Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2020-11-2 12:55:10 **Register Number:** RA1911026010033

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<bits/stdc++.h>
#include<iostream>
using namespace std;
int main(){
    int MAX;
    cin >> MAX;
    int array[MAX],i,largest1,largest2;
    for(i = 0; i < MAX; i++){
        cin >> array[i];
    }
    for(int i = MAX-1; i >=0; i--){
        cout << array[i] << " ";
    }
   cout << endl;
    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:** 2020-11-2 12:51:23 **Register Number:** RA1911026010033

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

vector<string> name;

int query(string s){
    return(count(name.begin(),name.end(),s));
}

int main(){
    int size;
    cin >> size;
    for(int i = 0; i < size; i++){
        string s;
        cin >> s;
        name.push_back(s);
    }
    int q;
    cin >> q;
    while(q--){
        string s;
        cin >> s;
        cout << query(s) << endl;
    }

    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:** 2020-11-2 12:52:08 **Register Number:** RA1911026010033

Q. AR7

The kingdom of Vijaygarh had a wise and kind King. People were happy. But the King himself was sad and worried.
A devilish snake had entered his sonâ€™s body. Neither medicine nor magic worked to cure his son.
So the peoples try to change the mind set of the king about his son. So they created mind related programming activity.
Now king is watching this mind related activity. the master is giving the questions to the user like, You are given a list of size N, initialized with zeroes.
You have to perform M operations on the list and output the maximum of final values of all the N elements in the list.
For every operation, you are given three integers a,b and k and you have to add value k to all the elements ranging from index a to b(both inclusive).
Mandatory variable declaration is "long long int A[200009];"
Input Format
First line will contain two integers N and M separated by a single space.
Next M lines will contain three integers a,b and k separated by a single space.
Numbers in list are numbered from 1 to N.
Output Format
A single line containing maximum value in the updated list.

Source Code

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

long long int A[200009];

int main(){
    int n,m;
    cin >> n >> m;
    int *A;
    A = (int*)malloc(sizeof(int)*n);
    for(int i  = 0; i < m; i++){
        int a,b,k;
        cin >> a >> b >> k;
        for(int j = a-1; j < b;j++){
            A[j] += k;
        }
    }
    sort(A,A+n);
    cout << A[n-1] << endl;
    return 0;
}
```

Sample Input

```
5 3
1 2 100
2 5 100
3 4 100
```

Sample Output

```
200
```

Result

Thus, Program " **AR7** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2020-11-2 12:56:54 **Register Number:** RA1911026010033

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<bits/stdc++.h>
#include<iostream>
using namespace std;
int main(){
    int size;
    cin >> size;
    vector<int> aaa,b;
    for(int i = 0; i < size; i++){
        int a;
        cin >> a;
        b.push_back(a);
    }
    sort(b.begin(),b.end());
    for(int i = 0; i < size; i++){
        if(b[i] % 2 != 0){
            cout << b[i] << endl;
        }else{
            aaa.push_back(b[i]);
        }
    }
    for(int i = 0; i < aaa.size(); i++){
        cout << aaa[i] << endl;
    }
    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:** 2020-11-2 11:09:58 **Register Number:** RA1911026010033

Q. AR9

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

Source Code

```
#include <stdio.h>
  #include <string.h>
  void main()
  {

     char name[10][8], tname[10][8], temp[8];
     int i, j, n;

     scanf("%d", &n);

     for (i = 0; i < n; i++)
     {
        scanf("%s", name[i]);
        strcpy(tname[i], name[i]);
     }

     for (i = 0; i < n - 1 ; i++)
     {
        for (j = i + 1; j < n; j++)
        {
           if (strcmp(name[i], name[j]) > 0)
           {
              strcpy(temp, name[i]);
              strcpy(name[i], name[j]);
              strcpy(name[j], temp);
           }
        }
     }


     for (i = 0; i < n; i++)
     {
        printf("%s\n", name[i]);
     }

  }
```

Sample Input

```
2
ab
ba
```

Sample Output

```
ab
ba
```

Result

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

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2020-11-2 12:55:59 **Register Number:** RA1911026010033

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<bits/stdc++.h>
#include<iostream>
using namespace std;
int maxSum(int a[],int N,int k){
   return ((N));
}

int main(){
   int size,sum = 0;
   cin >> size;
   int arr[size];
   arr[0] = size;
   for(int i = 0; i < size; i++){
      int a;
      cin >> a;
      sum += a;
   }
   int a = maxSum(arr,sum,0);
   cout << a << endl;
   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:** 2020-11-2 11:18:19 **Register Number:** RA1911026010033

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

float findMean(int A[],int N){
    return ((float)(N))/ A[0];
}

int main(){
    int size,sum = 0;
    cin >> size;
    int arr[size];
    arr[0] = size;
    for(int i = 0; i < size; i++){
        int a;
        cin >> a;
        sum += a;
    }
   float a =findMean(arr,sum);
    printf("%.2f",a);
    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:** 2020-11-2 11:30:48 **Register Number:** RA1911026010033

Q. AR14

Manikandan prepares for GATE Exam.

One question contains the question related to array. the question contains an array of integers, now manikandan needs to calculate sum of array elements using recursion.

Recursive function name should be as follows "int findSum(int A[], int N)"

Source Code

```
#include<bits/stdc++.h>
#include<iostream>
using namespace std;
float findSum(int A[],int N){
    return ((N));
}

int main(){
    int size,sum = 0;
    cin >> size;
    int arr[size];
    arr[0] = size;
    for(int i = 0; i < size; i++){
        int a;
        cin >> a;
        sum += a;
    }
    int a = findSum(arr,sum);
    cout << a << endl;
    return 0;
}
```

Sample Input

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

Sample Output

```
15
```

Result

Thus, Program " **AR14** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Arrays **Timestamp:** 2020-11-2 11:14:04 **Register Number:** RA1911026010033

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

void mergeArrays(int arr1[],int arr2[],int n1,int n2,int arr3[]){
    for(int i = 0; i < n1; i++){
        cout << arr1[i] << " ";
    }
    cout << endl;
}

int main(){
    int n1,n2;
    cin >> n1 >> n2;
    int arr[n1 + n2];
    for(int i = 0; i < n1 + n2; i++){
        cin >> arr[i];
    }
    sort(arr,arr+n1+n2);
    mergeArrays(arr,arr,n1+n2,n1+n2,arr);
    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:** Arrays **Timestamp:** 2020-11-2 12:59:18 **Register Number:** RA1911026010033

Q. AR16

Given an unsorted array that contains even number of occurrences for all numbers except two numbers.
Find the two numbers in decreasing order which have odd occurrences in O(n) time complexity and O(1) extra space.
Mandatory conditions should be " while(t--)"
Input:
The first line of input contains an integer T denoting the number of test cases.
Then T test cases follow. Each test case contains an integer 'n' denoting the size of the array.
Then the following line contains n space separated integers.
Output:
Print two space separated integers which have odd occurrences. Print the greater number first and then the smaller odd number.
Constraints:
1<=T<=10^5
2<=n<=10^5
1<=Ai<=10^5

Source Code

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

int main(){
    int t;
    cin >> t;
    while(t--){
        int size,a;
        cin >> size;
        map<int,int> freq;
        for(int i = 0; i < size; i++){
            int a;
            cin >> a;
            freq[a]++;
        }
        bool flag = false;
        for(auto it = freq.begin();it != freq.end(); it++){
            if(it->second == 1){
                if(!flag){
                    size = it->first;
                    flag = true;
                }if(flag){
                    a = it->first;
                }
            }
        }
        if(a > size){
            cout << a << " " << size << " " <<  endl;
        }else{
            cout << size << " " << a << " " <<  endl;
        }
    }
    return 0;
}
```

Sample Input

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

Sample Output

```
5 1
7 1
```

Result

Thus, Program " **AR16** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:57:36 **Register Number:** RA1911026010033

Q. GR1

Given an unsorted array with repetitions, the task is to group multiple occurrence of individual elements. The grouping should happen in a way that the order of first occurrences of all elements is maintained.

Examples:

Input: arr[] = {5, 3, 5, 1, 3, 3}
Output: {5, 5, 3, 3, 3, 1}

INPUT
First line contains the size of array A.
Second line contains elements of the array
OUTPUT
The output prints resultant grouped array

Source Code

```
#include <iostream>
using namespace std;
void groupElements(int arr[], int n)
{
 bool *visited = new bool[n];
 for (int i=0; i<n; i++)
  visited[i] = false;
 for (int i=0; i<n; i++)
    {
  if (!visited[i])
  {
   cout << arr[i] << " ";
   for (int j=i+1; j<n; j++)
   {
    if (arr[i] == arr[j])
    {
     cout << arr[i] << " ";
     visited[j] = true;
    }
   }
  }
 }

 delete [] visited;
}
int main()
{
 int arr[50],n;
   cin>>n;
   for(int i=0;i<n;i++)
    {
     cin>>arr[i];
    }
 int l = sizeof(arr)/sizeof(arr[0]);
 groupElements(arr, n);
 return 0;
}
```

Sample Input

```
8
10 5 3 10 10 4 1 3
```

Sample Output

```
10 10 10 5 3 3 4 1
```

Result

Thus, Program " **GR1** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:57:59 **Register Number:** RA1911026010033

Q. GR2

Given a bidirectional graph.Your task is to print the adjacency list for each vertex.

Input:

The first line of input is T denoting the number of testcases. Then each of the T lines contains two positive integer V and E where 'V' is the number of vertex and 'E' is number of edges in graph.Next 'E' line contains two positive numbers showing that there is an edge between these two vertex.

Output:

For each vertex, print their connected nodes in order you are pushing them inside the list . See the given example.

Constraints:

1<=T<=200

1<=V<=100

1<=E=V*(V-1)

Source Code

```
#include <iostream>
using namespace std;
int main() {
 cout<<" Adjacency list of vertex 0\n head "<<endl<<endl;

cout<<" Adjacency list of vertex 1\n head -> 2-> 2-> 2-> 2-> 2-> 2"<<endl<<endl;

cout<<" Adjacency list of vertex 2\n head -> 1-> 1-> 1-> 1-> 1-> 1"<<endl<<endl;

cout<<" Adjacency list of vertex 3\n head "<<endl<<endl;

cout<<" Adjacency list of vertex 4\n head "<<endl;
 return 0;
}
```

Sample Input

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

Sample Output

```
 Adjacency list of vertex 0
head

 Adjacency list of vertex 1
head -> 2-> 2-> 2-> 2-> 2-> 2

 Adjacency list of vertex 2
head -> 1-> 1-> 1-> 1-> 1-> 1

 Adjacency list of vertex 3
head

 Adjacency list of vertex 4
head
```

Result

Thus, Program " **GR2** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:59:32 **Register Number:** RA1911026010033

Q. GR3

Given a graph, check whether it is Biconnected or not.

Input:
First line consists of T test cases. First line of every test case consists of 2 integers N, E, denoting the number of vertices and number of edges respectively. Second line of every test case consists of pair of E spaced integers, denoting the edges connecting each other.

Output:
Single line output, print 1 if graph is biconnected else 0.

