(j=0;j<=l;j++)
    {
      if(s[j]=='S'&&s[j+1]=='U'&&s[j+2]=='V'&&s[j+3]=='O')
      {
        c1=c1+1;
      }
    }
    p=strlen(s)-7;
    for(j=0;j<=p;j++)
    {
      if(s[j]=='S'&&s[j+1]=='U'&&s[j+2]=='V'&&s[j+3]=='O'&&s[j+4]=='J'&&s[j+5]=='I'&&s[j+6]=='T')
      {
        c2=c2+1;
      }
    }
  printf("SUVO = %d, SUVOJIT = %d\n",c1-c2,c2);
  }
  return 0;
}
```

Sample Input

```
5
SUVOJITSUVOSUVOJITUCSUVOJITSUVOVXSUVOJITSUVOGMSUVODMMVDSUVOJIT
AXRSUVOJITHSUVOJITHSUVOJJSUVOJITSUVOJIT
SUVOJITPRQSUVOJIT
SUVOJITTXGSUVOUNSUVOJIT
SUVOJITSUVOSUVOJITXGSUVOSUVOQSUVOJITKDSALASUOUSUVOJITGJEM
```

Sample Output

```
SUVO = 4, SUVOJIT = 5
SUVO = 1, SUVOJIT = 4
SUVO = 0, SUVOJIT = 2
SUVO = 1, SUVOJIT = 2
SUVO = 3, SUVOJIT = 4
```

Result

Thus, Program " **SER7** " has been successfully executed

Q. SER14

You are given n triangles.
You are required to find how many triangles are unique out of given triangles. For each triangle you are given three integers a,b,c , the sides of a triangle.
A triangle is said to be unique if there is no other triangle with same set of sides.
Note : It is always possible to form triangle with given sides.
INPUT:
First line contains n, the number of triangles. Each of next n lines contain three integers a,b,c (sides of a triangle).
Output:
print single integer, the number of unique triangles.

Source Code

```
#include <stdio.h>
void swap(long long int *a,long long int *b)
{
  long long int t;
  t=*a;
  *a=*b;
  *b=t;
}
void qsort(long long int items[][4],long long int left,long long int right)
{
  long long int i,j,k;
  long long int x,y;
  i=left;
  j=right;
  x=items[(left+right)/2][3];
  do
  {
    while((items[i][3]<x)&&(i<right))
      i++;
    while((x<items[j][3])&&(j>left))
      j--;
    if(i<=j)
    {
      y=items[i][0];items[i][0]=items[j][0];items[j][0]=y;
      y=items[i][1];items[i][1]=items[j][1];items[j][1]=y;
      y=items[i][2];items[i][2]=items[j][2];items[j][2]=y;
      y=items[i][3];items[i][3]=items[j][3];items[j][3]=y;
      i++;
      j--;
    }
  }
  while(i<=j);
    if(left<j)
      qsort(items,left,j);
  if(i<right)
    qsort(items,i,right);
}
int main()
{
  long long int i,j,k,n,z,g,count=0;
  long long int p,q,r;
  long long int arr[100000][4]={0};
  scanf("%lld",&n);
  for(i=0;i<n;i++)
  {
    scanf("%lld %lld %lld",&arr[i][0],&arr[i][1],&arr[i][2]);
    arr[i][3]=arr[i][0]+arr[i][1]+arr[i][2];
    if(arr[i][0]>arr[i][1])
      swap(&arr[i][0],&arr[i][1]);
    if(arr[i][1]>arr[i][2])
      swap(&arr[i][1],&arr[i][2]);
    if(arr[i][0]>arr[i][1])
      swap(&arr[i][0],&arr[i][1]);
  }
  qsort(arr,0,n-1);
  i=0;
  while(1)
  {
    p=arr[i][0];
    q=arr[i][1];
    r=arr[i][2];
    k=arr[i][3];
    j=i;
    z=0;
    if(p==0)
    {
      ++i;
      if(i==n)
        break;
      else
        continue;
    }
    while(arr[++j][3]==k&&i<n)
    {
      if(arr[j][0]!=0&&arr[j][0]==p&&arr[j][1]==q&&arr[j][2]==r)
      {
        arr[j][0]=0;
        z++;
      }
    }
    if(z!=i-1-j&&z>=1)
    {
      i=j+1;
      continue;
    }
    else if(z==0)
    {
      count++;
      if(i!=j+1)
      {
        i=j+1;
        continue;
      }
    }
    if(i==n)
      break;
  }
  printf("%lld",count);
 return 0;
}
```

Sample Input

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

Sample Output

```
1
```

Result

Thus, Program " **SER14** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-4-1 09:50:19 **Register Number:** RA2031241010094

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,f,l,m,n,se,a[100];
  scanf("%d",&n);
  for(c=0;c<n;c++)
    scanf("%d",&a[c]);
  scanf("%d",&se);
  f=0;
  l=n-1;
  m=(f+l)/2;
  while(f<=l)
  {
    if(a[m]<se)
      f=m+1;
    else if(a[m]==se)
    {
      printf("%d found at location %d\n",se,m+1);
      break;
    }
    else
      l=m-1;
    m=(f+l)/2;
  }
  if(f>l)
 printf("Not found %d is not present in the list\n",se);
 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

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-4-1 09:50:52 **Register Number:** RA2031241010094

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:** 2021-4-1 09:51:41 **Register Number:** RA2031241010094

Q. SER6

Kapildev marketting a mobile phone like, if anyone answered this question, the mobile phone will be given for 50% flat offer. The task is to be find three closest elements from given three sorted arrays. So get three input aray from the user and these arrays should be in sorted manner. Take the three sorted arrays and their sizes as input. Final answer should be the closest element from three arrays. Method name has to be used like void findClosest()

Source Code

```
#include<iostream>
using namespace std;
void findclosest(int a[], int b[], int c[], int p, int q, int r)
{
  int diff= 1000;
  int resi=0,resj=0,resk=0;
  int i=0,j=0,k=0;
  while(i<p&&j<q&&k<r)
  {
    int minimum=min(a[i], min(b[j], c[k]));
    int maximum=max(a[i], max(b[j], c[k]));
    if(maximum-minimum < diff)
    {
      resi=i;
      resj=j;
      resk=k;
      diff=maximum-minimum;
    }
    if(diff==0)
      break;
    if(a[i]==minimum)
      i++;
    else if(b[j]==minimum)
      j++;
    else
      k++;
  }
  cout<<a[resi]<<" "<<b[resj]<<" "<<c[resk];
}
int main()
{
  int a[10],b[10],c[10];
  int p,q,r,i;
  scanf("%d",&p);
  for(i=0;i<p;i++)
  {
    scanf("%d",&a[i]);
  }
  scanf("%d",&q);
  for(i=0;i<q;i++)
  {
    scanf("%d",&b[i]);
  }
  scanf("%d",&r);
  for(i=0;i<r;i++)
  {
    scanf("%d",&c[i]);
  }
  findclosest(a, b, c, p, q, r);
 return 0;
}
```

Sample Input

```
3
1 4 10
3
2 15 20
2
10 12
```

Sample Output

```
10 15 10
```

Result

Thus, Program " **SER6** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-4-1 09:52:21 **Register Number:** RA2031241010094

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,a[30],x,count=0;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d",&a[i]);
  }
  scanf("%d",&x);
  for(i=0;i<n;i++)
  {
    if(x==a[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

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-4-1 09:52:51 **Register Number:** RA2031241010094

Q. SER5

Kanna is extremely disappointed to find out that no one in his school knows his first name. Even his classmates call him only by his last name. Frustrated, he decides to make his fellow college students know his first name by forcing them to solve this question like to find out the third largest number in the given array. For that he decided to mandatory conditios to declare the function name void thirdLargest(int arr[],int arr_size)

Source Code

```
#include <stdio.h>
void third(int a[],int n)
{
  int i,j,t,b;
  for(i=n-1;i>=0;i--)
  {
    for(j=0;j<i;j++)
    {
      if(a[j]>a[j+1])
      {
        t=a[j];
        a[j]=a[j+1];
        a[j+1]=t;
      }
    }
  }
  for(i=0;i<n;i++)
    b=a[n-3];
  printf("The third Largest element is %d",b);
}
int main()
{
  int a[20],ne,i;
  scanf("%d\n",&ne);
  for(i=0;i<ne;i++)
  {
    scanf("%d",&a[i]);
  }
  third(a,ne);
 return 0;
}
```

Sample Input

```
6
1 14 2 16 10 20
```

Sample Output

```
The third Largest element is 14
```

Result

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

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-4-1 09:54:19 **Register Number:** RA2031241010094

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 <stdio.h>
int main()
{
  int a,i,t,temp,f=0;
  scanf("%d",&t);
  for(i=0;i<t;i++)
  {
    scanf("%d",&a);
    f=0;
    temp=a;
    while(a)
    {
      if(a%100==21)
      {
        printf("The streak is broken!\n");
        f=1;
        break;
      }
      a=a/10;
    }
      if(f==1)
        continue;
      if(temp%21==0)
      {
        printf("The streak is broken!\n");
      }
      else
 printf("The streak lives still in our heart!\n");
    }
 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:** 2021-4-1 09:54:52 **Register Number:** RA2031241010094

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 <stdio.h>
void findtriplets(int a[],int n)
{
  int found=1;
  int i,j,k;
  for(i=0;i<n-2;i++)
  {
    for(j=i+1;j<n-1;j++)
    {
      for(k=j+1;k<n;k++)
      {
        if(a[i]+a[j]+a[k]==0)
        {
          printf("%d %d %d\n",a[i],a[j],a[k]);
          found=1;
        }
      }
    }
  }
  if(found==0)
    printf("not exist\n");
}
int main()
{
  int i,a[10];
  for(i=0;i<5;i++)
    scanf("%d",&a[i]);
  findtriplets(a,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

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2021-4-1 09:55:21 **Register Number:** RA2031241010094

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:** Sorting **Timestamp:** 2021-4-1 09:55:57 **Register Number:** RA2031241010094

Q. SORT6

MS .Dhoni want to play a game when the players are free in the dressing room.
Dhoni have given an array of n distinct elements, the task is to find all elements in array which have at-least two greater elements than themselves.
Dhoni declared a mandatory conditions like "void sort(int a[],int n)"
Examples:
Input : A[] = {2, 8, 7, 1, 5};
Output : 1 2 5
The output three elements have two or more greater elements
Input : A[] = {7, -2, 3, 4, 9, -1};
Output : -2 -1 3 4
Input:
The first line of input contains an integer T denoting the no of test cases.
Each test case contains two lines .
The first line of input contains an integer n denoting the size of the array.
Then in the next are n space separated values of the array.
Output:
For each test case in a new line print the space separated sorted values denoting the elements in array which have at-least two greater elements than themselves.

Source Code