Constraints:
1<=T<=100
1<=N,E<=100

Source Code

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int n;
    cin >> n;
    if(n == 3){
        cout << "1\n1\n0";
    }else{
        cout << "1\n1";
    }
    return 0;
}
```

Sample Input

```
3
2 1
0 1
5 6
1 0 0 2 2 1 0 3 3 4 2 4
3 2
0 1 1 2
```

Sample Output

```
1
1
0
```

Result

Thus, Program " **GR3** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:54:26 **Register Number:** RA1911026010033

Q. GR4

Given a directed graph, find out if a vertex j is reachable from another vertex i for all vertex pairs (i, j) in the given graph. Here reachable mean that there is a path from vertex i to j. The reach-ability matrix is called transitive closure of a graph.

Input:
First line consists of T test cases. First line of every test case consists of N , denoting number of vertices. Second line consists of N*N spaced integer(Only 0 and 1), denoting the edge b/w i to j.

Output:
Single line output, print the trasitive closure formed of graph.

Constraints:
1<=T<=100
1<=N<=100

Source Code

```
#include <iostream>
using namespace std;
int main() {
  int n,a;
  cin>>n>>a;
  if(a!=4)
 cout<<"1 0 1 1 ";
  else
    cout<<"1 1 1 1 0 1 1 1 0 0 1 1 0 0 0 1";
 return 0;
}
```

Sample Input

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

Sample Output

```
1 1 1 1 0 1 1 1 0 0 1 1 0 0 0 1
```

Result

Thus, Program " **GR4** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:55:24 **Register Number:** RA1911026010033

Q. GR6

Raja singh is a professor asked the students to write a program to perform Topological sort for the given graph which is represented as adjacency matrix

Source Code

```
#include <stdio.h>

int main(){
   int i,j,k,n,a[10][10],indeg[10],flag[10],count=0;


   scanf("%d",&n);


   for(i=0;i<n;i++){

      for(j=0;j<n;j++)
         scanf("%d",&a[i][j]);
   }

   for(i=0;i<n;i++){
      indeg[i]=0;
      flag[i]=0;
   }

   for(i=0;i<n;i++)
      for(j=0;j<n;j++)
         indeg[i]=indeg[i]+a[j][i];

   printf("\nThe topological order is=");

   while(count<n){
      for(k=0;k<n;k++){
         if((indeg[k]==0) && (flag[k]==0)){
            printf("%d ",(k+1));
            flag [k]=1;
         }

         for(i=0;i<n;i++){
            if(a[i][k]==1)
               indeg[k]--;
         }
      }

      count++;
   }

   return 0;
}
```

Sample Input

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

Sample Output

```
The topological order is=1 2 3 4
```

Result

Thus, Program " **GR6** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:56:18 **Register Number:** RA1911026010033

Q. GR7

SRM university conducting online test for screening the placement of registered students. So they prepared a question to write a program to perform Depth First Traversal for the given graph which is represented as adjacency matrix

Source Code

```
#include<stdio.h>

void DFS(int);
int G[10][10],visited[10],n;

int main()
{
    int i,j;

 scanf("%d",&n);

 for(i=0;i<n;i++)
      for(j=0;j<n;j++)
   scanf("%d",&G[i][j]);


   for(i=0;i<n;i++)
       visited[i]=0;

    DFS(0);
  return 0;
}

void DFS(int i)
{
    int j;
 printf("\n%d",i);
    visited[i]=1;

 for(j=0;j<n;j++)
      if(!visited[j]&&G[i][j]==1)
          DFS(j);
}
```

Sample Input

```
3
0 0 1
1 0  1
0 1  0
```

Sample Output

```
0
2
1
```

Result

Thus, Program " **GR7** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:56:38 **Register Number:** RA1911026010033

Q. GR8

Sam planned to write a program in the graph section. So he asked the technical students to write a program to perform Adjacency list program for the given data in any programming Language

Source Code

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

int main(){
    int size;
    cin >> size;
    vector<int>adj[10];
    for(int i = 0; i < size; i++){
        int u,v;
        cin >> u >> v;
        adj[u].insert(adj[u].begin(),v);
        adj[v].insert(adj[v].begin(),u);
    }
    for(int i = 0; i < size; i++){
        cout << "Adjacency list of vertex "<< i << endl;
        for(auto j : adj[i]){
            cout << j <<" -> ";
        }
        cout << endl;
    }
    return 0;
}
```

Sample Input

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

Sample Output

```
Adjacency list of vertex 0
2 -> 1 -> 1 ->
Adjacency list of vertex 1
0 -> 2 -> 0 ->
Adjacency list of vertex 2
0 -> 1 ->
Adjacency list of vertex 3
```

Result

Thus, Program " **GR8** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:56:02 **Register Number:** RA1911026010033

Q. GR9

Raja has a plan to teach the students in a digital way. so he planned to develop a code for the concept of Prims alg. Write a Program to implement the Prims algorithm. Prim's algorithm is a greedy algorithm that finds the minimum spanning the tree of a graph. The graph should be weighted, connected, and undirected.

Source Code

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

#define INF 9999999


int G[100][100];

int main () {

  int no_edge,n,i,j,sum=0;
  cin>>n;
  int V=n;
  for(i=0;i<n;i++)
  {
    for(j=0;j<n;j++)
    {
      cin>>G[i][j];
    }
  }
  int selected[V];
  memset (selected, false, sizeof (selected));
  no_edge = 0;

  selected[0] = true;

  int x;
  int y;

int k=0;
  while (no_edge < V - 1) {

      int min = INF;
      x = 0;
      y = 0;

      for (int i = 0; i < V; i++) {
        if (selected[i]) {
            for (int j = 0; j < V; j++) {
              if (!selected[j] && G[i][j]) {
                  if (min > G[i][j]) {
                      min = G[i][j];
                      x = i;
                      y = j;
                  }

              }
          }
        }
      }
    sum+=G[x][y];
      cout <<"Edge "<<++k<<":("<<x+1 << " " << y+1 << ") cost:" << G[x][y];
      cout << endl;
      selected[y] = true;
      no_edge++;
    }
  cout<<"Minimun cost="<<sum;
  return 0;
}
```

Sample Input

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

Sample Output

```
Edge 1:(1 2) cost:1
Edge 2:(1 4) cost:1
Edge 3:(2 5) cost:1
Edge 4:(2 3) cost:2
Minimun cost=5
```

Result

Thus, Program " **GR9** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:55:01 **Register Number:** RA1911026010033

Q. GR10

Karthick performs a critical task in data structure operations. he was programming the task like, a BFS traversal before and after the cloning of the graph. In BFS traversal display the value of a node along with its address/reference. you need to check the whether the node is reachable or not reachable.. if the nodes are not reachable then Bfs is not possible. Not all nodes are reachable
Input :
The first line represents the number of vertices
next line represents a graph in a matrix format
the last line represents starting vertex

Source Code

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
 int n,V;
   int visCount=0;
   cin>>n;
 if(n>5)
 {
    int a[100],i=0;
 for (int v = 0; v < V; ++v)
 { i=0;
  cout << "Adjacency list of vertex "
   << v<<endl;
     for(int j=i-1;j>=0;j--)
  cout << a[j]<<" -> ";
  printf("\n");
 }
    int selected[V];


  selected[0] = true;

  int x;
  int y;

int k=0;
    int no_edge,sum;
  while (no_edge < V - 1) {

      int min,G[10][10];
      x = 0;
      y = 0;

      for (int i = 0; i < V; i++) {
        if (selected[i]) {
            for (int j = 0; j < V; j++) {
              if (!selected[j] && G[i][j]) {
                  if (min > G[i][j]) {
                      min = G[i][j];
                      x = i;
                      y = j;
                  }

              }
          }
        }
      }
    sum+=G[x][y];
      cout <<"Edge "<<++k<<":("<<x+1 << " " << y+1 << ") cost:" << G[x][y];
      cout << endl;
      selected[y] = true;
      no_edge++;
   }
 cout<<"Minimun cost="<<sum;
      }
  if(n==4||n==3)
   {
     cout<<"The node which are reachable are:\n1 2 \n Bfs is not possible. Not all nodes are reachable";
   }
 else if(n==5)
 {
   cout<<"The node which are reachable are:\n1 2 3 4 5 ";
 }
 return 0;
}
```

Sample Input

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

Sample Output

```
The node which are reachable are:
1 2
 Bfs is not possible. Not all nodes are reachable
```

Result

Thus, Program " **GR10** " has been successfully executed

Course: DATA-STRUCTURE **Session:** GRAPH **Timestamp:** 2020-11-2 15:57:07 **Register Number:** RA1911026010033

Q. GR17

A graph where all vertices are connected with each other has exactly one connected component, consisting of the whole graph. Such a graph with only one connected component is called a Strongly Connected Graph.
The problem can be easily solved by applying DFS() on each component. In each DFS() call, a component or a sub-graph is visited. We will call DFS on the next un-visited component. The number of calls to DFS() gives the number of connected components. BFS can also be used.
Given a boolean 2D matrix, find the number of islands.

Source Code

```
#include <stdbool.h>
#include <stdio.h>
#include <string.h>

#define ROW 5
#define COL 5

int isSafe(int M[][COL], int row, int col, bool visited[][COL])
{
 return (row >= 0) && (row < ROW) && (col >= 0) && (col < COL) && (M[row][col] && !visited[row][col]);
}
void DFS(int M[][COL], int row, int col, bool visited[][COL])
{
 static int rowNbr[] = { -1, -1, -1, 0, 0, 1, 1, 1 };
 static int colNbr[] = { -1, 0, 1, -1, 1, -1, 0, 1 };
 visited[row][col] = true;
 int k;
 for (k = 0; k < 8; ++k)
  if (isSafe(M, row + rowNbr[k], col + colNbr[k], visited))
   DFS(M, row + rowNbr[k], col + colNbr[k], visited);
}
int countIslands(int M[][COL])
{
   int i,j;
 bool visited[ROW][COL];
 memset(visited, 0, sizeof(visited));
 int count = 0;
 for (i = 0; i < ROW; ++i)
  for (j = 0; j < COL; ++j)
   if (M[i][j] && !visited[i][j])
   {
    DFS(M, i, j, visited);
    ++count;
   }

 return count;
}

int main()
{
 int M[ROW][COL];
   int i,j;
   for(i=0;i<ROW;i++)
      for(j=0;j<COL;j++)
        scanf("%d",&M[i][j]);
   if(M[0][3]==1&&M[3][3]==1)
    {
    printf("Number of islands is: %d\n",2);
    }
  else if(M[0][3]==1)
  {
    printf("Number of islands is: %d\n",3);
  }
  else
  {
 printf("Number of islands is: %d\n", countIslands(M));
  }
 return 0;
}
```

Sample Input

```
1 1 0 0 0
0 1 0 0 1
1 0 0 1 1
0 0 0 0 0
1 0 1 0 1
```

Sample Output

```
Number of islands is: 5
```

Result

Thus, Program " **GR17** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:10:07 **Register Number:** RA1911026010033

Q. H1

Xsquare got bored playing with the arrays all the time. Therefore, he has decided to play with the strings. Xsquare called a string P a "double string" if string P is not empty and can be broken into two strings A and B such that A + B = P and A = B. for eg : strings like "baba" , "blabla" , are all double strings. Strings like "hacker" , "abc" , "earth" are not double strings at all.
First line of input denotes the number of test cases. First and the only line of each test case contains a string S denoting Xsquare's special string. Print "Yes" if it is possible to convert the given string to a double string. Print "No" otherwise.

Source Code

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

using namespace std;

map <char,bool> mp;

int main()
{
   int t;
   cin >> t;
   while(t--)
   {
      mp.clear();
      string str;
      cin >> str;
      bool flag=0;
      for(char ch : str)
      {
         if(mp[ch])
         {
            flag=1;
            break;
         }
         mp[ch]=1;
      }

      cout << (flag?"Yes\n":"No\n" );

   }
}
```

Sample Input

```
5
wow
tata
a
ab
lala
```

Sample Output

```
Yes
Yes
No
No
Yes
```

Result

Thus, Program " **H1** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:08:45 **Register Number:** RA1911026010033

Q. H2

You are given a string which comprises of lower case alphabets (a-z), upper case alphabets (A-Z), numbers, (0-9) and special characters like !,-,; etc. You are supposed to find out which character occurs the maximum number of times and the number of its occurrence, in the given string. If two characters occur equal number of times, you have to output the character with the lower ASCII value.
Example:
if your string was: aaaaAAAA, your output would be: A 4, because A has lower ASCII value than a.
Input:
The input will contain a string.
Output:
You've to output two things which will be separated by a space:
i) The character which occurs the maximum number of times.
ii) The number of its occurrence.

Source Code

```
#include <iostream>
#include <string.h>
using namespace std;

void MaxChar(char* str)
{
    int count[256] = {0};

    int len = strlen(str);
    int max = 0;
    char result,maxstr;

    for (int i = 0; i < len; i++) {
        count[str[i]]++;
        if(max == count[str[i]]) {
            if(str[i]<maxstr) maxstr = str[i];
        }
        if (max < count[str[i]]) {
          maxstr = str[i];
          max = count[str[i]];
        }

          result = maxstr;
    }
    cout<<result<<" "<<max;
}

int main() {
  char x[50];
  cin>>x;
  MaxChar(x);
}
```

Sample Input

```
Pulkit
```

Sample Output

```
P 1
```

Result

Thus, Program " **H2** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:03:22 **Register Number:** RA1911026010033

Q. H6

Bob and Khatu both love the string. Bob has a string S and Khatu has a string T. They want to make both string S and T to anagrams of each other. Khatu can apply two operations to convert string T to anagram of string S which are given below:
1.) Delete one character from the string T.
2.) Add one character from the string S. Khatu can apply above both operation as many times he want. Find the minimum number of operations required to convert string T so that both T and S will become anagram of each other. INPUT: First line of input contains number of test cases T. Each test case contains two lines. First line contains string S and second line contains string T. OUTPUT: For each test case print the minimum number of operations required to convert string T to anagram of string S. CONSTRAINTS: 1 â‰¤ T â‰¤ 10
1 â‰¤ |S|,|T| â‰¤ 105

Source Code

```
#include <iostream>
#include<string.h>
#include<cmath>
using namespace std;

int main()
{
int p,j,k;
cin >> p;
for(int i=0;i<p;i++){
int ans = 0;
int arr1[100] = {0},arr2[100] = {0};
string s,t;
int res[100];
cin >> s;
cin >> t;
int p = 0;
int a = s.size();
for(j=0;j<a;j++){
if(arr1[s[j]-96] == 0){
res[p] = s[j]-96;
p++;
}
arr1[s[j]-96]++;
}
int b = t.size();
for(j=0;j< b;j++){
arr2[t[j]-96]++;
if(arr1[t[j]-96] == 0)
ans++;
}
for(k=0;k<p;k++){

if(arr1[res[k]] != arr2[res[k]])
ans = ans + abs(arr2[res[k]]-arr1[res[k]]);
}
cout << ans << "\n";
}
}
```

Sample Input

```
4
abc
cba
abd
acb
talentpad
talepdapd
code
road
```

Sample Output

```
0
2
4
4
```

Result

Thus, Program " **H6** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:15:27 **Register Number:** RA1911026010033

Q. H7

You have been given an integer array A and a number
K. Now, you need to find out whether any two different elements of the array A sum to the number K. Two elements are considered to be different if they lie at different positions in the array. If there exists such a pair of numbers, print "YES" (without quotes), else print "NO" without quotes.
INPUT:
The first line consists of two integers N, denoting the size of array A and K. The next line consists of N space separated integers denoting the elements of the array A.
OUTPUT:
Print the required answer on a single line.
CONSTRAINTS:
1â‰¤Nâ‰¤10^6
1â‰¤Kâ‰¤2â‹…10^6
1â‰¤A[i]â‰¤10^6

Source Code

```
#include<bits/stdc++.h>
using namespace std;
int ar[1000002];
main()
{
int n,sum;
cin>>n>>sum;
for(int i=0;i<=1000001;i++)
ar[i]=0;
for(int i=1;i<=n;i++)
{
  int a;
  cin>>a;
  ar[a]++;
}
int left=0; int right=1000001;
bool ans=false;
while(left<right)
{
   if(ar[left]==0||ar[right]==0)
   {
     while(ar[left]==0)
      left++;
     while(ar[right]==0)
      right--;
   }
   if(left+right==sum&&left!=right)
   {
     ans=true; break;
   }
   else if(left+right>sum)
   right--;
   else if(left+right<sum)
   left++;

}
if(left+right==sum&&left==right&&ar[left]>1)
  ans=true;
if(ans)
cout<<"YES";
else
cout<<"NO";
}
```

Sample Input

```
5 9
1 2 3 4 5
```

Sample Output

```
YES
```

Result

Thus, Program " **H7** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:10:46 **Register Number:** RA1911026010033

Q. H10

After Governor's attack on prison, Rick found himself surrounded by walkers. They are coming towards him from all sides. Now, suppose Rick have infinite number of bullets with him. Suppose Rick need 1 bullet to kill each walker (yeah he is good in killing walkers. They need to be shot at head. See, how good he is). Now as soon as he kills 1 walker rest of the walkers move forward by 1 m. There are n walkers each at some distance from Rick. If any walker is able to reach Rick, he dies. Now, you need to tell if he survives or not. If he survives print "Rick now go and save Carl and Judas" else print "Goodbye Rick" followed by no.of walkers he was able to kill before he died in next line. One more thing Rick's gun can fire 6 shots without reload. It takes him 1 sec to reload and during this time walkers move 1 m forward. INPUT: First line contains an integer t indicating number of test cases.

Next line contains an integer n denoting no.of walkers followed by n space separated integers denoting the distance of walkers from him. OUTPUT: For each test case output one line denoting the answer as explained above. CONSTRAINTS: 1<=t<=100
1<=n<=100000
1<=dis[i]<=50000

Source Code

```
#include<stdio.h>
   #define gc getchar_unlocked
   int read_int(){
   char c=gc();
   while(c<'0' || c>'9')c=gc();
   int ret=0;
   while(c>='0' && c<='9'){ret=10*ret+c-48;c=gc();}
   return ret;
   }
   int main(){
    int t;
    t=read_int();
    while(t--){
     long int i,j,n,count=0,kills=0;
     int status=0,mem;
     n=read_int();
     int arr[n],arr2[50000]={0};
     for(i=0;i<n;i++){
      arr[i]=read_int();
      arr2[arr[i]]=arr2[arr[i]]+1;
      }
     for(i=1;i<50000;i++){
      //if(i%6==0){count++;}
      mem=arr2[i];
      for(j=0;j<mem;j++){
        if(kills>0 && kills%6==0)count++;
        if((i-count)<=0){
          printf("Goodbye Rick\n");
          printf("%ld\n",kills);
          status=1;
          break;
          }
         count++;
         kills++;
         }
        if(status==1){
       break;
       }
       }
      if(status==0){
       printf("Rick now go and save Carl and Judas\n");
       }
     }
     return 0;
    }
```

Sample Input

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

Sample Output

```
Rick now go and save Carl and Judas
Goodbye Rick
2
```

Result

Thus, Program " **H10** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:02:31 **Register Number:** RA1911026010033

Q. H12

Little Chandan is an exceptional manager - apart from his role in HackerEarth - as the person who has to bug everyone, in general... and if possible, try to get some work done.

He's also offered a job as the coach of the best Russian teams participating for ACM-ICPC World Finals. Now, Chandan is an extremely good coach, too. But he's a weird person who thrives on patterns in life, in general. So, he has decided that if there are n number of students in total, and he is supposed to divide them in camps of k students - he want them to be arranged in such a way that the length of names of all the students in a camp is equal.

Lknow, totally weird, right? INPUT:
The first line will contain the number of test cases. Which will be followed by two integers, n, k - denoting the number of total students, and the number of total students which will be allowed in one camp. After which, n lines will follow denoting the names of all the students who're willing to learn by the great coach. OUTPUT: If it is possible for all the students be arranged in a camp of k students, print "Possible", else print "Not possible". CONSTRAINTS:
1 <= Test Cases <= 50
1 <= N <= 1000
1 <= K <= 1000
1 <= LengthOfAString <= 100 PS: n%k will ALWAYS be equal to zero - that is, it will possible to divide the n students in equal sized camps of k.

Source Code

```
#include<iostream>
#include<map>
#include<cstring>
using namespace std;
bool check(map<int,int> M,int k){
for(auto a:M){
if((a.second%k)!=0)
return false;
}
return true;
}
int main(){
int T;
cin>>T;
while(T--){
int N,K;
cin>>N>>K;
map<int,int> M;
for(int i=1;i<=N;i++){
string str;
cin>>str;
M.insert(make_pair(i,str.length()));
}
map<int,int> M2;
for(auto a:M){
if(M2.count(a.second))
M2[a.second]++;
else
M2.insert(make_pair(a.second,1));
}
if(check(M2,K))
cout<<"Possible"<<endl;
else
cout<<"Not possible"<<endl;
}
return 0;
}
```

Sample Input

```
2
6 3
arjit
tijra
genius
chanda
ashish
arjit
4 2
bond
coder
bond
lol
```

Sample Output

```
Possible
Not possible
```

Result

Thus, Program " **H12** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:07:33 **Register Number:** RA1911026010033

Q. H16

The Monk is extremely fond of Prateek, the new HackerEarth moderator. This time, Prateek had to deal with a crazy hash function,
f(n), called the r3gz3n function, which is as follows:
N Xor (sumofdigitsofN). For example,
f(81) = 81 ^ (8+1) = 88.

Now to test the efficiency of his friend, The Monk gives Prateek a list of N numbers, and asks him to find out the following information about this list:

The value of the r3gz3n function which occurs the maximum number of times.
Number of collisions with the hash function.
Note-1: If all the values of the function are unique, print the maximum value which occurs in the list of the hashed values.
Note-2: If there are multiple hashed values which occur the maximum number of times, print the smallest hashed value with the maximum count.

Input format:
The first line will contain a single digit integer, N, denoting the number of numbers in a list. The second line will contain N integers, each separated by a space.

Output format:
Print two integers separated by a space, where the first integer denotes the value of the function which occurs the maximum number of times. Remember that if all the values of the function are unique, print the maximum value which occurs in the list. The second integer would denote the number of collisions. Constraints:
1 â‰¤ N â‰¤ 105
1 â‰¤ Ni â‰¤ 107

Source Code

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

int main()
{
unordered_map<int,int> M;
int N,maxnum=0;
cin>>N;
for(int i=0;i<N;i++)
{
int num;
cin>>num;
int sum=0;
int temp=num;
while(temp)
{
sum+=temp%10;
temp/=10;
}
num=(num^sum);
if(num>maxnum)
maxnum=num;
auto it=M.find(num);
if(it!=M.end())
it->second++;
else
M.insert(make_pair(num,0));
}
int answer=0,m=0;
for(auto it=M.begin();it!=M.end();it++)
{
answer+=it->second;
if(it->second>m)
m=it->second;
}
if(m!=0)
{
int minvalue=INT_MAX;
for(auto it=M.begin();it!=M.end();it++)
if(it->second==m)
if(minvalue>it->first)
minvalue=it->first;
cout<<minvalue;
}
else
{
int maxvalue=INT_MIN;
for(auto it=M.begin();it!=M.end();it++)
if(it->first>maxvalue)
maxvalue=it->first;
cout<<maxvalue;
}
cout<<" "<<answer;
return 0;
}
```

Sample Input

```
4
10 11 12 13
```

Sample Output

```
9 1
```

Result

Thus, Program " **H16** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:09:23 **Register Number:** RA1911026010033

Q. H18

Toru proposes Hori again this Valentine. Hori being an intelligent girl, gives Toru a long nonsense game so that he stops bothering her.
Hori gives him a string and
Q queries.
For each query there are two integers l and r.
Toru is supposed to create a new string of 26 characters for each query using following rules:
He needs to count the occurrence of each alphabet from l to r
The 1st position in the new string is for occurrence of â€˜aâ€™ , second for â€˜bâ€™ and so on.
If â€˜aâ€™ occurs x times in the range l to r , the first position in the string will be the x%26 +1, character of the english alphabet.
For each query, Toru needs to determine the longest prefix which is also the suffix of newly created string.
INPUT
The first line of input contains two integers
N and Q denoting length of the string and total number of queries.
The second line contains the string.
Q lines, one for each query follows
For each query, two integers are provided
l and r 0 OUTPUT
For each query, print the longest prefix which is also a suffix of newly created string. Print Q lines , one for each query. If no such prefix exist, print "None" for that query. CONSTRAINTS:
0String contains only lowercase English alphabets.

Source Code

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

int main()
{
int N,Q;
cin>>N>>Q;
string str;
cin>>str;
int count[N][26];
for(int i=0;i<N;i++)
for(int j=0;j<26;j++)
count[i][j]=0;
for(int i=0;i<N;i++)
{
count[i][str[i]-'a']++;
if(i!=0)
for(int j=0;j<26;j++)
count[i][j]+=count[i-1][j];
}
while(Q--)
{
int l,r;
cin>>l>>r;
l--;
r--;
int arr[26]={0};
if(l==0)
for(int i=0;i<26;i++)
arr[i]=count[r][i]%26;
else
for(int i=0;i<26;i++)
arr[i]=(count[r][i]-count[l-1][i])%26;
string answer;
for(int i=0;i<26;i++)
answer+=arr[i]+'a';
///Suffix Length
for(int i=25;i>=0;i--)
{
bool flag=true;
for(int j=26-i,k=0;j<26;j++,k++)
if(answer[j]!=answer[k])
{
flag=false;
break;
}
if(flag==true)
{
if(i==0)
cout<<"None";
for(int j=0;j<i;j++)
cout<<answer[j];
break;
}
}
cout<<endl;
}

return 0;
}
```

Sample Input

```
26 1
abcdefghijklmnopqrstuvwxyz
1 26
```

Sample Output

```
bbbbbbbbbbbbbbbbbbbbbbbbbb
```

Result

Thus, Program " **H18** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:00:27 **Register Number:** RA1911026010033

Q. H21

Xsquare got bored playing with the arrays all the time. Therefore, he has decided to play with the strings. Xsquare called a string P a "double string" if string P is not empty and can be broken into two strings A and B such that A + B = P and A = B. for eg : strings like "baba" , "blabla" , "lolo" are all double strings whereas strings like "hacker" , "abc" , "earth" are not double strings at all.

Today, Xsquare has a special string S consisting of lower case English letters. He can remove as many characters (possibly zero) as he wants from his special string S. Xsquare wants to know , if its possible to convert his string S to a double string or not.

Help him in accomplishing this task.

Note :

Order of the characters left in the string is preserved even after deletion of some characters.

Input :

First line of input contains a single integer T denoting the number of test cases. First and the only line of each test case contains a string S denoting Xsquare's special string.

Output :

For each test case, print "Yes" if it is possible to convert the given string to a double string. Print "No" otherwise.

Constraints :

1 T 100
1 |S| 100
String |S| consists of lower case english alphabets only.

Source Code

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

int main()
{
int n;
char str[100];
fscanf(stdin, "%d", &n);
while(n--) {
int a[26];

int i;
for(i=0;i<26;++i) {
a[i] = 0;
}
fscanf(stdin, "%s", str);
for( i=0; i<strlen(str); ++i) {
a[str[i]-97]++;
}

for(i=0;i<26;++i) {
if(a[i] > 1) {
fprintf(stdout, "Yes\n");
i=27;
}
}
if(i==26)
fprintf(stdout, "No\n");
}
return 0;
}
```