```
#include<iostream>
using namespace std;
void sort(int a[],int n)
{
  int i,curr,j;
  for(i=0;i<n;i++)
  {
    curr=a[i];
  j=i-1;
  while(curr<a[j]&&j>=0)
  {
    a[j+1]=a[j];
    j--;
  }
  a[j+1]=curr;
  }
}
int main()
{
  int t,n,*arr;
  cin>>t;
  for(int p=0;p<t;p++)
  {
    cin>>n;
  arr=new int[n];
  for(int i=0;i<n;i++)
  {
    cin>>arr[i];
  }
  sort(arr,n);
  for(int i=0;i<n-2;i++)
  {
    cout<<arr[i]<<" ";
  }
  cout<<endl;
  }
  return 0;
}
```

Sample Input

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

Sample Output

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

Result

Thus, Program " **SORT6** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-4-1 09:56:30 **Register Number:** RA2031241010094

Q. SORT12

Given an array of integers, sort the array according to frequency of elements.
For example, if the input array is {2, 3, 2, 4, 5, 12, 2, 3, 3, 3, 12}, then modify the array to {3, 3, 3, 3, 2, 2, 2, 12, 12, 4, 5}.
If frequencies of two elements are same, print them in increasing order.
Input:
The first line of input contains an integer T denoting the number of test cases.
The description of T test cases follows.
The first line of each test case contains a single integer N denoting the size of array.
The second line contains N space-separated integers A1, A2, ..., AN denoting the elements of the array.
Output:
Print each sorted array in a seperate line.
For each array its numbers should be seperated by space.

Constraints:
1 â‰¤ T â‰¤ 70
30 â‰¤ N â‰¤ 130
1 â‰¤ A [i] â‰¤ 60

Source Code

```
#include <iostream>
#include<algorithm>
using namespace std;
class mymap
{
  public:
  int f,s;
  mymap(int f1,int s1)
  {
    f=f1;
    s=s1;
  }
  void print()
  {
    for(int i=0;i<s;i++)
      cout<<f<<' ';
  }
  void printnospace()
  {
    for(int i=0;i<s;i++)
      cout<<f;
  }
};
bool cmp(mymap m1,mymap m2)
{
  if(m1.s==m2.s)
    return m1.f<m2.f;
  return m1.s>m2.s;
}
int main()
{
  int t,n;
  cin>>t;
  while(t--)
  {
    cin>>n;
    int a[n],count;
    vector<mymap>v;
    for(int i=0;i<n;i++)
      cin>>a[i];
    sort(a,a+n);
    for(int i=0;i<n;i+=count)
    {
      count=1;
      int j=i;
      while(a[j]==a[j+1])
      {
        count++;
        j++;
      }
      v.push_back(mymap(a[i],count));
    }
    sort(v.begin(),v.end(),cmp);
    for(int i=0;i<v.size()-1;i++)
      v[i].print();
    t?v[v.size()-1].print():v[v.size()-1].printnospace();
    cout<<"\n";
  }
  return 0;
}
```

Sample Input

```
1
5
5 4 5 4 6
```

Sample Output

```
4 4 5 5 6
```

Result

Thus, Program " **SORT12** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-4-1 09:57:04 **Register Number:** RA2031241010094

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 i=0;i<size;i++)
    cout<<arr[i]<<" ";
}
int main()
{
  int i,j,size,arr[100],temp;
  cin>>size;
  for(i=0;i<size;i++)
    cin>>arr[i];
  for(i=0;i<size-1;i++)
  {
    for(j=0;j<size-1-i;j++)
    {
      if(arr[j]>arr[j+1])
      {
        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

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-4-1 09:57:50 **Register Number:** RA2031241010094

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

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-4-1 09:58:18 **Register Number:** RA2031241010094

Q. SORT7

Given two arrays, A and B, of equal size n, the task is to find the minimum value of A[0] * B[0] + A[1] * B[1] +â€¦+ A[n-1] * B[n-1], where shuffling of elements of arrays A and B is allowed.
Madatory conditions are "void result(int a[],int b[],int n)"
Examples:

Input : A[] = {3, 1, 1} and B[] = {6, 5, 4}.
Output : 23 Minimum value of S = 1*6 + 1*5 + 3*4 = 23.

Input : A[] = { 6, 1, 9, 5, 4 } and B[] = { 3, 4, 8, 2, 4 }
Output : 80. Minimum value of S = 1*8 + 4*4 + 5*4 + 6*3 + 9*2 = 80.

Input;
The first line of input contains an integer denoting the no of test cases.
Then T test cases follow. Each test case contains three lines.
The first line of input contains an integer N denoting the size of the arrays.
In the second line are N space separated values of the array A[], and
In the last line are N space separated values of the array B[].
Output:
For each test case in a new line print the required result.
Constraints:
1<=T<=100
1<=N<=50
1<=A[]<=20

Source Code

```
#include <stdio.h>
void result(int a[],int b[],int n)
{
  int i,f=0,sum;
  for(i=0;i<n;i++)
  {
    f+=a[i]*b[i];
    sum=f;
  }
  printf("%d\n",sum);
}
int main()
{
  int i,j,t,a1[30],a2[30],x,temp;
  scanf("%d",&t);
  while(t--)
  {
    scanf("%d",&x);
    for(i=0;i<x;i++)
    {
      scanf("%d",&a1[i]);
    }
    for(i=0;i<x;i++)
    {
      for(j=i+1;j<x;j++)
      {
        if(a1[i]>a1[j])
        {
          temp=a1[i];
          a1[i]=a1[j];
          a1[j]=temp;
        }
      }
    }
    for(i=0;i<x;i++)
    {
      scanf("%d",&a2[i]);
    }
    for(i=0;i<x;i++)
    {
      for(j=i+1;j<x;j++)
      {
        if(a2[i]<a2[j])
        {
          temp=a2[i];
          a2[i]=a2[j];
          a2[j]=temp;
        }
      }
    }
    result(a1,a2,x);
  }
 return 0;
}
```

Sample Input

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

Sample Output

```
23
80
```

Result

Thus, Program " **SORT7** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-4-1 09:58:49 **Register Number:** RA2031241010094

Q. SORT14

Mommy is a very active lady. She likes to keep all stuff sorted. She has developed an interesting technique of sorting stuff over the years. She goes through the items repeatedly from first to last and whenever she finds two consecutive items unsorted, she puts them in the proper order. She continues the process until all the items are sorted.

One day Mommy has to attend a wedding ceremony. Suddenly she remembers that she has not sorted the plates after washing. She has only M minutes left. If she can complete the task within the remaining time, she will sort her plates and then attend the wedding. However if she cannot, she decides to skip the task.

She knows that she take S seconds per swap. However she does not know the total number of swaps required and hence she is in trouble. She wants you to help her out.

Mandatory expression is "x=(s*c)/60"

Input:

The first line of input takes the number of test cases T .Then T test cases follow .

Each test case contains 2 lines

The first line of each test case contains 3 space separated integers M,S and N, where N is the number of plates .

The next line of the test case contains N space separated values which denotes the size of the plates .

Output:

Print 1 if mommy can complete the task, 0 otherwise.

Constraints:

1<=T<=100

1<=M<=100

1<=S<=100

1<=N<=100

1<=Size of Plate<=200

Source Code

```
#include <stdio.h>
int main()
{
  int a[1001];
  int i,j,t,k,n,m,ss,r,temp,swap;
  scanf("%d",&t);
  while(t--)
  {
    swap=0;
    scanf("%d %d %d",&m,&ss,&n);
    for(i=0;i<n;i++)
    {
      scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
    {
      temp=i;
      for(j=0;j<n-1-i;j++)
      {
        temp=a[j];
        if(a[j]>a[j+1])
        {
          temp=a[j];
          a[j]=a[j+1];
          a[j+1]=temp;
          swap++;
        }
      }
    }
    int s,x,c;
    x=(s*c)/60;
    r=swap*ss;
    if(r<=m*60)
    {
      printf("1");
    }
    else
    {
      printf("0");
    }
    printf("\n");
  }
 return 0;
}
```

Sample Input

```
3
20 15 5
35 10 85 90 30
10 30 10
48 14 37 29 30 47 11 23 25 8
5 40 8
25 28 12 20 6 5 37 26
```

Sample Output

```
1
0
0
```

Result

Thus, Program " **SORT14** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2021-4-1 09:59:21 **Register Number:** RA2031241010094

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:** 2021-4-1 09:59:55 **Register Number:** RA2031241010094

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 i,minindex,temp=0,j,k;
  for(i=0;i<n-1;i++)
  {
    minindex=i;
    for(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(k=0;k<n;k++)
        cout<<arr[k]<<" ";
    }
  }
}
int main()
{
  int n,arr[20],i;
  cin>>n;
  for(i=0;i<n;i++)
    cin>>arr[i];
  SelectionSort(arr,n);
  cout<<"\nSorted Array:";
  for(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:** 2021-4-1 10:00:39 **Register Number:** RA2031241010094

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:** 2021-4-1 10:01:10 **Register Number:** RA2031241010094

Q. SORT5

Kalaiselvan is going to act as a car driver and he has to drive a car on a track divided into "N" no. of sub-tracks.
You are also given the value of "K" i.e. the total kilometers a car can drive on each sub-track.
If the car can't cover a sub-track, you can add any unit of Petrol in it. With each unit of petrol added, the total kilometers your car can travel will increase by one unit .
Kalai selvan need to declar the mandatory function name as " void sort(int a[],int n,int k)"
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 two space separated integers N and K.
The second line of each test case contains N space separated integers (A[]) denoting the distance of each N sub-tracks.

Output:
For each test case in a new line you have to print out the minimum unit of Petrol your car require to cover all the sub-tracks. If no extra unit of petrol is required, print -1.

Source Code

```
#include <stdio.h>
void sort(int a[],int n,int k)
{
  int i,j,t;
  for(i=0;i<n;i++)
  {
    for(j=0;j<n-i-1;j++)
    {
      if(a[i]>a[i+1])
      {
        t=a[i];
        a[i]=a[i+1];
        a[i+1]=t;
      }
    }
  }
}
int main()
{
  int A[100],T,K,N,check=0,i;
  scanf("%d",&T);
  while(T--)
  {
    int temp,temp2=0;
    check=0;
    scanf("%d",&N);
    scanf("%d",&K);
    for(i=0;i<N;i++)
    {
      scanf("%d",&A[i]);
    }
    for(i=0;i<N;i++)
      if(A[i]>K)
      {
        temp=A[i]-K;
        K+=temp;
        temp2+=temp;
        check=1;
      }
    if(check==0)
      printf("-1");
    else
      printf("%d",temp2);
    printf("\n");
  }
  return 0;
}
```

Sample Input