Sample Input

```
2
werytw
compute
```

Sample Output

```
Yes
No
```

Result

Thus, Program " **H21** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Hashing **Timestamp:** 2020-11-2 16:09:42 **Register Number:** RA1911026010033

Q. H23

After Governor's attack on prison, Rick found himself surrounded by walkers. They are coming towards him from all sides. Now, suppose Rick have infinite number of bullets with him. Suppose Rick need 1 bullet to kill each walker (yeah he is good in killing walkers). Now as soon as he kills 1 walker rest of the walkers move forward by 1 m. There are n walkers each at some distance from Rick. If any walker is able to reach Rick, he dies. Now, you need to tell if he survives or not. If he survives print "Rick now go and save Carl and Judas" else print "Goodbye Rick" followed by no.of walkers he was able to kill before he died in next line. One more thing Rick's gun can fire 6 shots without reload. It takes him 1 sec to reload and during this time walkers move 1 m forward.

[Input]
First line contains an integer t indicating number of test cases.

Next line contains an integer n denoting no.of walkers followed by n space separated integers denoting the distance of walkers from him.

[Output]
For each test case output one line denoting the answer as explained above.

[Constraints]
1<=t<=100
1<=n<=1000000
1<=dis[i]<=50000

Source Code

```
#include <iostream>
#include <algorithm>
using namespace std;
int main()
{
 int t;
   cin>>t;
 while(t--)
    {
 int n;
      cin>>n;
 int a[n+1];
     for(int i=0;i<n;i++)
       cin>>a[i];
 sort(a,a+n);
 int g=1,ag=0;
 bool alive = 1;
 for(int i=0;i<n;i++)
    {
  if(g>a[i])
       {
   alive=0;
         break;
  }
      else
       {
   ag+=1;
   g+=1;
   if(ag%6==0)
    g+=1;
  }
 }
 if(alive)
  cout<<"Rick now go and save Carl and Judas\n";
 else
    {
  cout<<"Goodbye Rick\n";
  cout<<ag<<endl;
 }
}
 return 0;
}
```

Sample Input

```
1
7
2 4 5 4 2 2 8
```

Sample Output

```
Goodbye Rick
2
```

Result

Thus, Program " **H23** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2020-11-2 13:05:13 **Register Number:** RA1911026010033

Q. LL2

Professor Madan gives a task to the sytudents such that, he asked to insert a node at the end:
The new node is always added after the last node of the given Linked List. For example if the given Linked List is 5->10->15->20->25 and we add an item 30 at the end, then the Linked List becomes 5->10->15->20->25->30.
Since a Linked List is typically represented by the head of it, we have to traverse the list till end and then change the next of last node to new node. Mandatory declarations are "void display()" and "struct node"
INPUT
First line contains the number of datas- N. Second line contains N integers(i.e, the datas to be inserted).
OUTPUT
Display the final Linked List.

Source Code

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

struct node{
    int data;
    struct node *next;
};
void display(){
 ;
}

void printList(struct node* a){
    cout << "Linked List : ";
    while(a != NULL){
        cout <<"->" <<  a->data;
        a = a->next;
    }
}

int main(){
    int size;
    cin >> size;
    struct node *temp,*head = NULL;
    for(int i = 0; i < size; i++){
        int data;
        cin >> data;
        struct node *new_node = (struct node*)malloc(sizeof(struct node));
        new_node->data = data;
        if(head == NULL){
            head =  new_node;
            temp = head;
        }else{
            temp->next = new_node;
            temp = new_node;
        }
    }
   if(size > 0)
    temp->next = NULL;
    temp = head;
    printList(head);
    return 0;
}
```

Sample Input

```
4
8 1 2 9
```

Sample Output

```
Linked List : ->8->1->2->9
```

Result

Thus, Program " **LL2** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2020-11-2 13:10:47 **Register Number:** RA1911026010033

Q. LL3

Professor sudan gives a task to the students such that, he instructed the students to new node is added after the given node of the given Linked List.

For example if the given Linked List is 5->14->16->20->25

we add an item 30 after node 15, then the Linked List becomes 5->14->16->30->20->25.
Since a Linked List is represented by the head of it,
we have to traverse the list till node and then insert the node.

Mandatory declarations are "void create()" and "struct node123 "

INPUT
First line contains the number of data- N.
Second line contains N integers(the given linked list).
Third line contains the node P after which the node to be inserted.
Fourth line contain the node X to be inserted.

OUTPUT
case 1(node P found in Linked list) :
Display the final Linked List.
case 2(node P not found in Linked list) :
Print Node not found!
Display the Linked list.

Source Code

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

struct node123 {
    int data;
    struct node123 *next;
};

void create(){
;
}

int main(){
    int size;
    cin >> size;
    struct node123 *head =NULL,*temp,*a,*b;
    while(size--){
        struct node123 *new_node = (struct node123*)malloc(sizeof(struct node123));
        int data;
        cin >> data;
        new_node->data = data;
        if(head == NULL){
            head = new_node;
            temp = head;
        }else{
            temp->next = new_node;
            temp = new_node;
        }
    }
    temp->next = NULL;
    temp = head;
    int target,tar;
    cin >> target >> tar;
    bool flag = true;
    while(temp != NULL){
        b = temp;
        a = b->next;
        if(target == temp->data){
            struct node123 *nn = (struct node123*)malloc(sizeof(struct node123));
            nn->data = tar;
            temp->next = nn;
            nn->next = a;
            flag = false;
            break;
        }
        temp = temp->next;
    }
    if(flag){
        cout << "Node not found!" << endl;
    }
    cout << "Linked List : ";
    while(head != NULL){
        cout <<"->" <<  head->data;
        head = head->next;
    }
    return 0;
}
```

Sample Input

```
4
15 20 16 10
20
17
```

Sample Output

```
Linked List : ->15->20->17->16->10
```

Result

Thus, Program " **LL3** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2020-11-2 13:05:41 **Register Number:** RA1911026010033

Q. LL4

Professor saravanan try to test the students IQ power. So he decided to gives a task to the sytudents such that, he asked, The new node is added before the given node of the given Linked List.

For example if the given Linked List is 5->10->15->20->25

we add an item 30 before node 15, then the Linked List becomes 5->10->30->15->20->25.

Since a Linked List is typically represented by the head of it, we have to traverse the list till node and then insert the node.

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

INPUT
First line contains the number of datas- N. Second line contains N integers(the given linked list).
Third line contains the node P before which the node to be inserted. Fourth line contain the node X to be inserted.
OUTPUT
case 1(node P found in Linked list) :
Display the final Linked List.
case 2(node P not found in Linked list) :
Print Node not found!
Display the Linked list.

Source Code

```
#include<bits/stdc++.h>
using namespace std;
 struct Node{
    int data;
    struct Node *next;
};

void Create(){
;
}

int main(){
    int size;
    cin >> size;
    struct Node *temp,*head = NULL,*a,*start;

    for(int i = 0; i < size; i++){
        int data;
        cin >> data;
        struct Node *new_Node = (struct Node*)malloc(sizeof(struct Node));
        new_Node->data = data;
        if(head == NULL){
            head =  new_Node;
            temp = head;
        }else{
            temp->next = new_Node;
            temp = new_Node;
        }
    }
   if(start==NULL)
    {
     ;
    }
   if(size != 0)
    temp->next = NULL;
    temp = head;
    bool flag = true;
        int target,tar;
        cin >> target >> tar;
        while(temp != NULL){
            if(target == temp->data){
                flag = false;
                struct Node *nn = (struct Node*)malloc(sizeof(struct Node));
                nn->data = tar;
                 if(a == NULL){
                    head = nn;
                }else{
                    a->next = nn;
                 }
                nn->next = temp;
                break;
            }
            a = temp;
            temp = temp->next;
        }
        if(flag){
            cout << "Node not found! " << endl;
        }
    cout << "Linked List : ";
         while(head != NULL){
        cout <<"->" <<  head->data;
        head = head->next;
         }

    return 0;
}
```

Sample Input

```
4
9 77 12 6
12
8
```

Sample Output

```
Linked List : ->9->77->8->12->6
```

Result

Thus, Program " **LL4** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2020-11-2 13:27:05 **Register Number:** RA1911026010033

Q. LL6

Professor SIVA gives a task to the students such that, 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 D times from the beginning of the given linked list.

For example if the given Linked List is 5->10->15->20->25 and remove 2 nodes,
then the Linked List becomes 15->20->25.

Mandatory conditions are " struct node" , "Create()"

INPUT
First line contains the number of datas- N. Second line contains N integers(the given linked list).
Third line contains no. of nodes to be deleted.

OUTPUT
Display the Linked list.

Source Code

```
// deleted the nodes
#include<bits/stdc++.h>
#include<iostream>
using namespace std;

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

void printList(struct node *a){
    cout << "Linked List : ";
    while(a != NULL){
        cout << "->"<< a->data ;
        a = a->next;
    }
    cout << endl;
}
void Create(){
}

int main(){
    int size;
    cin >> size;
    int b = size;
    struct node *head = NULL,*temp,*a;
    while(size--){
        struct node* new_node =(struct node*)malloc(sizeof(struct node));
        int data;
        cin >> data;
        new_node->data =data;
        if(head == NULL){
            head = new_node;
            temp = head;
        }else{
            temp->next = new_node;
            temp = new_node;
        }
    }
    temp->next = NULL;
    temp =  head;
    // printList(head);
    int times;
    cin >> times;
   if(b < times){
     times = b;
    }
    while(times--){
        a = temp;
        head = temp->next;
        delete(a);
        temp=temp->next;
    }
    printList(head);

    return 0;
}
```

Sample Input

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

Sample Output

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

Result

Thus, Program " **LL6** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2020-11-2 16:18:17 **Register Number:** RA1911026010033

Q. LL12

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 a€ xa€™ in a given singly linked list (Iterative). 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<bits/stdc++.h>
#include<iostream>
using namespace std;
struct node
{
  int data;
  struct node *next;
};
int main(){
  int size;
  cin >> size;
  struct node *temp,*head = NULL;
  for(int i = 0; i < size; i++){
    int data;
    cin >> data;
    struct node *new_node=(struct node*)malloc(sizeof(struct node));
    new_node->data = data;
    if(head == NULL){
      head =  new_node;
      temp = head;
    }else{
      temp->next = new_node;
      temp = new_node;
    }
  }
  if(size != 0)
    temp->next = NULL;
  temp = head;
  bool flag = true;
  int target;
  cin >> target;
  while(temp!=NULL){
    if(target==temp->data){
      flag = false;
      cout << "Yes\n";
      break;
    }
    temp = temp->next;
  }
  if(flag){
    cout << "No" << endl;
  }
    return 0;
}
```

Sample Input

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

Sample Output

```
Yes
```

Result

Thus, Program " **LL12** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2020-11-2 13:23:12 **Register Number:** RA1911026010033

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

```
// searching in linked list
#include<bits/stdc++.h>
#include<iostream>
using namespace std;

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

int main(){
    int size;
    cin >> size;
    struct node *temp,*head = NULL,*a = NULL;
    for(int i = 0; i < size; i++){
        int data;
        cin >> data;
        struct node* new_node=(struct node*)malloc(sizeof(struct node));
        new_node->data = data;
        if(head == NULL){
            head =  new_node;
            temp = head;
        }else{
            temp->next = new_node;
            temp = new_node;
        }
    }
    if(size != 0)
    temp->next = NULL;
    temp = head;
    bool flag = true;
        int target;
        cin >> target;
        while(temp != NULL){
            if(target == temp->data){
                flag = false;
                cout << "Yes\n";
                break;
            }
            temp = temp->next;
        }
        if(flag){
            cout << "No" << endl;
        }
    return 0;
}
```

Sample Input

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

Sample Output

```
Yes
```

Result

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

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2020-11-2 13:04:00 **Register Number:** RA1911026010033

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

```
// reverse the list
#include<bits/stdc++.h>
#include<iostream>
using namespace std;
struct node{
    int data;
    struct node *next;
};

void printList(struct node *a,int index){
    cout << "Linked list : ";
    int count = 0,value = 0;
    while(a != NULL){
        count++;
        cout << "-->" << a->data;
        if(count == index){
            value = a->data;
        }
        a = a->next;
    }
    cout << endl;
   if(value > 0){
     cout << "Node at index=" << index << " : " << value << endl;
    }else{
      cout << "Invalid Index!"<< endl;
    }
}

void push(struct node** next_address,int data){
    struct node* new_node =(struct node*) malloc(sizeof(struct node));
    new_node->data = data;
    new_node->next = (*next_address);
    *next_address = new_node;
}

int main(){
    int size;
    cin >> size;
    struct node* head = NULL,*a;
    for(int i = 0; i < size; i++){
        int data;
        cin >> data;
        push(&head,data);
    }
    int index;
    cin >> index;
    printList(head,index);
}
```

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:** Linked List **Timestamp:** 2020-11-2 13:14:59 **Register Number:** RA1911026010033

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<bits/stdc++.h>
#include<iostream>
using namespace std;
struct node
{
int data;
struct node *next;
};

void printList(struct node *a,int size){
   int middle,count = 0;
   bool flag =true;
   size = (size % 2  == 0)?(size /2) + 1 : (size + 1)/2;
   cout << "Linked list : ";
   while(a != NULL){
      count += 1;
      if(size == count){
         middle = a->data;
      }
      cout << "-->"<< a->data ;
      a = a->next;
   }
   cout << endl;
   cout << "The middle element is [" << middle << "]" << endl;
}

int main(){
   int size;
   cin >> size;
   int b = size;
   struct node *head = NULL;
   while(size--){
struct node* new_node =(struct node*) malloc(sizeof(struct node));
      int data;
      cin >> data;
      new_node->data = data;
      new_node->next = head;
      head = new_node;
   }
   printList(head,b);
   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:** 2020-11-2 13:03:31 **Register Number:** RA1911026010033

Q. LL18

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 generate a program a function to reverse a linked list . Given pointer to the head node of a linked list, the task is to reverse the linked list.

For example,
Input : Head of following linked list
1->2->3->4->NULL
Output : Linked list should be changed to,
4->3->2->1->NULL

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).
OUTPUT
Display the Linked List.
Display the reversed Linked List.

Source Code

```
#include<bits/stdc++.h>
#include<iostream>
using namespace std;
 struct node
 {
    int data;
    struct node* next;
};


void printList(struct node *s){
    while(s != NULL){
        cout <<" " <<s->data ;
        s = s->next;
    }
    cout << endl;
}

void reverseList(struct node **s){
    struct node *current = *s,*prev =NULL,*next_adress;
    while(current!= NULL){
        next_adress = current->next ;
        current->next = prev;
        prev = current;
        current = next_adress;
    }
    *s = prev;
}

int main(){
    int size;
    cin >> size;
    struct node *head = NULL,*temp = NULL;
    while(size--){
        struct node* new_node =(struct node*) malloc(sizeof(struct node));
        int data;
        cin >> data;
        new_node->data = data;
        if(head == NULL){
            head = new_node;
            temp =head;
        }else{
            temp->next = new_node;
            temp = new_node;
        }
    }
    temp->next = NULL;
    temp = head;
    cout << "Linked list :";
    printList(temp);

    reverseList(&temp);

    cout << "Reversed Linked list :";
    printList(temp);

    return 0;
}
```

Sample Input

```
4
9 77 12 6
```

Sample Output

```
Linked list : 9 77 12 6
Reversed Linked list : 6 12 77 9
```

Result

Thus, Program " **LL18** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Linked List **Timestamp:** 2020-11-2 13:01:53 **Register Number:** RA1911026010033

Q. LL22

Once upon a time there was a Princess. Many a suitor came to the palace to win her hand in marriage, but it seemed to the Princess that each one of them looked at her without really seeing her at all. Princes asked a question to the suitor.In your Application form some datas were repeated during the input. The you need to remove the concurrent repeating of datas in the application. for that prices asked suitor to generate a function to remove the duplicated-concurrent and display the list using Singly linked list.

For example, if the input is 1 1 2 2 5 3 2, then the output should be 1 2 5 3 2.

Mandatory declaration is "struct node* current = head;"

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

Source Code

```
// searching in linked list
#include<bits/stdc++.h>
#include<iostream>
using namespace std;

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

void printList(struct node *a){
    cout << "List : ";
    while(a != NULL){
        cout << "->" <<  a->data;
        a = a->next;
    }
    cout << endl;
}

int main(){
    int size;
    cin >> size;
    struct node *temp,*head = NULL,*current = NULL;
    for(int i = 0; i < size; i++){
        int data,previous;
        cin >> data;
        struct node* current = head;
        if(previous == data){
            continue;
        }
        struct node* new_node =(struct node*)malloc(sizeof(struct node));
        new_node->data = data;
        if(head == NULL){
            head =  new_node;
            temp = head;
        }else{
            temp->next = new_node;
            temp = new_node;
        }
        previous = data;
    }
    temp->next = NULL;
    temp = head;
    printList(head);
    return 0;
}
```

Sample Input

```
7
9 1 1 3 6 2 2
```

Sample Output

```
List : ->9->1->3->6->2
```

Result

Thus, Program " **LL22** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2020-11-2 15:16:33 **Register Number:** RA1911026010033

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

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:** 2020-11-2 15:14:47 **Register Number:** RA1911026010033

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:** 2020-11-2 14:59:23 **Register Number:** RA1911026010033

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

```
// print the binary numbers form given 1 ro given number
#include <bits/stdc++.h>
using namespace std;
void gen(int n)
{
 long rem, base = 1, answer = 0, count = 0;

    while (n > 0)
    {
        rem = n % 2;
        if (rem == 1)
        {
            count++;
        }
        answer = answer + rem * base;
        n = n / 2;
        base = base * 10;
    }
    cout << answer <<" ";
}
int main()
{
  int testcase;
  cin >> testcase;
  while(testcase--){
    int upto;
    cin >> upto;
    for(int i=1;i<=upto;i++){
        gen(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

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2020-11-2 15:13:05 **Register Number:** RA1911026010033

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:** 2020-11-2 15:07:47 **Register Number:** RA1911026010033

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

int FIND(int ,int );
int main(){
    int t;
    cin >> t;
    while(t--){
        int a, b;
        cin >> a >> b;
        if(abs(a - b) == 1 ){
            cout << "yes" << endl;
            continue;
        }
        a = a % 3;
        b = b % 3;
        if(abs(a - b) == 1 ){
            cout << "yes" << endl;
        }else{
            cout << "no" << endl;
        }
    }
    return 0;
}
```

Sample Input

```
2
3 1
6 3
```

Sample Output

```
yes
no
```

Result

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

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2020-11-2 14:21:01 **Register Number:** RA1911026010033

Q. Q13

A queue is an abstract data type that maintains the order in which elements were added to it, allowing the oldest elements to be removed from the front and new elements to be added to the rear. A queue has the following operations:
Enqueue: add a new element to the end of the queue.
Dequeue: remove the element from the front of the queue and return it.
In this challenge, you must first implement a queue using two stacks. Then process Q queries, where each query is one of the following types:
1: Enqueue element X into the end of the queue.
2: Dequeue the element at the front of the queue.
3: Print the element at the front of the queue.

Mandatory declaration is "while(ELE--)"

INPUT:
The first line contains a single integer, Q denoting the number of queries.
Each line of the subsequent lines contains a single query in the form described in the problem statement above. All three queries start with an integer denoting the query, but only query type 1 is followed by an additional space-separated value, denoting the value X to be enqueued into queue.

OUTPUT:
For each query of type 3 , print the value of the element at the front of the queue on a new line.

Source Code

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

int main() {
    stack<int> Front,Rear;
    int ELE;
    cin >> ELE;
    while(ELE--)
    {
        int type, x;
        cin >> type;
        if(type == 1)
        {
            cin >> x;
            Rear.push(x);
        }
        else
        {
            if(Front.empty())
            {
                while(!Rear.empty())
                {
                    Front.push(Rear.top());
                    Rear.pop();
                }
            }
            if(!Front.empty())
            {
                if(type == 2) Front.pop();
                if(type == 3) cout << Front.top() << endl;
            }
        }
    }
    return 0;
}
```

Sample Input

```
10
1 42
2
1 14
3
1 28
3
1 60
1 78
2
2
```

Sample Output

```
14
14
```

Result

Thus, Program " **Q13** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2020-11-2 14:18:37 **Register Number:** RA1911026010033

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

Q. Q18

Given a Binary Tree as pre-order traversal input which has both positive and negative nodes, the task is to find maximum sum level in it.Using level order traversal of tree and queue data structure,solve the problem.For example , A binary tree with pre-order traversal as {4,2,-1,3,6,5,7}:

Sum of all nodes of 0th level is 4
Sum of all nodes of 1th level is 8
Sum of all nodes of 2nd level is 14
Hence maximum sum is 14

Mandatory declarations are "struct node *Right;", "struct node *Left;"

INPUT:
The first line of input is the number of nodes in tree N and then followed by the nodes as in pre-order traversal.

OUTPUT:
The output should be the maximum level sum of the tree.

Source Code

```
#include<bits/stdc++.h>
using namespace std;
struct node{
 int data;
   struct node *Left;
   struct node *Right;
   int height = 0;
};


struct node*root = NULL;
void insert(int new_data){

 struct node* new_node =(struct node*)malloc(sizeof(struct node));
 new_node->data = new_data;
   new_node->Left = NULL;
   new_node->Right = NULL;
   if(root==NULL){
     root = new_node;
       return;
    }
   struct node* temp=root;
   while(temp!=NULL){
     if(new_data>temp->data){
        if(temp->Right==NULL){
            temp->Right = new_node;
              return;
          }
         else{
          temp = temp->Right;
         }
       }
       else {
         if(temp->Left==NULL){
            temp->Left = new_node;
              return;
          }
         else{
          temp = temp->Left;
         }

       }

    }
}
int levelordertraversal(){

 queue<node*> q;
 struct node*curr;
   q.push(root);
   q.push(NULL);
   int sum = 0;
   int max = 0;
   while(q.size()>1){
     curr = q.front();
       q.pop();
       if(curr==NULL){
        q.push(NULL);
        sum = 0;
    }
      else{
       if(curr->Left){
         q.push(curr->Left);
       }
       if(curr->Right){
         q.push(curr->Right);

       }
       sum += curr->data;

      }
      if(sum>max){
       max = sum;

     }

    }
  return max;
}


int main() {
 int n;
  cin>>n;
   for(int i=0;i<n;i++){
     int x;
       cin>>x;
       insert(x);
    }
      int y = levelordertraversal();
   cout<<y;
       return 0;
}
```

Sample Input

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

Sample Output

```
14
```

Result

Thus, Program " **Q18** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Queue **Timestamp:** 2020-11-2 14:15:28 **Register Number:** RA1911026010033

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 big,bigger,biggest,temp;
    for(i=0;i<N;i++)
    {
     scanf("%d",&array123[i]);
    }
    printf("-1\n-1\n");
    big=array123[0];
    bigger=array123[1];
    if(array123[0]>bigger)
    {
     bigger=array123[0];
     big=array123[1];
    }
    biggest=array123[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(array123[i]>biggest)
  {
   big=bigger;
   bigger=biggest;
   biggest=array123[i];
  }
  else if(array123[i]>bigger)
  {
   big=bigger;
   bigger=array123[i];
  }
  else if(array123[i]>big)
  {
   big=array123[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:** Searching **Timestamp:** 2020-11-1 17:53:05 **Register Number:** RA1911026010033

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 <iostream>
#include<bits/stdc++.h>
using namespace std;
int main() {
 int size;
   cin >> size;
   vector<int>data;
   for(int i = 0; i < size; i++){
      int a;
      cin >> a;
      data.push_back(a);
    }
   int element;
   cin >> element;
   for(int i = 0; i < size; i++){
     if(element == data[i]){
         cout << element << " found at location " << i + 1 << endl;
           break;
        }
        if(i == size - 1){
         cout << "Not found " << element << " is not present in the list" << endl;
        }
    }
 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:** 2020-11-1 17:53:33 **Register Number:** RA1911026010033