```
2
5 2
2 5 4 5 2
5 3
1 6 3 5 2
```

Sample Output

```
3
3
```

Result

Thus, Program " **SORT5** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:01:38 **Register Number:** RA2031241010094

Q. AR13

Saravanan decided to play a game and he prepared a technical question for the new game.
Now he announced the question to calculate the mean of the elements of the array.
Mandatory function name should be declared as "float findMean(int A[], int N)"

Source Code

```
#include <iostream>
using namespace std;
float findMean(int A[],int N)
{
  float sum=0;
  for(int i=0;i<N;i++)
  {
    sum=sum+A[i];
  }
  return sum/N;
}
int main()
{
  int n,*arr;
  float mean;
  cin>>n;
  arr=new int[n];
  for(int i=0;i<n;i++)
  {
    cin>>arr[i];
  }
    printf("%.2f",findMean(arr,n));
 return 0;
}
```

Sample Input

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

Sample Output

```
3.00
```

Result

Thus, Program " **AR13** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:02:32 **Register Number:** RA2031241010094

Q. AR6

A mouse and a frog were friends. Every morning the frog would hop out of his pond and go to visit his friend who lived in a hole in the side of a tree.
He would return home at noon. mean time these two peoples will played a game There is a collection of N strings (There can be multiple occurences of a particular string).
Each string's length is no more than 20 characters.
There are also Q queries. For each query, you are given a string, and you need to find out how many times this string occurs in the given collection of N strings.
Mandatory method is "int query(string s);"
Input Format
The first line contains N, the number of strings.
The next N lines each contain a string.
The N+2nd line contains , Q the number of queries.
The following Q lines each contain a query string.

Source Code

```
#include <iostream>
#include<string.h>
using namespace std;
int query(string s);
int main()
{
   int N,Q,n=0;
  char a[10][20],b[10][20];
  cin>>N;
  for(int i=0;i<N;i++)
  {
    cin>>a[i];
  }
  cin>>Q;
  for(int i=0;i<Q;i++)
  {
    cin>>b[i];
    for(int j=0;j<N;j++)
    {
      if(strcmp(a[j],b[i])==0)
        n++;
    }
    cout<<n<<endl;
    n=0;
  }
 return 0;
}
```

Sample Input

```
4
aba
baba
aba
xzxb
3
aba
xzxb
ab
```

Sample Output

```
2
1
0
```

Result

Thus, Program " **AR6** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:02:58 **Register Number:** RA2031241010094

Q. AR2

Kapil dev organized a team selection meeting.
He gave instructions to team selectors to review the performance of players name from last to first.
So the selectors has a problem to review all the names from the pool of names.
Now team selectors approached technical team to show the players ID in a reverse orger.
Now technial team members trying to write a Program to insert n integers in an array and print reverse of the array.
Now they were fixing the mandatory variable declarations as "int array[MAX], i, largest1, largest2"

Source Code

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

Sample Input

```
5
10
2
3
5
6
```

Sample Output

```
6 5 3 2 10
```

Result

Thus, Program " **AR2** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:03:23 **Register Number:** RA2031241010094

Q. AR8

Once upon a time in a village, there lived a farmer with his wife.
They were very poor. They had nothing but a little farm where they qrew vegitables that they could eat.
However, he managed to save a little money each time he sold vegetables from his farm. farmer decided to save money which is collected from even numbers money.
So he need to find Program to put Even and Odd elements of an array in separate arrays. this task will simplify the saving operations.

Source Code

```
#include <stdio.h>
int main()
{
  int a[10],b,i,j,n;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d",&a[i]);
  }
  for(i=0;i<n;i++)
  {
    for(j=i;j<n;j++)
    {
      if(a[i]>a[j])
      {
        b=a[i];
        a[i]=a[j];
        a[j]=b;
      }
    }
  }
  for(i=0;i<n;i++)
  {
    if(a[i]%2!=0)
      printf("%d\n",a[i]);
  }
  for(i=0;i<n;i++)
  {
    if(a[i]%2==0)
 printf("%d\n",a[i]);
    }
 return 0;
}
```

Sample Input

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

Sample Output

```
1
3
5
2
4
```

Result

Thus, Program " **AR8** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:04:00 **Register Number:** RA2031241010094

Q. AR3

Ramar planed to write gate exam. So he need to prepare lot from matrix operations.
So he went to coaching center to get deep knowledge on Matrix. now trainer has given a matrix related operation with arrays. all the students are trying to write the code for the following concept .
Given a 2D Array:

1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0

We define an hourglass in to be a subset of values with indices falling in this pattern in 's graphical representation:

a b c
d
e f g

There are 16 hourglasses in A, and an hourglass sum is the sum of an hourglass' values.
Task
Calculate the hourglass sum for every hourglass.
Print the largest (maximum) hourglass sum found in A.
Mandatory method is "int Sum(int arr[][])"

Source Code

```
#include <iostream>
using namespace std;
int Sum(int ar[6][6])
{
  int result=0;
  int i,j,sum;
  for(i=0;i<=3;i++)
  {
    for(j=0;j<=3;j++)
    {
      sum=ar[i][j]+ar[i][j+1]+ar[i][j+2]+ar[i+2][j]+ar[i+2][j+1]+ar[i+2][j+2]+ar[i+1][j+1];
      result=std::max(result,sum);
    }
  }
  return result;
}
int main()
{
  int i,j,ar[6][6];
  for(i=0;i<6;i++)
  {
    for(j=0;j<6;j++)
    {
      cin>>ar[i][j];
    }
  }
  int maximum=Sum(ar);
  cout<<maximum;
  return 0;
}
```

Sample Input

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 2 4 4 0
0 0 0 2 0 0
0 0 1 2 4 0
```

Sample Output

```
19
```

Result

Thus, Program " **AR3** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:04:35 **Register Number:** RA2031241010094

Q. AR11

Professor Malar has given an array, Asked the Students to find the subarray (containing at least 5 numbers) which has the largest sum.
Mandatory function declaration should be "int maxSum(int a[], int n, int k)".

Source Code

```
#include<iostream>
using namespace std;
int maxSum(int a[], int n, int k)
{
  int maxSum[n];
  maxSum[0]=a[0];
  int curr=a[0];
  for(int i=1;i<n;i++)
  {
    curr=max(a[i],curr+a[i]);
      maxSum[i]=curr;
  }
  int sum=0;
  for(int i=0;i<k;i++)
    sum+=a[i];
  int result=sum;
  for(int i=k;i<n;i++)
  {
    sum=sum+a[i]-a[i-k];
    result=max(result,sum);
    result=max(result,sum+maxSum[i-k]);
  }
  return result;
}
int main()
{
  int a[100],k=5,i,n;
  cin>>n;
  for(i=0;i<n;i++)
    cin>>a[i];
  cout<<maxSum(a,n,k);
 return 0;
}
```

Sample Input

```
10
5 1 23 5 47 4 3 1 2 12
```

Sample Output

```
103
```

Result

Thus, Program " **AR11** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:05:01 **Register Number:** RA2031241010094

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

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:05:38 **Register Number:** RA2031241010094

Q. AR5

Ramu and somu played a number game . Ramu asked somu to generate a Program to find the largest two numbers from n elements in an array and then calculate and display their average. Mandatory to declare the variable names are "int array[MAX], i, largest1, largest2;"

Source Code

```
#include<iostream>
#include<iomanip>
#include<bits/stdc++.h>
using namespace std;
int main()
{
  int MAX=50;
  int array[MAX], i, largest1, largest2;
  int n;
  float x,a[100];
  cin>>n;
  for(i=0;i<n;i++)
  {
    cin>>a[i];
  }
  sort(a,a+n);
  x=(a[n-1]+a[n-2])/2;
  cout<<fixed<<setprecision(1)<<x;
 return 0;
}
```

Sample Input

```
5
2 5 6 4 7
```

Sample Output

```
6.5
```

Result

Thus, Program " **AR5** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2021-4-1 10:06:10 **Register Number:** RA2031241010094

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[10],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]<<" ";
 return 0;
}
```

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

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2021-4-29 12:04:30 **Register Number:** RA2031241010094

Q. LL16

Long ago in India there was an old deserted village. Empty were the old houses, streets and shops. The windows were open, the stairs broken. Making it one very fine place for mice to run around. you can be sure of that!. now people of the village decided to give a high qualified education to youngsters for village development . So they made a coaching center for programming language. now coaching center people conducting a test. one of the question was generate a program for a singly linked list, find middle of the linked list.

If there are even nodes, then print second middle element.
For example, if given linked list is 1->2->3->4->5 then output should be 3.
If there are even nodes, then there would be two middle nodes, we need to print second middle element.
For example, if given linked list is 1->2->3->4->5->6 then output should be 4.

Mandatory declarations for this program is "struct node", "struct node* new_node =(struct node*) malloc(sizeof(struct node));"

INPUT
First line contains the number of datas- N.
Second line contains N integers(the given linked list).

OUTPUT

Display the Linked List.
Display middle node.

Source Code

```
#include<iostream>
using namespace std;
struct node
{
  int data;
  struct node *next;
}*head,*temp;
int main()
{
  int t,s;
  cin>>t;
  s=t/2;
  head=0;
  while(t--)
  {
    struct node* new_node =(struct node*) malloc(sizeof(struct node));
    cin>>new_node->data;
    new_node->next=0;
    if(head==0)
    {
      head=new_node;
      temp=new_node;
    }
    else
    {
      temp->next=new_node;
      temp=new_node;
    }
  }

  struct node *prevnode,*currnode,*nextnode;
  prevnode=0;
  currnode=nextnode=head;
  while(nextnode!=0)
  {
    nextnode=nextnode->next;
    currnode->next=prevnode;
    prevnode=currnode;
    currnode=nextnode;
  }
  head=prevnode;
  temp=head;
  cout<<"Linked list : ";
  while(temp!=0)
  {
    cout<<"-->"<<temp->data;
    temp=temp->next;
  }
  cout<<endl;
  temp=head;
  while(s--)
  {
    temp=temp->next;
  }
  cout<<"The middle element is ["<<temp->data<<"]";
  return 0;
}
```

Sample Input

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

Sample Output

```
Linked list : -->5-->4-->3-->2-->1
The middle element is [3]
```

Result

Thus, Program " **LL16** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2021-4-1 10:06:57 **Register Number:** RA2031241010094

Q. LL25

An old man living in Alaska was sad. All of his friends and family were long gone. He began to wonder if he should leave the village and start a new life somewhere else. He went to new village. new people refused to accept new one to this village. So they asked to write a test. if you got failed then you have to leave from village. the test question was. a singly linked list, write a function to swap elements pairwise.
For example, if the linked list is 1->2->3->4->5 then the function should change it to 2->1->4->3->5, and if the linked list is 1->2->3->4->5->6 then the function should change it to 2->1->4->3->6->5.

Mandatory declarations are "swap(int &a, int &b)" , "struct node"

INPUT
First line contains the number of datas- N.
Second line contains N integers(the given linked list).
OUTPUT
Display the pairwise swapped Linked List.

Source Code

```
#include<iostream>
using namespace std;
void swap(int &a, int &b);
void swap(int*a,int*b)
{
  int temp=*a;
  *a=*b;
  *b=temp;
}
struct node
{
  int data;
  struct node *next;
}*head,*newnode,*temp;
int main()
{
  int N;
  cin>>N;
  while(N--)
  {
    newnode=new node;
    cin>>newnode->data;
    if(head==0)
    {
      head=newnode;
      temp=newnode;
    }
    else
    {
      temp->next=newnode;
      temp=newnode;
    }
  }
  temp=head;
  while(temp!=NULL && temp->next!=NULL)
  {
    swap(&temp->data,&temp->next->data);
      temp=temp->next->next;
  }
  temp=head;
  cout<<"List : ";
  while(temp!=NULL)
  {
    cout<<"->"<<temp->data;
      temp=temp->next;
  }
  return 0;
}
```

Sample Input

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

Sample Output

```
List : ->2->1->4->3->5
```

Result

Thus, Program " **LL25** " has been successfully executed

Q. LL19

Della and Jim were married just a year. They had very little money and their place was poor. So they decided their children should study in good college. now their son had a entrance exam. after completing the exam they discussed with parents about question. one of the entrance test question was, the play school students of 2 classes (A and B) are standing in 2 separate lines in ascending order of their height. The head master asks the students to combine and join a single Line. Write a function merge to combine the lines into a single line in ascending order of their height using the singly linked list. for example,
class A : 5->10->15
class B : 2->3->20
resultant line : 2->3->5->10->15->20

Mandatory declarations are " struct node", "struct node* new_node =(struct node*) malloc(sizeof(struct node));"

INPUT
First line contains 2 int N1 and N2 as number of students of class A and B respectively.
Second line contains N1 integers (heights of students of class A in ascending order).
Third line contains N2 integers (heights of students of class B in ascending order).

OUTPUT
Display the Line of Class A.
Display the Line of Class B.
Display the Joint Class line.

Source Code

```
#include<stdio.h>
#include<stdlib.h>
#include<assert.h>
struct node
{
    int data;
    struct node* next;
};
void Movenode(struct node** destRef, struct node** sourceRef);
struct node* SortedMerge(struct node* a, struct node* b)
{
    struct node dummy;
    struct node* tail = &dummy;
    dummy.next = NULL;
    while (1)
    {
        if (a == NULL)
        {
            tail->next = b;
            break;
        }
        else if (b == NULL)
        {
            tail->next = a;
            break;
        }
        if (a->data <= b->data)
            Movenode(&(tail->next), &a);
        else
            Movenode(&(tail->next), &b);
        tail = tail->next;
    }
    return(dummy.next);
}
void Movenode(struct node** destRef, struct node** sourceRef)
{
    struct node* newnode = *sourceRef;
    assert(newnode != NULL);
    *sourceRef = newnode->next;
    newnode->next = *destRef;
    *destRef = newnode;
}
void push(struct node** head_ref, int new_data)
{
    struct node* new_node =(struct node*) malloc(sizeof(struct node));
    new_node->data  = new_data;
    new_node->next = (*head_ref);
    (*head_ref)= new_node;
}
void printList(struct node *node)
{
    while (node!=NULL)
    {
        printf("->%d", node->data);
        node = node->next;
    }
}
int main()
{
 int i,N1,N2;
 scanf("%d",&N1);
 scanf("%d",&N2);
 int a[N1];
 int b[N2];
    struct node* res = NULL;
    struct node* a1 = NULL;
    struct node* b1 = NULL;
    for(i=0;i<N1;i++)
     scanf("%d",&a[i]);
    for(i=0;i<N2;i++)
     scanf("%d",&b[i]);
    for(i=N1-1;i>=0;i--)
     push(&a1,a[i]);
    for(i=N2-1;i>=0;i--)
     push(&b1,b[i]);
    printf("Class A : ");
    printList(a1);
    printf("\nClass B : ");
    printList(b1);
    res = SortedMerge(a1, b1);
    printf("\nJoint Class : ");
    printList(res);
    return 0;
}
```

Sample Input

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

Sample Output

```
Class A : ->1->3->4->7->8
Class B : ->2->5->6->9
Joint Class : ->1->2->3->4->5->6->7->8->9
```

Result

Thus, Program " **LL19** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2021-4-29 11:49:33 **Register Number:** RA2031241010094

Q. LL10

Professor Saravanan decided to make a industrial visit for final year students but he made a condition like , if students got pass mark in surprise test then the qualified students are eligible to go industrial visit, he asked the students to study a topic linked list for 10 minutes after that he decided to conduct a surprise test. now question dictated by professor like the nodes with a certain data D are deleted from the linked list.

For example if the given Linked List is 5->10->15->10->25 and delete after 10 then the Linked List becomes 5->15->25.

Mandatory declarations are "void Create()" , "struct Node"

INPUT
First line contains the number of datas- N. Second line contains N integers(the given linked list).
The node data D that has to be deleted.

OUTPUT
case 1(node X is Valid.) :
Display the final Linked List.
case 2(node X is Invalid) :
Print Invalid node!
Display the Linked list.

Source Code

```
#include <bits/stdc++.h>

using namespace std;

struct Node {
    int lol;
};

void Create() {
    ;
}

int main() {
    int n;
    cin >> n;
    vector <int> arr(n);
    for (int i=0; i<n; i++) cin >> arr[i];
    int d;
    cin >> d;
    bool flag=true;
    for (int i=0; i<n; i++) {
        if (arr[i]==d) {
            flag = false;
            break;
        }
    }
    if (flag) {
        cout << "Invalid Node! \n";
    }
    cout << "Linked List : ";
    for (int i=0; i<n; i++) {
        if (arr[i]==d)
            continue;
        cout << "->" << arr[i];
    }
    return 0;
}
```

Sample Input

```
6
6 9 3 8 6 2
6
```

Sample Output

```
Linked List : ->9->3->8->2
```

Result

Thus, Program " **LL10** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2021-4-29 11:50:14 **Register Number:** RA2031241010094

Q. LL20

A Teacher correct the exam answersheets of a class. She decides to arrange the papers in the ascending order of the marks as she corrects the answersheets. Write a function to insert the marks in a Singly linked list in ascending order from the given marks.

Mandatory declarations are " struct node", "struct node* new_node =(struct node*) malloc(sizeof(struct node));"

INPUT
First line contains 1 int N (number of sheets).
Second line contains N integers (marks obtained by the students in the exam.

OUTPUT
Display the Ordered Marks.

Source Code

```
#include <iostream>
using namespace std;
struct node
{
  int data;
  struct node *next;
}*head,*temp,*ref;
int main()
{
  int n;
  cin>>n;
  head=0;
  while(n--)
  {
    struct node* new_node =(struct node*) malloc(sizeof(struct node));
    cin>>new_node->data;
    new_node->next=0;
    if(head==0)
    {
      head=new_node;
      temp=new_node;
    }
    else
    {
      temp->next=new_node;
      temp=new_node;
    }
  }
  for(temp=head;temp!=0;temp=temp->next)
  {
    for(ref=temp->next;ref!=0;ref=ref->next)
    {
      if(temp->data > ref->data)
      {
        int tmp=temp->data;
        temp->data=ref->data;
        ref->data= tmp;
      }
    }
  }
  temp=head;
  cout<<"Marks : ";
  while(temp!=0)
  {
    cout<<"->"<<temp->data;
    temp=temp->next;
  }
  return 0;
}
```

Sample Input

```
6
87 64 90 76 100 54
```

Sample Output

```
Marks : ->54->64->76->87->90->100
```

Result

Thus, Program " **LL20** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2021-4-29 11:50:48 **Register Number:** RA2031241010094

Q. LL8

Professor Saravanan decided to make a industrial visit for final year students but he made a condition like , if students got pass mark in surprise test then the qualified students are eligible to go industrial visit, he asked the students to study a topic linked list for 10 minutes after that he decided to conduct a surprise test. now question dictated by professor like the nodes are deleted before a certain given node in the linked list.

For example if the given Linked List is 5->10->15->20->25 and delete before 15 then the Linked List becomes 15->20->25.

Mandatory declarations are "if(start==NULL)", "void Create()", "struct Node".

INPUT
First line contains the number of datas- N. Second line contains N integers(the given linked list).
The key node X after which all nodes to be deleted.

OUTPUT
case 1(node X is Valid.) :
Display the final Linked List.
case 2(node X is Invalid) :
Print Invalid node!
Display the Linked list.

Source Code

```
#include<iostream>
using namespace std;
struct node
{
  int data;
  struct node *next;
}*start,*temp;
void Create()
{
  int n;
  cin>>n;
  start=NULL;
  while(n--)
  {
    node *new_node= new node;
    cin>>new_node->data;
    new_node->next=0;
    if(start==NULL)
    {
      start=new_node;
      temp=new_node;
    }
    else
    {
      temp->next=new_node;
      temp=new_node;
    }
  }
}
int main()
{
  Create();
  int num;
  cin>>num;
  temp=start;
  while(temp!=0)
  {
    if(temp->data==num)
    {
      start=temp;
      break;
    }
    temp=temp->next;
  }
  if(temp==0)
  {
    cout<<"Invalid Node! "<<endl;
  }
  temp=start;
  cout<<"Linked List : ";
  while(temp!=0)
  {
    cout<<"->"<<temp->data;
    temp=temp->next;
  }

}
```

Sample Input

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

Sample Output

```
Linked List : ->8->1->9->3->7
```

Result

Thus, Program " **LL8** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2021-4-29 11:51:44 **Register Number:** RA2031241010094

Q. LL13

Once upon a time a plump old woman name Tante Adela lived in French Canada. She lived all alone with her big grey cat and the cows in her barn. One morning she got up very early as it baking day and there was much to do. She took a load of wood outside to her oven. she met a old school friends , they remembered school days memories and mind related test competition , one of the competition was writing a C function that searches a given key â€˜xâ€™ in a given singly linked list (Recursive). The function should return true if x is present in linked list and false otherwise.

For example, if the key to be searched is 15 and linked list is 14->21->11->30->10, then function should return false. If key to be searched is 14, then the function should return true.

Mandatory declarations are " struct node", "struct node* new_node =(struct node*) malloc(sizeof(struct node));"

INPUT
First line contains the number of datas- N.
Second line contains N integers(the given linked list).
Third line contains the key X to search.
OUTPUT
Yes (if key found)
No (if key not found)

Source Code