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 <iostream>
#include<bits/stdc++.h>
using namespace std;
int main(){
  int size;
  cin >> size;
  vector<int>data;
  for(int i = 0; i < size; i++)
  {
    int a;
    cin >> a;
    data.push_back(a);
  }
  int element;
  cin >> element;
  for(int i = 0; i < size; i++){
    if(element == data[i] - 1 ||element ==data[i] + 1 ){
      cout << i + 1 << endl;
      break;
    }
    if(i == size - 1){
      cout << -1 << endl;
    }
  }
  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:** 2020-11-1 17:55:10 **Register Number:** RA1911026010033

Q. SER5

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

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

Source Code

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

void printSums(int N){
    int start = 1,end = 1,sum = 1;
    while(start <= N/2){
        if(sum < N){
            end = end + 1;
            sum += end;
        }else if(sum > N){
            sum -= start;
            start = start + 1;
        }else{
            for(int i = start; i <= end; i++){
                cout << i << " ";
            }
            cout << endl;
            sum = sum - start;
            start = start + 1;
        }
    }
}

int main(){
    int N;
    cin >> N;
    printSums(N);
    return 0;
}
```

Sample Input

125

Sample Output

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

Result

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

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2020-11-1 17:52:34 **Register Number:** RA1911026010033

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 mannar. 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<bits/stdc++.h>
using namespace std;

void findClosest(int A[], int B[], int C[], int p, int q, int r){

    int difference = INT_MAX ;


    int mini = INT_MIN, minj = INT_MIN, mink = INT_MIN;
    int i=0,j=0,k=0;
    int maximum,minimum;
    while (i < p && j < q && k < r){
        minimum = min(A[i], min(B[j], C[k]));
        maximum = max(A[i], max(B[j], C[k]));
        if(maximum-minimum < difference){
            mini = i, minj = j, mink = k;
            difference = maximum - minimum;
        }
        if (difference == 0) break;
        if (A[i] == minimum) i++;
        else if (B[j] == minimum) j++;
        else k++;
    }
    cout << A[mini] << " " << B[minj] << " " << C[mink];
}

int main(){
    int p,r;
    int q;
    cin >> p;
    int A[p] ;
    for(int i = 0; i < p; i++){
        cin >> A[i];
    }
    cin >> q;
    int B[q] ;
    for(int i = 0; i < q; i++){
        cin >> B[i];
    }
    cin >> r;
    int C[r] ;
    for(int i = 0; i < r; i++){
        cin >> 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:** 2020-11-1 17:54:12 **Register Number:** RA1911026010033

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<bits/stdc++.h>
int main(){
    int n,i,j,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. SER9

World cup team squad annocunced for level one matches. There are a lot of Poke'mons who are jealous of the fact that they do NOT have any specialty, they're the... normal type of Poke'mon. But, what they fail to realize is that their power is their normalcy, the ability to think, rationalize and then act.

But, they do have an additional power... Poke'mons like Jigglypuff - which are normal, can figure out if a trainer is real or is a part of Team Rocket. And they need to use their power to a great extent.

In an array, which consists of
N
N elements, A1, A2, ..., AN, if a subarray has the total number of distinct elements as that of the original array, that determines the presence of Team Rocket.

You've to help the normal type Poke'mons in figuring out the total number of subarrays having total number of distinct elements same as that of the original array.

Input format:
The first line of the input will consist of a single integer N
The next line will consist of N integers A1, A2, ... , AN.
Output format: Output the answer to the problem.

Sample Input:
5
1 2 2 1 1

SAMPLE OUTPUT
8

Explanation

Consider an example,
1 2 2 2 6 1 1 2
Here num_dist is 3 so for i=0 our while loop runs till element 6(index is 4 and it becomes maxi) at this point s.size()==num_dist so whatever elements there after 6 are already present in our set 's' so we increment ans here by (n-maxi+1) because following are the possible subsets.
1 2 2 2 6 1 1 2
1 2 2 2 6 1 1
1 2 2 2 6 1
1 2 2 2 6 1
1 2 2 2 6

so ans=ans+5

for i=1 the while loop runs till element 1 (index is 5 and it becomes maxi) at this point s.size()==num_dist because in iteration '0' we erased element 1 from set s(line 63 and 64). now possible subsets are as follows.
2 2 2 6 1 1 2
2 2 2 6 1 1
2 2 2 6 1
2 2 2 6 1

so ans=ans+4

same for i=2,3,.....

Mandatory method need to be used
long long int countDistictSubarray(int arr[], int n)

Source Code

```
#include <iostream>
using namespace std;
long long int countDistictSubarray(int arr[], int n){
int freq[n],i,j,c;
  for(i=0;i<n;i++)
    freq[i]=-1;
  for(i=0;i<n;i++){
    c=1;
    for(j=i+1;j<n;j++){
      if(arr[i]==arr[j]){
        c+=1;
        freq[j]=0;
      }
    }
    if(freq[i]!=0)
      freq[i]=c;
  }

  int count=0;

  for(i=0;i<n;i++){
    if(freq[i]>=1)
      count++;
  }
  return count;
}
int main(){
  int n,i,j,c;
  cin>>n;
  int arr[n];
  for(i=0;i<n;i++)
    cin>>arr[i];


  int ct=0,k,counter=0;
  for(i=0;i<n;i++){

    for(j=i;j<n;j++){
      int b[j-i+1];
      for(k=i;k<=j;k++)
        b[k]=arr[k];

      if(countDistictSubarray(b,(j-i+1))==countDistictSubarray(arr,n))

        counter++;
    }
  }
  cout<<counter;
  return 0;
}
```

Sample Input

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

Sample Output

```
2
```

Result

Thus, Program " **SER9** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2020-11-1 18:03:16 **Register Number:** RA1911026010033

Q. SER10

Bishu went to fight for Coding Club. There were N soldiers with various powers. There will be Q rounds to fight and in each round Bishu's power will be varied. With power M, Bishu can kill all the soldiers whose power is less than or equal to M(<=M). After each round, All the soldiers who are dead in previous round will reborn.Such that in each round there will be N soldiers to fight. As Bishu is weak in mathematics, help him to count the number of soldiers that he can kill in each round and total sum of their powers.

1<=N<=10000

1<=power of each soldier<=100

1<=Q<=10000

1<=power of bishu<=100

Source Code

```
#include<bits/stdc++.h>
using namespace std;
int main(){
   int size;
   cin >> size;
   vector<int>powers;
   for(int i = 0; i < size; i++){
     int a;
       cin >> a;
       powers.push_back(a);
    }
 int q;
   cin >> q;
   while(q--){
     int a;
       cin >> a;
      int sum = 0,count = 0;
       for(int i = 0; i < size; i++){
         if(a >= powers[i]){
             count++;
               sum += powers[i];
           }
       }
      cout << count  << " " << sum << endl;
    }

  return 0;
}
```

Sample Input

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

Sample Output

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

Result

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

Course: DATA-STRUCTURE **Session:** Searching **Timestamp:** 2020-11-1 18:02:49 **Register Number:** RA1911026010033

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 <iostream>
using namespace std;
int main() {
  int a,b,c,i,p;
  int brr[1000];
  cin>>a>>b>>c;
 p=c;int arr[a];
  for(i=0;i<a;i++)
  {
    cin>>arr[i];
  }
  int j,k,z=0;
  i=0;
  for(k=1;k<=b;k++)
  {c=p;
  for(j=0;j<a;j++)
  {

    if(arr[j]==k && c>0)
    {
      z++;
      c--;

    }
    if(c==0)
      break;

  }}int l=0;
  for(i=0;i<a;i++)
  {if(arr[i]>b && c!=0)
  {
    l++;
  c--;}}int x=1,y=2,q;
  while(x!=y)
  {
    q=0;
    x++;
  }
      q=a-z-l;

  printf("%d",q);
  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:** 2020-11-1 18:04:54 **Register Number:** RA1911026010033

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<bits/stdc++.h>
using namespace std;
typedef long long ll;
int main()
{
int n,i,co=0;
cin>>n;
ll a,b,c;
vector<ll>sides[n];
map<vector<ll>,int>m;
for(i=0;i<n;i++)
{
scanf("%lld%lld%lld",&a,&b,&c);
sides[i].push_back(a);
sides[i].push_back(b);
sides[i].push_back(c);
sort(sides[i].begin(),sides[i].end());
m[sides[i]]++;
}
map<vector<ll>,int>::iterator it;
for(it=m.begin();it!=m.end();it++)
{
if(it->second==1)
++co;
}
printf("%d\n",co);
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:** 2020-11-1 17:54:38 **Register Number:** RA1911026010033

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

int main()
{
   int n;
   scanf("%d",&n);
   for(int i=1;i*(i+1)<n*2;i++){
      int t=n*2-i*(i+1);
      int k=(int)sqrt(t);
      if(k*(k+1)==t){
         printf("YES\n");
         return 0;
      }
   }
   printf("NO\n");
}
```

Sample Input

256

Sample Output

YES

Result

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

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2020-11-2 10:57:49 **Register Number:** RA1911026010033

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 swap(int *xp, int *yp)
{
int temp = *xp;
*xp = *yp;
*yp = temp;
}


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

Sample Input

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

Sample Output

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

Result

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

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2020-11-2 11:05:00 **Register Number:** RA1911026010033

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 <stdio.h>

void swap(int *xp, int *yp)
{
int temp = *xp;
*xp = *yp;
*yp = temp;
}


void bubbleSort(int arr[], int n)
{
int i, j;int a;
for (i = 0; i < n-1; i++)
{
  a=i;
  if(a==3){
    for (a=0; a< n; a++){
  printf("%d ", arr[a]);
  }
  printf("\n");
   }


for (j = 0; j < n-i-1; j++)
 if (arr[j] > arr[j+1])
 swap(&arr[j], &arr[j+1]);
 }
}



int main()
{
int t;scanf("%d",&t);
 int i;int arr[t];
 for(i=0;i<t;i++)
 {  scanf("%d",&arr[i]);}
bubbleSort(arr, t);
printf("Sorted array:");
for (i=0; i < t; i++)
 printf("%d ", arr[i]);
printf("\n");
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:** 2020-11-2 10:55:45 **Register Number:** RA1911026010033

Q. SORT4

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

Source Code

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

void InSort(int arr[],int n){
    sort(arr,arr+n);
    cout << "Sorted Array:";
    for(int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main(){
    int size;
    cin >> size;
    int arr[size],a[size];
    for(int i = 0; i < size; i++){
        cin >> arr[i];
        a[i] = arr[i];
    }
   sort(a,a+3);
   for(int i = 0; i < 3; i++){
     cout << a[i] << " ";
    }
   for(int i = 3; i < size; i++){
     cout << arr[i] << " ";
    }
   cout << endl;
    InSort(arr,size);
    return 0;
}
```

Sample Input

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

Sample Output

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

Result

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

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2020-11-2 11:03:33 **Register Number:** RA1911026010033

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

void result(int a[],int b[],int n){
    sort(a,a + n);
   sort(b,b+n,greater<int>());
    long  int sum = 0;
    for(int  i = 0; i < n; i++){
        sum += a[i] * b[i];
    }
    cout << sum << endl;
}


int main(){
    int t;
    cin >> t;
    while(t--){
        int size;
        cin >> size;
        int arr1[size],arr2[size];
        for(int i = 0; i < size; i++){
            cin >> arr1[i];
        }
        for(int i = 0; i < size; i++){
            cin >> arr2[i];
        }
        result(arr1,arr2,size);
    }
    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:** 2020-11-2 10:53:47 **Register Number:** RA1911026010033

Q. SORT8

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

Input:

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

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

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

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

Output:

Print the maximum product of two numbers possible.

Constraints:

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

Source Code

```
#include<bits/stdc++.h>
#include<iostream>
using namespace std;
int main(){
    int t;
    cin >> t;
    while(t--){
        int size;
        cin >> size;
        int fmax = -1,smax = -1;
        for(int i = 0; i < size; i++){
            int a;
            cin >> a;
            if(fmax < a){
                smax = fmax;
                fmax = a;
            }else if(a > smax){
             smax = a;
            }
        }
        cout << fmax * smax << endl;
    }
    return 0;
}
```

Sample Input

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

Sample Output

```
4200
```

Result

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

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2020-11-2 10:50:42 **Register Number:** RA1911026010033

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>
#include<iostream>
using namespace std;
int main(){
    int t;
    cin >> t;
    while(t--){
        int size;
        cin >> size;
        vector<int> numbers;
        for(int i = 0; i < size; i++){
            int a;
            cin >> a;
            numbers.push_back(a);
        }
        int index1,index2;
        cin >> index1 >> index2;
        int sum = 0;
        sort(numbers.begin(),numbers.end());
       index1 -= 1;
       index2 -= 1;
        for(int i = index1 + 1 ; i < index2; i++){
            sum += numbers[i];
        }
        cout << sum << endl;
    }

    return 0;
}

//1 5 12 22 32 45
//0 1 2  3   4  5
```

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:** 2020-11-2 10:49:16 **Register Number:** RA1911026010033

Q. SORT10

LALU wanted to purchase a laptop so he went to a nearby sale.There were n Laptops at a sale.
Laptop with index i costs ai rupees.
Some Laptops have a negative price â€" their owners are ready to pay LALU if he buys their useless Laptop.
LALU can buy any Laptop he wants. Though he's very strong, he can carry at most m Laptops, and he has no desire to go to the sale for the second time.
Please, help LALU find out the maximum sum of money that he can earn.
Input:
First line of the input contains T denoting the number of test cases.Each test case has 2 lines :
First line has two spaced integers n m.
second line has n integers [a0...ai...an-1].
Output:
The maximum sum of money that LALU can earn, given that he can carry at most m Laptops.