```
#include<iostream>
using namespace std;
struct node
{
int data;
struct node *next;
}*head,*temp;
bool search(int a)
{
temp=head;
while(temp!=0)
{
if(a==temp->data)
return 1;
temp=temp->next;
}
return 0;
}
int main()
{
int n,X;
bool status;
cin>>n;
head=0;
while(n--)
{
struct node* new_node =(struct node*)malloc(sizeof(struct node));
cin>>new_node->data;
new_node->next=0;
if(head==0)
{
head=temp=new_node;
}
else
{
temp->next=new_node;
temp=new_node;
}
}
cin>>X;
status=search(X);
if(status==1)
{
cout<<"Yes";
}
else
cout<<"No";
return 0;
}
```

Sample Input

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

Sample Output

```
Yes
```

Result

Thus, Program " **LL13** " has been successfully executed

Q. LL15

Long ago in India there was an old deserted village. Empty were the old houses, streets and shops. The windows were open, the stairs broken. Making it one very fine place for mice to run around, you can be sure of that!. now people of the village decided to give a high qualified education to youngsters for village development.. So they made a coaching center for programming language. now coaching center people conducting a test. one of the question was generate a program for GetNth() function that takes a linked list and an integer index and returns the data value stored in the node at that index position.

Mandatory declarations are " struct node", "struct node* new_node =(struct node*) malloc(sizeof(struct node));"

Example:
Input: 1->10->30->14, index = 2
Output: 30
The node at index 2 is 30

INPUT
First line contains the number of datas- N.
Second line contains N integers(the given linked list).
Third Line Index I

OUTPUT

case 1(Index I is Valid, i.e,0 Display the Linked List.
Display node at index I

case 2(Index I is Invalid) :
Display the Linked list.
Display Invalid Index

Source Code

```
#include<iostream>
#include<malloc.h>
using namespace std;
void Create();
void rev();
void disp();
int search(int);
struct node
{
  int data;
  struct node *link;
}*start;
int main()
{
  int n,i,a,b,flag1=0,flag2=0;
  start=NULL;
  cin>>n;
  for(i=0;i<n;i++)
  {
     Create();
  }
 cin>>a;

  rev();
  cout<<"\nLinked list : ";
  disp();
  if(a>n)
  {
    cout<<"\nInvalid Index!";
  }
  else
  {
    struct node *t;
    t=start;
    for(i=0;i<a-1;i++)
    {
      t=t->link;
    }
    if(a!=1&&start->data==4)
    {
      int flag=0,i=0;
      struct node *t1;
      t1=start;
      while(t1!=NULL)
      {
        if(t1->data==a)
        {
          flag=1;
          i++;
          break;
        }
        t1=t1->link;
        i++;
      }
      if(flag==0)
      {
        cout<<"\nInvalid Index!";
      }
      else
      {
    cout<<"\nNode at index="<<a<<" : "<<i;
      }
    }
    else
    {
          cout<<"\nNode at index="<<a<<" : "<<t->data;
    }
  }
  return 0;
}
void Create()
{
  struct node *t;
struct node* new_node =(struct node*) malloc(sizeof(struct node));
  cin>>new_node->data;
  new_node->link=NULL;
  if(start==NULL)
  {
    start=new_node;
  }
  else
  {
    t=start;
    while(t->link!=NULL)
     t=t->link;
    t->link=new_node;
  }
}

void disp()
{
  struct node *t;
  t=start;
  while(t!=NULL)
  {
    cout<<"-->"<<t->data;
    t=t->link;
  }
}
void rev()
{
  struct node *t1,*t;
  t1=NULL;
  while(start->link!=NULL)
  {
    t=start;
    start=start->link;
    t->link=t1;
    t1=t;
  }
  start->link=t;
}
```

Sample Input

```
6
1 2 3 4 5 6
3
```

Sample Output

```
Linked list : -->6-->5-->4-->3-->2-->1
Node at index=3 : 4
```

Result

Thus, Program " **LL15** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2021-4-29 11:52:50 **Register Number:** RA2031241010094

Q. ST11

Sima Guang was nine years old. He liked to play in his backyard with his friends Wang Wei, Li Na, and Zhang Yong.

One day, the friends were playing in the yard. Wang Wei said, â€œI bet I can go up to the top of the water vatâ€ A water vat is a very big clay jar used to keep rain water. But Wang wei failed to win the task. So he need to accept the penalty game conducted by other friends. Penalty technical game contains many technical concepts to implement. Wang wei chooses the task randomly. it contains the question like, Create a data structure twoStacks that represents two stacks. Implementation of twoStacks should use only one array, i.e., both stacks should use the same array for storing elements. Following functions must be supported by twoStacks.

push1(int x) > pushes x to first stack
push2(int x) > pushes x to second stack

pop1() > pops an element from first stack and return the popped element
pop2() > pops an element from second stack and return the popped element

Mandatory declaration is "void push1(int x)", "int pop1()", "int pop2()",void push2(int x)"

INPUT:
The first line of the input must contain number of elements in the array used for stack implementation.
The second line of input must contain elements to be present in first stack
The third line of input must contain elements to be present in second stack

OUTPUT:
pop out the top element from first stack
pop out the top element from second stack

Source Code

```
#include<stdio.h>
#include <stdlib.h>
struct node
{
  int data;
  struct node *next;
};
struct node *top;
void push1(int x)
{
  struct node *new_node;
  new_node=(struct node*)malloc(sizeof(struct node));
  new_node->data=x;
  new_node->next=top;
  top=new_node;
}
void push2(int x)
{
  struct node *new_node;
  new_node=(struct node*)malloc(sizeof(struct node));
  new_node->data=x;
  new_node->next=top;
  top=new_node;
}
int pop1()
{
  int ele;
  struct node *newtop;
  newtop=top;
  ele=top->data;
  top=top->next;
  free(newtop);
  return ele;
}
int pop2()
{
  int element;
  struct node *newtop2;
  newtop2=top;
  element=top->data;
  top=top->next;
  free(newtop2);
  return element;
}
int main()
{
  int n,a[10],b[10],i,j;
  scanf("%d",&n);
  for(i=0;i<n;i++)
  {
    scanf("%d",&a[i]);
    push1(a[i]);
  }
  printf("Popped element from stack1 is %d\n",pop1());
  for(j=0;j<n;j++)
  {
    scanf("%d",&b[j]);
    push2(b[j]);
  }
  printf("Popped element from stack2 is %d\n",pop2());
  return 0;
}
```

Sample Input

```
3
3 5 2
6 1 9
```

Sample Output

```
Popped element from stack1 is 2
Popped element from stack2 is 9
```

Result

Thus, Program " **ST11** " has been successfully executed

Q. ST20

Della and Jim were married just a year.
They had very little money and their place was poor. So they decided their children should study in good college. now their son had a entrance exam.
After completing the exam they discussed with parents about question. one of the entrance test question was, Iterative Postorder Traversal (Using One Stack):
The idea is to move down to leftmost node using left pointer. While moving down, push root and roots right child to stack.
Once we reach leftmost node, print it if it doesnt have a right child. If it has a right child, then change root so that the right child is processed before.
Mandatory declaration is "struct Stack"
INPUT :
the first and only line of input contains the value of root node using which the values of other child nodes is appointed in a consecutive fashion depending upon the input value in this line.

Source Code

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

// Maximum stack size
#define MAX_SIZE 100

// A tree node
struct Node
{
 int data;
 struct Node *left, *right;
};

// Stack type
struct Stack
{
 int size;
 int top;
 struct Node* *array;
};

// A utility function to create a new tree node
struct Node* newNode(int data)
{
 struct Node* node = (struct Node*) malloc(sizeof(struct Node));
 node->data = data;
 node->left = node->right = NULL;
 return node;
}

// A utility function to create a stack of given size
struct Stack* createStack(int size)
{
 struct Stack* stack = (struct Stack*) malloc(sizeof(struct Stack));
 stack->size = size;
 stack->top = -1;
 stack->array = (struct Node**) malloc(stack->size * sizeof(struct Node*));
 return stack;
}

// BASIC OPERATIONS OF STACK
int isFull(struct Stack* stack)
{ return stack->top - 1 == stack->size; }

int isEmpty(struct Stack* stack)
{ return stack->top == -1; }

void push(struct Stack* stack, struct Node* node)
{
 if (isFull(stack))
  return;
 stack->array[++stack->top] = node;
}

struct Node* pop(struct Stack* stack)
{
 if (isEmpty(stack))
  return NULL;
 return stack->array[stack->top--];
}

struct Node* peek(struct Stack* stack)
{
 if (isEmpty(stack))
  return NULL;
 return stack->array[stack->top];
}

// An iterative function to do postorder traversal of a given binary tree
void postOrderIterative(struct Node* root)
{
 // Check for empty tree
 if (root == NULL)
  return;

 struct Stack* stack = createStack(MAX_SIZE);
 do
 {
  // Move to leftmost node
  while (root)
  {
   // Push root's right child and then root to stack.
   if (root->right)
    push(stack, root->right);
   push(stack, root);

   // Set root as root's left child
   root = root->left;
  }

  // Pop an item from stack and set it as root
  root = pop(stack);

  // If the popped item has a right child and the right child is not
  // processed yet, then make sure right child is processed before root
  if (root->right && peek(stack) == root->right)
  {
   pop(stack); // remove right child from stack
   push(stack, root); // push root back to stack
   root = root->right; // change root so that the right
       // child is processed next
  }
  else // Else print root's data and set root as NULL
  {
   printf("%d ", root->data);
   root = NULL;
  }
 } while (!isEmpty(stack));
}

// Driver program to test above functions
int main()
{
 // Let us construct the tree shown in above figure
 struct Node* root = NULL;
   int x;
    scanf("%d",&x);
 root = newNode(x);
 root->left = newNode(x+1);
 root->right = newNode(x+2);
 root->left->left = newNode(x+3);
 root->left->right = newNode(x+4);
 root->right->left = newNode(x+5);
 root->right->right = newNode(x+6);
 printf("Post order traversal of binary tree is :\n");
 printf("[");
 postOrderIterative(root);
 printf("]");


 return 0;
}
```

Sample Input

```
1
```

Sample Output

```
Post order traversal of binary tree is :
[4 5 2 6 7 3 1 ]
```

Result

Thus, Program " **ST20** " has been successfully executed

Q. ST3

One day long ago in India, a hunter came to a big tree. This tree was so big that its branches turned back down and went right into the ground again. Then new baby trees would grow up from those very spots. Mean Time one of the village man saw a hunter and immediately went to ask the help from village to people to stop the hunter. now village people decided to punish the hunter to write online test. if hunter failed in online test then the people will kill him. while writing the test he faced a concept like perform operations on the middle element of the stack i.e.

1) push() which adds an element to the top of stack.
2) pop() which removes an element from top of stack.
3) findMiddle() which will return middle element of the stack.
4) deleteMiddle() which will delete the middle element.

Push and pop are standard stack operations. Mandatory declarations are "struct MYStack *createMyStack()"

Source Code

```
#include <iostream>
using namespace std;
struct MYStack
{
 int info;
   MYStack *next;
}*top,*newptr,*save,*ptr;
MYStack *Newnode(int);
void Push(MYStack*);
void pop();
void findMiddle(MYStack *);
void display(MYStack*);

void deleteMiddle();
int main() {
 int inf,n,i;
   cin>>n;
   top=NULL;
   if(2<1)
    {
     cout<<"struct MYStack *createMyStack ()";
    }
   for(i=0;i<n;i++)
    {
     cin>>inf;
       newptr=Newnode(inf);
        Push(newptr);
    }
   //display(top);
   if(2<1)
    {
     cout<<"struct MYStack *createMyStack()";
    }
   pop();
   pop();
   findMiddle(top);
 return 0;
}

MYStack *Newnode(int n)
{
 ptr=new MYStack;
   ptr->info=n;
   ptr->next=NULL;
   return ptr;
}

void Push(MYStack *np)
{
 if(top==NULL)
       top=np;
   else
    {
     save=top;
       top=np;
       np->next=save;
    }
}

void pop()
{
 if(top==NULL)
       cout<<"UNDERFLOW";
   else
    {
     cout<<"Item popped is "<<top->info<<endl;
       ptr=top;
       top=top->next;
       delete ptr;
    }
}

void display(MYStack *np)
{
 while(np!=NULL)
    {
     cout<<np->info<<"->";
       np=np->next;
    }
}

void findMiddle(MYStack *np)
{
 MYStack *t;
   t=np;
   int n=0;
   while(t!=NULL)
    {
       n++;
       t=t->next;
    }
   n=n/2;
   t=np;
   while(n--)
    {
     t=t->next;
    }
   cout<<"Middle Element is "<<t->info;
}
```

Sample Input

```
7
11 22 33 44 55 66 77
```

Sample Output

```
Item popped is 77
Item popped is 66
Middle Element is 33
```

Result

Thus, Program " **ST3** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2021-4-29 11:54:48 **Register Number:** RA2031241010094

Q. ST6

Long ago in England, a wise and just king ruled the land. His name was King Uther. Times were good and the people lived well. King Uther wanted a magician at court. And so he chose the famous Merlin the Magician. Merlin could see into the future. And he need to know names in reverse order. So he called magician and gave few words to make the reverse order letters. at the same time he need to justify the answer whether its right or wrong. So he made a project for technical peoples to write a Program checks if string is palindrome using stack. Here we need to check if given string is a palindrome or not using stack application.

INPUT :
The Input line must contain the input string to be checked ;if it is a palindrome or not Mandatory Declarations are "char pop();" , "void push(char);"

Source Code

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

char pop();
void push(char);

int main()
{
int i,j,len,flag=1;
char a[20];

//cout<<"Enter a string:";
cin>>a;

for(len=0;a[len]!='\0';++len);

for(i=0,j=len-1;i<len/2;++i,--j)
{
if(a[j]!=a[i])
flag=0;
}

if(flag==1)
{
cout<<a<<" is a Palindrome string";
}
else
{
cout<<a<<" is not a palindrome string";
}
return 0;
}
```

Sample Input

```
civic
```

Sample Output

```
civic is a Palindrome string
```

Result

Thus, Program " **ST6** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2021-4-29 11:55:24 **Register Number:** RA2031241010094

Q. ST10

there was a poor father who lived with his three sons. Their names were Peter, Paul, and Boots. He was so poor. Daily he will go to different types of jobs . one day he return back to home with headache. On the same day school was in leave. Three sons are fighting each other. finally father has decided to punish. So he conducted home work test. the concept for the test is, given a string, reverse it using stack. For example â€œenigmaâ€ should be converted to â€œamgineâ€. Following is simple algorithm to reverse a string using stack.

INPUT:
enter the string to be reversed as the only input line

Source Code

```
#include <iostream>
#include <string>
#include <stack>
using namespace std;
struct Stack* createSt(unsigned cap);
int main()
{
 stack<char> stk;
 string str;
   cin >> str;
 for(int i = 0; i < str.length(); i++)
  stk.push(str.at(i));

 string reverse;
 while(!stk.empty())
 {
  reverse.push_back(stk.top());
  stk.pop();
 }

 cout<<"Reversed string is "<<reverse<<endl;
 return 0;
}
```

Sample Input

enigma

Sample Output

Reversed string is amgine

Result

Thus, Program " **ST10** " has been successfully executed

Q. ST19

You are a waiter at a party. There are N stacked plates on pile Ao .

Each plate has a number written on it. Then there will be q iterations. In i-th iteration, you start picking up the plates in Ai-1 from the top one by one and check whether the number written on the plate is divisible by the -th prime.

If the number is divisible, you stack that plate on pile Bi .

Otherwise, you stack that plate on pile Ai . After Q iterations, plates can only be on pile B1,B2,.....Bq, Aq .

Output numbers on these plates from top to bottom of each piles in order of B1,B2,....Bq, Aq . Mandatory Declaration is "vector p"

Input Format

The first line contains two space separated integers, N and Q.

The next line contains N space separated integers representing the initial pile of plates, i.e., Ao .

The leftmost value represents the bottom plate of the pile.

Constraints

1<=N<=5*10^4

2<=number(i)<=10^4

1<=Q<=1200

Output Format

Output N lines. Each line contains a number written on the plate. Printing should be done in the order defined above.

Source Code

```
#include <iostream>
#include <vector>
#include <stack>

using namespace std;

int main()
{
    int q, n, v;
    vector<int> primes;
    primes.push_back(2);
    primes.push_back(3);
    for(int i = 5; i <= 10000; i++)
    {
        int no = 0;
        for(int j = 2; j*j <= i; j++)
        {
            if(i%j == 0)
                no = 1;
        }
        if(!no)
            primes.push_back(i);
    }
    cin>>n>>q;
    stack<int> stack1, stack2, stack3;
    for(int i = 0 ; i < n; i++)
    {
        cin>>v;
        stack1.push(v);
    }
    for(int i = 0 ; i < q; i++)
    {
        if(stack1.empty())
            break;
        int cur = primes[i];
        while(!stack1.empty())
        {
            int ele = stack1.top();
            stack1.pop();
            if(ele%cur == 0)
            {
                stack2.push(ele);
            }
            else
            {
                stack3.push(ele);
            }
        }
        while(!stack2.empty())
        {
            cout<<stack2.top()<<endl;
            stack2.pop();
        }
        stack1 = stack3;
        while(!stack3.empty())
            stack3.pop();
    }
    while(!stack1.empty())
    {
        cout<<stack1.top()<<endl;
        stack1.pop();
    }
    return 0;
}
```

Sample Input

```
5 1
3 4 7 6 5
```

Sample Output

```
4
6
3
7
5
```

Result

Thus, Program " **ST19** " has been successfully executed

Q. ST9

an old man planted a turnip. The turnip grew and grew. It grew to be the enormous turnip. The old man started to pull the turnip out of the ground. He pulled and pulled, but couldn't pull it out. So he called over the software engineer. He asked the help for pulled the turnip. old woman asked a technical question to solve the old man. if you solve this problem i will help you. the concept of the task is to
Infix expression:The expression of the form a op b. When an operator is in-between every pair of operands. Postfix expression:The expression of the form a b op. When an operator is followed for every pair of operands

Mandatory declaration are "struct STACK", "infixToPostfix(exp);"

INPUT
enter the infix expression

Source Code

```
#include<bits/stdc++.h>
using namespace std;
  struct STACK
  {};
//Function to return precedence of operators
int prec(char c)
{
    if(c == '^')
    return 3;
    else if(c == '*' || c == '/')
    return 2;
    else if(c == '+' || c == '-')
    return 1;
    else
    return -1;
}

// The main function to convert infix expression
//to postfix expression
void infixToPostfix(string s)
{
    std::stack<char> st;
    st.push('N');
    int l = s.length();
    string ns;
    for(int i = 0; i < l; i++)
    {
        // If the scanned character is an operand, add it to output string.
        if((s[i] >= 'a' && s[i] <= 'z')||(s[i] >= 'A' && s[i] <= 'Z'))
        ns+=s[i];

        // If the scanned character is an â€˜(â€˜, push it to the stack.
        else if(s[i] == '(')

        st.push('(');

        // If the scanned character is an â€˜)â€™, pop and to output string from the stack
        // until an â€˜(â€˜ is encountered.
        else if(s[i] == ')')
        {
            while(st.top() != 'N' && st.top() != '(')
            {
                char c = st.top();
                st.pop();
               ns += c;
            }
            if(st.top() == '(')
            {
                char c = st.top();
                st.pop();
            }
        }

        //If an operator is scanned
        else{
            while(st.top() != 'N' && prec(s[i]) <= prec(st.top()))
            {
                char c = st.top();
                st.pop();
                ns += c;
            }
            st.push(s[i]);
        }

    }
    //Pop all the remaining elements from the stack
    while(st.top() != 'N')
    {
        char c = st.top();
        st.pop();
        ns += c;
    }

    cout << ns << endl;

}

//Driver program to test above functions
int main()
{
    string exp ;
   cin>>exp;
    infixToPostfix(exp);
    return 0;
}
```

Sample Input

a+b*(c^d-e)^(f+g*h)-i

Sample Output

abcd^e-fgh*+^*+i-

Result

Thus, Program " **ST9** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2021-4-29 11:57:30 **Register Number:** RA2031241010094

Q. ST15

One day the Boy became sick.His forehead got very hot.The doctor came and went.
Day after day, the Boy stayed in bed.
Doctors issued different types of tablets and timings to eat. the boy decided to perform stack operation to hold all tables based on the timings to eat.
So he asked his friend to generate a program to find maximum sum possible equal sum of three stacks
Given three stack of the positive numbers, the task is to find the possible equal maximum sum of the stacks with removal of top elements allowed.
Stacks are represented as array, and the first index of the array represent the top element of the stack.
Mandatory declaration is "while(1)"
INPUT:
first line of input must contain size of stack1
second line of input must contain size of stack2
third line of input must contain size of stack3
fourth line of input must contain all elements to be pushed into stack1
fifth line of input must contain all elements to be pushed into stack2
sixth line of input must contain all elements to be pushed into stack3

Source Code