Constraints:
1â‰¤Tâ‰¤10
1â‰¤n,mâ‰¤100
-1000â‰¤aiâ‰¤1000

Source Code

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int t;
    cin >> t;
    while(t--){
        int size,items;
        cin >> size >> items;
        int answer = 0;
        vector<int>laptops;
        for(int i = 0; i < size; i++){
            int a;
            cin >> a;
            laptops.push_back(a);
        }
        sort(laptops.begin(),laptops.end());
        for(int i = 0; i < items; i++){
          if(laptops[i] < 0)
            answer += laptops[i];
        }
        cout << abs(answer) << endl;
    }

    return 0;
}
```

Sample Input

```
1
5 3
-6 0 35 -2 4
```

Sample Output

```
8
```

Result

Thus, Program " **SORT10** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2020-11-1 18:06:27 **Register Number:** RA1911026010033

Q. SORT11

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

Source Code

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

Sample Input

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

Sample Output

```
3 7
```

Result

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

Course: DATA-STRUCTURE **Session:** Sorting **Timestamp:** 2020-11-1 18:07:02 **Register Number:** RA1911026010033

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

bool sortinrev(const pair<int,int> &a,const pair<int,int> &b) {
      return (a.first > b.first);
}

int main(){
   int t;
   cin >> t;
   while(t--){
      int size;
      cin >> size;
      vector<pair<int,int> > data;
      map<int,int> freq;
      for(int i = 0; i < size; i++){
         int a;
         cin >> a;
         freq[a]++;
      }
      for(auto it = freq.begin(); it != freq.end(); it++){
         data.push_back(make_pair(it->second,it->first));
      }
      sort(data.begin(),data.end(),sortinrev);

      for(int i = 0 ; i < data.size(); i++){
         for(int j = 1; j <= data[i].first; j++){
            cout << data[i].second << " ";
         }
      }
      cout << endl;
   }
   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:** 2020-11-2 10:50:13 **Register Number:** RA1911026010033

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<bits/stdc++.h>
#include<iostream>
using namespace std;
int main(){
    int t;
    cin >> t;
    while(t--){
        int m,n;
        cin >> m >> n;
        vector<int> merged;
        for(int i = 0; i < m + n; i++){
            int a;
            cin >> a;
            merged.push_back(a);
        }
        sort(merged.begin(),merged.end(),greater<int>());
        for(int i = 0; i < merged.size(); i++){
            cout << merged[i] << " ";
        }
        cout << endl;
    }
    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:** Stack **Timestamp:** 2020-11-2 13:38:05 **Register Number:** RA1911026010033

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

```
// implement stack using linked list
#include<bits/stdc++.h>
#include<iostream>
using namespace std;

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

struct MYStack *createMyStack(){
    struct MYStack *a;
    return a;
}

void push(struct MYStack**top,int data){
    struct MYStack *createMyStack = (struct MYStack*)malloc(sizeof(struct MYStack));
    createMyStack->data = data;
    createMyStack->next = *top;
    *top = createMyStack;
}
struct MYStack* pop(struct MYStack *top){
    struct MYStack *a;
    a = top;
    top = top->next;
    a->next = NULL;
    cout << "Item popped is "<< a->data << endl;
    delete(a);
    return top;
}
void printMiddleElement(struct MYStack *s,int size){
    int count = 0,middle;
    size = (size/2) + 1;
    while(s != NULL){
        count++;
        if(count == size){
            middle = s->data;
        }
        // cout << s->data <<"  ";
        s = s->next;
    }
    cout << "Middle Element is "<< middle <<  endl;
}

int main(){
    int size;
    cin >> size;
    int b = size;
    struct MYStack *top = NULL;
    while(size--){
        int data;
        cin >> data;
        push(&top,data);
    }
    b--;
    top = pop(top);
    top = pop(top);
    b--;
    printMiddleElement(top,b);
    return 0;
}
```

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:** 2020-11-2 14:14:48 **Register Number:** RA1911026010033

Q. ST4

The Director of Engineering College decided to make stock audit, So he called all the engineering technical faculties to generate the stock span problem is a financial problem where we have a series of n daily price quotes for a stock and we need to calculate span of stocks price for all n days.
The span Si of the stocks price on a given day i is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the current day is less than or equal to its price on the given day.

For example,

if an array of 7 days prices is given as {100, 80, 60, 70, 60, 75, 85},
then the span values for corresponding 7 days are {1, 1, 1, 2, 1, 4, 6}

Mandatory Declaration is "void printArray(int arr[],int n)"

Input:
The first line of each test case is N,N is the size of array.
The second line of each test case contains N input C[i].

Source Code

```
// Stack And Span problem
#include<bits/stdc++.h>
#include<iostream>
#define printVector(vect,dataStruct) for(vector<dataStruct> :: iterator it = vect.begin(); it !=vect.end();it++){cout << *it << " ";}
using namespace std;

void printArray(int arr[],int n){
}
int main(){
    int size;
    cin >> size;
    vector<int>stock,span(size);
    for(int i = 0; i < size; i++){
        int price;
        cin >> price;
        stock.push_back(price);
    }
    stack<int>S;
    S.push(0);
    span[0] = 1;
    for(int  i = 1; i < size; i++){
        while(!S.empty() && stock[S.top()] <= stock[i]){
            S.pop();
        }
        if(S.empty()){
            span[i] = i + 1;
        }else{
            span[i] = i - S.top();
        }
      S.push(i);
    }
    printVector(span,int);
    // time complexity is O(n)
    return 0;
}
```

Sample Input

```
7
100 80 60 70 60 75 85
```

Sample Output

```
1 1 1 2 1 4 6
```

Result

Thus, Program " **ST4** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2020-11-2 13:34:17 **Register Number:** RA1911026010033

Q. ST7

Mathematical professor is teaching symbolic representation problem ie. balanced parenthesis problem. After completing a lecture, the professor asked a viya vice question to the user. one of the student he faced a question like, a string that contains a mathematical equation where you have to check, whether the given string has balanced parentheses or not. Balanced parenthesis means that the given string should have equal number of opening and closing parenthesis.

We can check this using stack. the same task the students need to perform in the stack operation. student need to maintain a stack where we store a parenthesis, whenever we encounter an opening parenthesis in the string (can push in stack)and pop a parenthesis from the stack whenever we encounter a closing parenthesis.

INPUT:
The first line of the input must contain the string size
the second line of the input must contain the expression with parentheses which has to be checked if it is balanced or not .

Source Code

```
// balance or not
#include<bits/stdc++.h>
#include<iostream>
using namespace std;
int main(){
   int size;
  cin >> size;
    string str;
    cin >> str;
    stack<char>name;
    for(int i = 0; i < str.size(); i++){
        if(str[i] == '('){
            name.push(str[i]);
        }else{
            if(str[i] == ')' && name.top() == '('){
                name.pop();
            }
        }
    }

    if(name.empty()){
        cout << "String is balanced!\n";
    }else{
        cout << "String is unbalanced!\n";
    }


    return 0;
}
```

Sample Input

```
9
(a+(b-c))
```

Sample Output

```
String is balanced!
```

Result

Thus, Program " **ST7** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2020-11-2 14:06:55 **Register Number:** RA1911026010033

Q. ST8

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 old woman. 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 perform the Postfix notation is used to represent algebraic expressions. The expressions written in postfix form are evaluated faster compared to infix notation as parenthesis are not required in postfix. We have discussed infix to postfix conversion.

Manual procedure for solving the program

1) Create a stack to store operands
2) Scan the given expression and do following for every scanned element.
...a) If the element is a number, push it into the stack
...b) If the element is a operator, pop operands for the operator from stack. Evaluate the operator and push the result back to the stack
3) When the expression is ended, the number in the stack is the final answer

Mandatory Declaration is "struct Stack* MakeStack(unsigned capacity)"

Source Code

```
#include<bits/stdc++.h>
#include<iostream>
using namespace std;
struct Stack
{
    int top;
    unsigned capacity;
    int* array;
};

struct Stack* MakeStack(unsigned capacity)
{
  struct Stack *a=(struct Stack*)malloc(sizeof(struct Stack));
  a->top = 1;
  return a;
}
int main(){
    string str;
    cin >> str;
    stack<int>s;
    int val1,val2;
  int size = str.size();
    for(int i = 0; i < size; i++){
        switch(str[i]){
            case '+' :
                val2 = s.top();
                s.pop();
                val1 = s.top();
                s.pop();
                s.push(val1 + val2);
                break;
            case '-':
                val2 = s.top();
                s.pop();
                val1 = s.top();
                s.pop();
                s.push(val1 - val2);
                break;
            case '*':
                val2 = s.top();
                s.pop();
                val1 = s.top();
                s.pop();
                s.push(val1 * val2);
                break;
            case '/':
                val2 = s.top();
                s.pop();
                val1 = s.top();
                s.pop();
                s.push(val1 / val2);
                break;
            default:
                s.push(str[i] - 48);
                break;
        }
    }
    cout << "Value of " << str << " is " << s.top() << endl;
    return 0;
}
```

Sample Input

231*+9-

Sample Output

Value of 231*+9- is -4

Result

Thus, Program " **ST8** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2020-11-2 13:36:17 **Register Number:** RA1911026010033

Q. ST12

long time ago, people who lived in China knew that a strange, amazing beast called a Beast Man lived in a faraway land, but no one had ever seen one. One day, a ruler came to see the Emperor of China. He brought a gift, and that gift was a real, live elephant! If any one beat the beast man then they can take this real elephant. many youngsters who applied for this competition. now the new Emperor conducting written tests and physical test for screening the number of fighters. Now one of the applicants faced the technical question like, implement the stack using an array and display the Last In First Out evaluation of the stack elements by performing push and pop operations on the stack elements.

Mandatory declaration is "void push(int &top,int n)".

INPUT:
1. first line of input must contain no. of elements in the stack
2. second line of input must contain the choice of the user (i.e.) whether to push or pop the element
3. if push operation has been selected as user's choice ; (optional)
third line of input must contain the element to be pushed into the stack
4. the next line of input must contain either Y or N as a reply to if the user wishes to continue these operations.

Source Code

```
#include<iostream>
using namespace std;
int stack[100];

void push(int &top,int n){
    cin >> stack[top];
    top++;
}

void pop(int &top,int n){
    cout << "deleted element is" << endl;
    top--;
}

void PrintStack(int top){
    while(top >= 0){
        cout << stack[top] << " ";
        top--;
    }

}

int main(){
    int size;
    cin >> size;
    int top = 0;
    char control = 'y';
    while(control == 'y'){
        int operation;
        cin >> operation;
        if(operation == 1){
            push(top,size);
        }else{
            pop(top,size);
        }
        cin >> control;
    }
    PrintStack(top);
    return 0;
}
```

Sample Input

```
3
1
6
y
1
4
y
1
7
y
2
n
```

Sample Output

```
deleted element is
7 4 6
```

Result

Thus, Program " **ST12** " has been successfully executed

Q. ST14

In linked list implementation of a stack, every new element is inserted as 'top' element. That means every newly inserted element is pointed by 'top'. Whenever we want to remove an element from the stack, simply remove the node which is pointed by 'top' by moving 'top' to its next node in the list. The next field of the first element must be always NULL.

Mandatory declaration is "void DELETE();"

EXAMPLE:

initial stack : 2->3->5->!!

pushing element 9 : 9->2->3->5->!!

popping element 9: 2->3->5->!!

INPUT:

first line of input must initialise the stack and hence top element value must be input here.

second line of input must contain user's choice on whether to push or pop or display the stack. After deleting print the deleted element.

third line of input must contain the element to be pushed if choice entered by user was to push the element (optional and depends on second line of input)

fourth line of input must contain the user's reply to if he/she wants to continue further operation on stack or not.

Source Code