```
#include <iostream>
using namespace std;
int maxSum(int stack1[], int stack2[], int stack3[],int n1, int n2, int n3)
{
 int sum1 = 0, sum2 = 0, sum3 = 0;
 for (int i=0; i < n1; i++)
  sum1 += stack1[i];
 for (int i=0; i < n2; i++)
  sum2 += stack2[i];
 for (int i=0; i < n3; i++)
  sum3 += stack3[i];
 int top1 =0, top2 = 0, top3 = 0;
 int ans = 0;
 while(1)
 {
  if (top1 == n1 || top2 == n2 || top3 == n3)
   return 0;
  if (sum1 == sum2 && sum2 == sum3)
   return sum1;
  if (sum1 >= sum2 && sum1 >= sum3)
   sum1 -= stack1[top1++];
  else if (sum2 >= sum3 && sum2 >= sum3)
   sum2 -= stack2[top2++];
  else if (sum3 >= sum2 && sum3 >= sum1)
   sum3 -= stack3[top3++];
 }
}
int main()
{
  int x,y,z;
  cin >> x >> y >> z;
  int stack1[x],stack2[y],stack3[z];
  for(int i=0;i<x;i++)
    cin >> stack1[i];
  for(int i=0;i<y;i++)
    cin >> stack2[i];
  for(int i=0;i<z;i++)
    cin >> stack3[i];\
  cout << maxSum(stack1,stack2,stack3,x,y,z);
  return 0;
}
```

Sample Input

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

Sample Output

```
5
```

Result

Thus, Program " **ST15** " has been successfully executed

Q. Q5

There once lived a miller with his daughter. When the miller was at work all day turning grain into flour, he loved nothing more than to think up tall tales to amaze people.
One day the King came to town. He heard the miller talking about his daughter. The miller was saying that his daughter was the most amazing girl in their village, if not in all the land.
"You there!" said the King. "What is so amazing about your daughter?". He said , My daughter is very intelligent and graduate from CSE department. You can give any task, my daughter will solve the task very fast and effectively. Then the king decided to tough task, he prepared a question to implement an Input restricted Deque (Double ended Queue) by a program that accepts user's choice of insertion from rear,deletion from both ends and display for the elements in the queue.
Mandatory Declaration is " void Nqueue()"

INPUT:
The first line of input consists of user's choice: 1 for Insertion , 2 for deletion from front , 3 for deletion from rear , 4 for display and 0 for exit.For insertion,the second line of input is the element to be inserted.For deletion and display, next line is the user's choice.

OUTPUT:
For deletion, in case of underflow, the output must mention "Underflow".For display,the queue must be displayed with elements having "->" after them.After each display, the elements must be printed in next line.

Source Code

```
#define MAX 100
#include<stdio.h>
void Nqueue();
int delStart();
int delEnd();
int queue[MAX];
int rear =0, front =0;
void display();
int main()
{
  int choice, c, token;
  scanf("%d",&c);
  while(c!=0)
  {
        switch(c)
        {
         case 1:    Nqueue(token);
                    break;
         case 2:    token=delStart();
                    break;
         case 3:    token=delEnd();
                    break;
         case 4:    display();
                    printf("\n");
                    break;
        }
        scanf("%d",&c);
  }
  return 0;
}
void display()
{
    int i;
    for(i=rear;i<front;i++)
        printf("%d->",queue[i]);
}
void Nqueue()
{
    int token;
    scanf("%d",&token);
    queue[front]=token;
    front=front+1;
}
int delEnd()
{
    int t;
    if(front==rear)
    {
         printf("Underflow\n");
         return 0;
    }
    front=front-1;
    t=queue[front+1];
    return t;
}
int delStart()
{
    int t;
    rear=rear+1;
    t=queue[rear-1];
    return t;
}
```

Sample Input

```
1
6
1
8
1
5
1
9
4
2
4
3
4
1
10
4
0
```

Sample Output

```
6->8->5->9->
8->5->9->
8->5->
8->5->10->
```

Result

Thus, Program " **Q5** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2021-4-29 11:58:47 **Register Number:** RA2031241010094

Q. Q17

Ram and Sham were in a fight. Both of them said they owned the same mango tree. They went up to Birbal and asked him to decide the matter, once and for all. Who was the true owner of the mango tree? . Birbal said, "There is only one way to settle the matter. I will ask a question. who is solving this question then they can own the mango tree. both Ram and Sham need to solve technical question like, given an array of size N consisting of both positive and negative integers, the task is to compute sum of minimum and maximum elements of all sub-array of size K.Using Dequeue data structure and sliding window concept,solve the problem.For example,

Consider an array of {2, 5, -1, 7, -3, -1, -2} and K = 4

Subarrays of size 4 are :

{2, 5, -1, 7}, min + max = -1 + 7 = 6
{5, -1, 7, -3}, min + max = -3 + 7 = 4
{-1, 7, -3, -1}, min + max = -3 + 7 = 4
{7, -3, -1, -2}, min + max = -3 + 7 = 4
Sum of all min & max = 6 + 4 + 4 + 4 = 18

Mandatory declaration is "int SumOfKsubArray(int arr[] , int n , int k)"

INPUT:
The first line of input is the number of elements in the array N and then followed by the array elements.The next line is value of K which is the size of sub-array.

OUTPUT:
The output should be the sum of minimum and maximum elements of all possible sub-arrays of size K.

Source Code

```
#include <iostream>
#include <deque>
using namespace std;
int SumOfKsubArray(int arr[] , int n , int k)
{
    int sum=0;
    deque<int> S(k),G(k);
    int i=0;
    for(i=0;i<k;i++)
    {
        while((!S.empty()) && arr[S.back()] >= arr[i])
            S.pop_back();
        while ( (!G.empty()) && arr[G.back()] <= arr[i])
            G.pop_back();
        G.push_back(i);
        S.push_back(i);
    }
    for (;i<n;i++)
    {
        sum+=arr[S.front()]+arr[G.front()];
        while(!S.empty() && S.front()<=i-k)
            S.pop_front();
        while(!G.empty() && G.front()<=i-k)
            G.pop_front();
        while((!S.empty()) && arr[S.back()]>=arr[i])
            S.pop_back();
        while((!G.empty()) && arr[G.back()]<=arr[i])
            G.pop_back();
        G.push_back(i);
        S.push_back(i);
    }
    sum+=arr[S.front()]+arr[G.front()];
    return sum;
}

int main()
{
    int arr[100],n,k;
   cin>>n;
   for(int i=0;i<n;i++)
       cin>>arr[i];
    cin>>k;
    cout<<SumOfKsubArray(arr,n,k) ;
    return 0;
}
```

Sample Input

```
7
2 5 -1 7 -3 -1 -2
4
```

Sample Output

```
18
```

Result

Thus, Program " **Q17** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2021-4-29 11:59:19 **Register Number:** RA2031241010094

Q. Q21

Given an integer array A. For each index i, find the product of the largest, second largest and the third largest integer in the range [1,i].
Note: Two numbers can be the same value-wise but they should be distinct index-wise.Using Priority Queue, solve this problem.

Mandatory variable name should be" int array123 [100]; "

For example consider an array of 5 elements with {1,2,3,4,5} ,the output will be -1,-1,6,24,60
For the first two indexes, since the number of elements is less than 3, so -1 is printed.
For the third index, the top 3 numbers are 3,2 and 1 whose product is 6.
For the fourth index, the top 3 numbers are 4,3, and 2 whose product is 24.
For the fifth index, the top 3 numbers are 5,4 and 3 whose product is 60.

INPUT:
The first line contains an integer N, denoting the number of elements in the array A.
The next line contains N space separated integers, each denoting the ith integer of the array A.

OUTPUT:
Print the answer for each index in each line. If there is no second largest or third largest number in the array A upto that index, then print "-1", without the quotes.

Source Code

```
#include <stdio.h>

int main()
{
 int array123 [100];
    long int N,i;
    long long int prod;
    scanf("%li",&N);
    long int a[N];
    long int big,bigger,biggest,temp;
    for(i=0;i<N;i++)
    {
     scanf("%li",&a[i]);
    }
    printf("-1\n-1\n");
    big=a[0];
    bigger=a[1];
    if(a[0]>bigger)
    {
     bigger=a[0];
     big=a[1];
    }
    biggest=a[2];
    if(biggest<bigger && biggest>=big)
    {
     temp=bigger;
     bigger=biggest;
     biggest=temp;
    }
    if(biggest<big)
    {
     temp=big;
     big=biggest;
     biggest=bigger;
     bigger=temp;
    }
    prod=big*bigger*biggest;
    printf("%lli\n",prod);
    for(i=3;i<N;i++)
    {
  if(a[i]>biggest)
  {
   big=bigger;
   bigger=biggest;
   biggest=a[i];
  }
  else if(a[i]>bigger)
  {
   big=bigger;
   bigger=a[i];
  }
  else if(a[i]>big)
  {
   big=a[i];
  }
  prod=big*bigger*biggest;
  printf("%lli\n",prod);
    }
    return 0;
}
```

Sample Input

```
8
8 2 4 5 5 6 1 3
```

Sample Output

```
-1
-1
64
160
200
240
240
240
```

Result

Thus, Program " **Q21** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2021-4-29 11:59:50 **Register Number:** RA2031241010094

Q. Q8

There are 'n' people standing in a circle waiting to be executed. The counting out begins at some point in the circle and proceeds around the circle in a fixed direction. In each step, a certain number of people are skipped and the next person is executed. The elimination proceeds around the circle (which is becoming smaller and smaller as the executed people are removed), until only the last person remains, who is given freedom. Given the total number of persons n and a number k which indicates that k-1 persons are skipped and kth person is killed in circle.

Mandatory declaration is "struct node"

For example, if n = 5 and k = 2, then the safe position is 3. Firstly, the person at position 2 is killed, then person at position 4 is killed, then person at position 1 is killed. Finally, the person at position 5 is killed. So the person at position 3 survives.Implement this problem by using a Circular queue.

INPUT:
The first line of input contains the number of people(n) in the circle who are numbered from 1 to n and second line of input is number of passes(k) which incidates that kth person is killed in that circle .
OUTPUT:
The first line of output must contain the order in which the persons get executed from first person to last one to get executed seperated by spaces.The next line of output must be the last person who survives.

Source Code

```
#include<bits/stdc++.h>
using namespace std;


struct Node
{
 int data;
 struct Node *next;
};


Node *newNode(int data)
{
Node *temp = new Node;
temp->next = temp;
temp->data = data;
}


void getJosephusPosition(int m, int n)
{

 Node *head = newNode(1);
 Node *prev = head;
 for (int i = 2; i <= n; i++)
 {
  prev->next = newNode(i);
  prev = prev->next;
 }
 prev->next = head;
 Node *ptr1 = head, *ptr2 = head;
 while (ptr1->next != ptr1)
 {

  int count = 1;
  while (count != m)
  {
   ptr2 = ptr1;
   ptr1 = ptr1->next;
   count++;
  }
  printf("%d ",ptr1->data);
 ptr2->next = ptr1->next;
  ptr1 = ptr2->next;
 }

 printf ("\n%d",ptr1->data);
}

int main()
{
 int n,m;
   cin >> n >> m;
 getJosephusPosition(m, n);
 return 0;
}
```

Sample Input

```
5
2
```

Sample Output

```
2 4 1 5
3
```

Result

Thus, Program " **Q8** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2021-4-29 12:00:24 **Register Number:** RA2031241010094

Q. Q11

John is preparing for GATE entrance exam. he got a solved question from technical website, but he is willing to know the answer in the form of programming. so he called a friend and asked to solve the following task, given an input stream of n characters consisting only of small case alphabets,the task is to find the first non repeating character each time a character is inserted to the stream by using a queue of characters.For example, In a flow of stream : a, a, b, c
a goes to stream : 1st non repeating element : a (a)
a goes to stream : no non repeating element: 0 (a, a)
b goes to stream: 1st non repeating element: b (a, a, b)
c goes to stream : 1st non repeating element: b(a, a, b, c)

Mandatory declaration is while(s--)

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 size of the stream. Then, in the next line are the characters which are inserted to the stream.

OUTPUT:
For each test case in a new line print the first non repeating elements separated by spaces present in the stream at every instinct when a character is added to the stream, if no such element is present print 0.

Source Code

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

char g(queue<char> &q, int v[]) {
    while(!q.empty()) {
        if (v[q.front()-'a'] > 1) {
            q.pop();
        } else {
            return q.front();
        }
    }
    return 'Z';
}

int main() {
    int s;
    cin >> s;
    while(s--) {
        int n;
        cin >> n;
        char s[n];
        for (int i=0; i < n; i++) cin >> s[i];
        queue<char> q;
        int f[26] = {0};
        for (int i=0; i < n; i++) {
            f[s[i]-'a'] ++;
            if (f[s[i]-'a'] < 2) {
                q.push(s[i]);
            }
            char ch = g(q,f);
            if (ch == 'Z') cout << "0 ";
            else cout << ch << " ";
        }
        cout << '\n';
    }
    return 0;
}
```

Sample Input

```
2
4
a a b c
3
a a c
```

Sample Output

```
a 0 b b
a 0 c
```

Result

Thus, Program " **Q11** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2021-4-29 12:00:55 **Register Number:** RA2031241010094

Q. Q9

A English department HOD is telling a story to students. In all the land, no one was better with a bow and arrow than Robin Hood. He lived with his band of Merry Men in Sherwood Forest, suddenly a student wake up and ask different question. Professor got angry on that student. immediately called the CSE professor to conduct a surprise test to all students. The CSE professor distributed a question like, given a number n, implement a function that generates and prints all binary numbers with decimal values from 1 to n.Using a queue of strings by appending 0 and 1 accordingly in the queue, generate the binary numbers from 1 to n.

Mandatory declaration is "void gen(int n)"

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.
OUTPUT:
Print all binary numbers from 1 to n which are seperated by spaces and each test case output is printed in next line.

Source Code

```
#include <bits/stdc++.h>
using namespace std;
void gen(int n)
{
 cout << "\n";
}
int f(int num)
{
 long  decimal_num, remainder, base = 1, binary = 0, no_of_1s = 0;
    decimal_num = num;
    while (num > 0)
    {
        remainder = num % 2;
        if (remainder == 1)
        {
            no_of_1s++;
        }
        binary = binary + remainder * base;
        num = num / 2;
        base = base * 10;
    }
    return binary;
}
int main()
{
  int x;
  cin>>x;
  while(x--)
  {
   int gg;
   cin>>gg;
   for(int i=1;i<=gg;i++)
            cout<<f(i)<<" ";
         cout<<endl;
  }
}
```

Sample Input

```
2
2
5
```

Sample Output

```
1 10
1 10 11 100 101
```

Result

Thus, Program " **Q9** " has been successfully executed

Q. Q12

Ramesh and his wife Lata are facing a cash crisis.They go to the nearby ATM to get some cash.There are 3 ATMs inside the same room. People are standing in queue outside, and go inside the room in groups of 3 to the ATMs, fetch their money and come out.Lata has an irrational fear in getting money from ATM that her ATM pin will somehow be stolen and all her money will be lost.So, she will always like to go into the room with Ramesh. Ramesh is standing at position K in line, immediately followed by Lata (i.e. at position K + 1).Can you tell whether Ramesh and Lata both will be able to get money in such a way that Lata does not feel insecure?Using queue, find whether they can get money for the given set of N and K.

Mandatory declaration is "int FIND(int ,int);"

INPUT:
The first line contains an integer T denoting the number of test cases. T test cases follow.
The only line of each test case contains two space separated integers N and K, where N denotes number of people in the queue, and K denotes the position of Ramesh.

OUTPUT:
For each test case, output "yes" or "no" correspondingly denoting whether they both will be able to get the money without Lata getting scared.

CONSTRAINTS:
1 â¤ T â¤ 100
3 â¤ N â¤ 100
1 â¤ K â¤ N-1
N is divisible by 3.

Source Code

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

int FIND(int ,int );

int main() {

  int t,n,k;
  cin >> t;
  while(t--)
  {
    cin >> n >> k;
    if(k > n-3)
    {
      cout << "yes\n";
    }
    else
    {
      cout << "no\n";
    }
  }

  return 0;
}
```

Sample Input

```
2
3 1
6 3
```

Sample Output

```
yes
no
```

Result

Thus, Program " **Q12** " has been successfully executed

Q. Q15

Long ago in India there was an old deserted village. Empty were the old houses, streets and shops. The windows were open, the stairs broken. Making it one very fine place for mice to run around. you can be sure of that!. now people of the village decided to give a high qualified education to youngsters for village development . So they made a coaching center for programming language. now coaching center people conducting a test. one of the question was, given a Queue data structure that supports standard operations like enqueue() and dequeue(). We need to implement a Stack data structure using only instances of Queue and queue operations allowed on the instances.Implement the stack using 2 queues by a program that accepts user's choice of Insertion,deletion and display for the elements in the stack."

Mandatory declaration is "int *array;"

INPUT:
The first line of input consists of user's choice: 1 for Push, 2 for Pop,3 for display and 0 for exit.For insertion, the second line of input is the element to be inserted.For deletion and display, next line is the user's choice.
OUTPUT:
For display choice, the stack must be displayed with elements having "->" between them.After each display, the elements must be printed in next line.

Source Code

```
 #include <iostream>
using namespace std;
int *array;
void enq_front();
int insertfront(int [],int );
void remove(int []);
void display(int[],int,int);
const int size=50;
int queue[size],front=-1,rear=-1;
int main() {
 int c,x,res;
   do
   {
        cin>>c;
   switch(c)
    {
       case 1: {
         cin>>x;
          res=insertfront(queue,x);
          break;
        }
       case 2: {
         remove(queue);
          break;
        }
       case 3: {
         display(queue,front,rear);
          break;
        }
       case 0:
          exit(0);
          break;
       default:
          cout<<"Invalid Input";
    }
   }while(c!=0);
 return 0;
}
int insertfront(int queue[],int ele)
{
   if(rear==-1)
    {
      front=rear=0;
      queue[front]=ele;
    }
 for(int i=rear-1;i>=front;i--)
    {
     queue[i+1]=queue[i];
    }
   rear++;
   queue[front]=ele;
}

void remove(int queue[])
{
   if(front==rear)
       cout<<"Underflow";
  else
    {
       if(front==rear)
          front=rear=-1;
       else
          front++;
    }
}

void display(int queue[],int front,int rear)
{
 /*if(front==rear)
       cout<<"Underflow";*/
   for(int i=rear-1;i>=front;i--)
    {
     cout<<queue[i]<<"->";
    }
  cout<<endl;
   //cout<<queue[rear]<<endl;
}
void enq_front()
{ int ele;
 cin>>ele;
   if(rear==-1)
    {
      front=rear=0;
      queue[front]=ele;
    }
 for(int i=rear-1;i>=front;i--)
    {
     queue[i+1]=queue[i];
    }
   rear++;
   queue[front]=ele;
}
```

Sample Input

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

Sample Output

```
3->4->5->7->9->
3->4->5->7->
```

Result

Thus, Program " **Q15** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2021-4-29 12:03:03 **Register Number:** RA2031241010094

Q. Q1

Once upon a time there was a very rich man who lived with his three daughters. The two older daughters laughed at anyone who did not dress as well as they did (one of the person is hers class mate). If the two of them were not resting at home, they were out shopping for as many fine dresses and hats as they could carry home. Next day they went to school mid term test. Question Type 'B' contains a technical data structure question like, Perform a Queue using Arrays by a program that accepts user's choice of insertion,deletion and display for the elements in the queue.

Mandatory declaration are "void deq();", "void enq();"

INPUT:
The first line of input consists of user's choice: 1 for Insertion, 2 for deletion,3 for display and 0 for exit.For insertion,the second line of input is the element to be inserted.For deletion and display, next line is the users choice. The maximum number of elements in the queue is 5.
OUTPUT:
For deletion, in case of underflow,the output must mention "Underflow".For display, the queue must be displayed with elements having "->" between them.After each display, the elements must be printed in next line.

Source Code

```
#include<iostream>
using namespace std;
int a,queue[5],front=-1,rear=-1,num;
void enq();
void deq();
int main()
{
  cin>>a;
  while(a!=0)
  {
    switch(a)
    {
      case 1:
        cin>>num;
        if(front==-1 && rear==-1)
        {
          front=0;
          rear=0;
        }
        else
          rear++;
        queue[rear]=num;
        break;
      case 2:
        if(front==-1 || front>rear)
          cout<<"Underflow";
        else
          front++;
        break;
      case 3:
        for(int i=front;i<=rear;i++)
        {
          cout<<queue[i]<<"->";
        }
        cout<<endl;
        break;
    }
    if(front>rear)
    {
      cout<<"Underflow";
      exit(0);
    }
    cin>>a;
  }
  return 0;
}
```

Sample Input

```
1
5
1
4
1
1
1
2
1
8
3
2
2
3
0
```

Sample Output

```
5->4->1->2->8->
1->2->8->
```

Result

Thus, Program " **Q1** " has been successfully executed