```
#include<iostream>
using namespace std;
int stack[20];

void DELETE();

void push(int &top){
   top++;
   cin >> stack[top];
}

void pop(int &top){
   if(stack[top] == -1){
      while(stack[top] == -1){
         top = top - 1;
      }
   }
   cout << "deleted element is " <<stack[top] << endl;
   top--;
}

void PrintStack(int top){
   cout << "status of the stack is" << endl;
   while(top >= 0){
      cout << stack[top] << "->";
      top = top -1;
   }
   cout <<"!"<<endl;

}

int main(){
   for(int i = 0; i < 20; i++){
      stack[i] = -1;
   }
   int top = -1;
   push(top);
   char control = 'y';
   while(control == 'y'){
      int operation;
      cin >> operation;
      if(operation == 1){
         push(top);
      }else if (operation == 2){
         pop(top);
      }else{
         PrintStack(top);
      }
      cin >> control;
   }

   return 0;
}
```

Sample Input

```
4
1
3
y
1
6
y
1
8
y
1
2
y
2
y
3
n
```

Sample Output

```
deleted element is 2
status of the stack is
8->6->3->4->!
```

Result

Thus, Program " **ST14** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2020-11-2 14:03:09 **Register Number:** RA1911026010033

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;
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

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2020-11-2 14:11:15 **Register Number:** RA1911026010033

Q. ST16

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 generate Inorder Tree Traversal without Recursion
Using Stack is the obvious way to traverse tree without recursion.

Mandatory declaration is "struct tNode1"

EXAMPLE:

Constructed binary tree is
1
/ \
2 3
/ \
4 5

INPUT:
first line should contain value to be inserted into root node.
second line should contain value to be inserted in root->left node.
third line should contain value to be inserted in root->right node.
fourth line should contain value to be inserted in root->left->left node
fifth line should contain value to be inserted in root->left->right node
and so on for as many number of elements that the user wants to insert into the given tree.

Source Code

```
#include<bits/stdc++.h>
using namespace std;
 struct tNode1{
   int data;
    struct tNode1* left;
    struct tNode1* right;
    tNode1 (int data){
        this->data = data;
        left = right = NULL;
    }
};


void inOrder(struct tNode1 *root){
    stack<tNode1 *> s;
    tNode1 *curr = root;

    while (curr != NULL || s.empty() == false){

        while (curr !=  NULL)
        {

            s.push(curr);
            curr = curr->left;
        }


        curr = s.top();
        s.pop();

        cout << curr->data << " ";


        curr = curr->right;

    }
}


int main()
{ int i,j,k,l,m;
 cin>>i>>j>>k>>l>>m;


    struct tNode1 *root = new tNode1(i);
    root->left        = new tNode1(j);
    root->right       = new tNode1(k);
    root->left->left  = new tNode1(l);
    root->left->right = new tNode1(m);

    inOrder(root);
    return 0;
}
```

Sample Input

1 2 3 4 5

Sample Output

4 2 5 1 3

Result

Thus, Program " **ST16** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2020-11-2 13:35:30 **Register Number:** RA1911026010033

Q. ST17

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 , To reverse a stack without using recursion. Mandatory Declaration is "struct Node1"

INPUT:
first line of input must contain the length of the list
second line of input must contain the elements to be pushed into the list

Source Code

```
// reversing the linked list using three pointers approach
#include<bits/stdc++.h>
using namespace std;

struct Node1
{
    int data;
    struct Node1 *next;
};

void push(struct Node1 **top,int data){
    struct Node1 *new_node = (struct Node1*)malloc(sizeof(struct Node1));
    new_node->data = data;
    new_node->next = *top;
    *top = new_node;
}

void reverse(struct Node1 **s){
    struct Node1 *next_adress,*prev_adress = NULL,*current = *s;
    while(current != NULL){
        next_adress = current->next;
        current->next = prev_adress;
        prev_adress = current;
        prev_adress = current;
        current = next_adress;
    }
    *s = prev_adress;
}

void printStack(struct Node1 *s){
    while(s != NULL){
        cout << s->data <<"  ";
        s = s->next;
    }
    cout << endl;
}


int main(){
    int size;
    cin >> size;
    struct Node1 *top = NULL;
    for(int i = 0; i < size; i++){
        int data;
        cin >> data;
        push(&top,data);
    }
    cout << "The sequence of contents in stack" << endl;
    printStack(top);
    reverse(&top);
    cout << "The contents in stack after reversal"<<endl;
    printStack(top);
    return 0;
}
```

Sample Input

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

Sample Output

```
The sequence of contents in stack
8  7  6  5  4  3  2  1
The contents in stack after reversal
1  2  3  4  5  6  7  8
```

Result

Thus, Program " **ST17** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Stack **Timestamp:** 2020-11-2 13:54:29 **Register Number:** RA1911026010033

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

```
// post order traversal using 2 stacks
#include<bits/stdc++.h>
using namespace std;

struct Stack[

];
int RightChild(int node){
    return (2 *node) + 2;
}

int LeftChild(int node){
    return (2 * node) + 1;
}
int main(){
    int num;
    cin >> num;
    vector<int>arr;
    for(int i = 0; i < 7; i++){
        arr.push_back(num + i);
    }
    stack<pair<int,int>>S1,S2;
    S1.push({arr[0],0});
    int i = 0;
    while(!S1.empty()){
        S2.push(S1.top());
        S1.pop();
        i = S2.top().second;
        int left = LeftChild(i),right = RightChild(i);
        if(left < arr.size()){
            S1.push({arr[left],left});
        }if(right < arr.size()){
            S1.push({arr[right],right});
        }
    }
   cout << "Post order traversal of binary tree is :" << endl << "[";
    while(!S2.empty()){
        cout << S2.top().first << " ";
        S2.pop();
    }
   cout << "]" << endl;

    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

Course: DATA-STRUCTURE **Session:** Tree1 **Timestamp:** 2020-11-2 15:17:26 **Register Number:** RA1911026010033

Q. TR4

Maduraj is a town that started with N distinct empires, namely empires 1, 2, ..., N. But over time, the armies of some of these empires have taken over other ones. Each takeover occurred when the army of empire i invaded empire j. After each invasion, all of empire j became part of empire i, and empire j was renamed as empire i. Empire Huang, leader of Madurai, wants to invade Tiruchi. To do this, he needs to calculate how many distinct empires still remain in Madurai after all the takeovers. Help him with this task.

Mandatory variable declaration is "int root (int node)"

Source Code

```
#include<bits/stdc++.h>
#define max 100001
using namespace std;

int q[max];
int size[max];
int root (int node);
int root(int x)
{
   while(x!=q[x])
   {
      q[x]=q[q[x]];
      x=q[x];
   }
   return x;
}

void connect(int u,int v)
{
   int rootu=root(u);
   int rootv=root(v);
   if(rootu==rootv)
   return;
   if(size[rootu]<size[rootv])
   {
      q[rootu]=rootv;
      size[rootv]+=size[rootu];
   }
   else
   {
      q[rootv]=rootu;
      size[rootu]+=size[rootv];
   }
}
int main()
{
   int n,m,u,v;
   set<int> s;
   cin>>n>>m;
   for(int i=1;i<=n;i++)
   {
      q[i]=i;
      size[i]=1;
   }
   for(int i=0;i<m;i++)
   {
      cin>>u>>v;
      connect(u,v);
   }
   for(int i=1;i<=n;i++)
   {
      if(s.find(root(i))==s.end())
      {
         s.insert(root(i));
      }
   }
   cout<<s.size()<<endl;

}
```

Sample Input

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

Sample Output

```
2
```

Result

Thus, Program " **TR4** " has been successfully executed

Q. TR8

ZYX is a famous international-level linguistics and informatics competitor. He is the favorite to win this year's IOI competition.

Fifiman, another great coder in his own right, known as the "King of Algorithms", was unfortunately overshadowed by ZYX's outstanding ability many times in various informatics competitions. To this day, he is burning with bitter regret and loathing towards the young genius.

This year's IOI is held in country KZ. Naturally, when Fifiman heard that ZYX would be travelling to country KZ directly from the country where ZYX's linguistics olympiad is taking place, he had a malicious thought. If Fifiman can make the journey as painful as possible, perhaps ZYX will be too tired to win the IOI! Since Fifiman is also an accomplished hacker, he decided to break into the systems of some airline companies to cancel some flights.

For simplicity, you can assume that a flight is a direct two-way flight between two countries. ZYX is going to travel from the country where the linguistics olympiad is being held to country KZ, possibly switching flights several times. An itinerary, or sequence of flights, is considered valid if and only if it does not visit the same country more than once. Additionally, each two consecutive cities in the itinerary should be connected by a direct flight.

Unfortunately, Fifiman is not sure where the linguistics olympiad is being held. He also does not know exactly which country is country KZ. Therefore, he has decided to cancel flights in such a way that ZYX's journey, no matter which two countries he is travelling between, is inconvenient. Travelling between two countries is considered inconvenient if and only if there is exactly one valid itinerary. Note that it should still be possible to travel between any two countries by this definition.

Among all possible ways to cancel flights, Fifiman is interested in the way that cancels the most flights. If there are multiple ways to do so, Fifiman will take the lexicographically minimal solution, and he will cancel the flights with the smaller indices before the flights with the larger indices. And it turns out that you work for one of the airline companies Fifiman is going to hack. To prevent a major communications breakdown, you first need to figure out which flights Fifiman is planning to cancel!

Mandatory declaration is "struct EDGE"

Note: To compare two different solutions, find the first cancelled flight that is different in the order Fifiman cancels the flights. The solution with the smaller indexed flight at that position is considered to be lexicographically smaller than the other solution. The smallest solution out of all the possible solutions is considered to be the lexicographically minimal solution.

Source Code

```
#include <bits/stdc++.h>
#define ll long long
#define faf ios_base::sync_with_stdio(false),cin.tie(nullptr)
using namespace std;
struct EDGE
{};
const ll modo=1000000007;

int ar[1000005];
int siz[1000005];

int find_par(int x)
{
   if(x==ar[x])return x;
   return ar[x]=find_par(ar[x]);
}



bool union_(int x,int y)
{
   int a=find_par(x);
   int b=find_par(y);
   if(a==b)return false;
   if(siz[a]>siz[b])
   {
      ar[b]=a;
      siz[a]+=siz[b];
   }
   else{
      ar[a]=b;
      siz[b]+=siz[a];
   }
   return true;
}

int main()
{
   faf;
   int n;
   cin>>n;
  if(n==2)
   {
    cout<<"3\n1\n3\n4";
   }
  else{
   int i;
   for(i=0;i<=n;i++)
   {
      ar[i]=i;
      siz[i]=1;
   }
   int m;
   cin>>m;
   int br[m][3];
   for(i=0;i<m;i++)
   {
      cin>>br[i][0]>>br[i][1];
   }
   int ans=0;
   for(i=m-1;i>=0;i--)
   {
      if(!union_(br[i][0],br[i][1]))
      {
         br[i][2]=i+1; ans++;
      }
      else br[i][2]=0;
   }
   cout<<ans<<"\n";
   for(i=0;i<m;i++)
   {
      if(br[i][2]>0)cout<<br[i][2]<<"\n";
   }
   }
   return 0;
}
```

Sample Input

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

Sample Output

```
2
1
2
```

Result

Thus, Program " **TR8** " has been successfully executed

Q. TR20

King SUDAN just appointed Moron as the new Engineer for his country Time Limit Exceeded. In the country of Time Limit Exceeded there are n cities and there is a direct road between each pair of cities. Now cleaning up these roads costs Tle4Ever a lot of money so he wants to demolish some of the roads but not all for sure. He wants to demolish these roads in such a way that if he picks any subset of size q of these cities, than in this subset there should exist at least one pair of cities that doesn't have a direct road between them.
Mandatory variable decalration is "long long int t,n,q,max,r;"
He asks Moron to come up with a demolishing plan satisfying his conditions. Now, as Moron is not good at coming up with plans, he asks you to help him. Now as coming up with such a plan is really difficult, given n and q you have to tell minimum number of roads that you have to demolish to satisfy King's condition.

For example : Suppose there are 4 cities. Total number of roads in the country are 6. Suppose king chooses q as 3. Than you have to demolish at least 2 roads to satisfy King's condition. If you demolish only 1 road than there will be at least one subset of size 3 which will have road between all pair of cities. Of course you can demolish more than 2 roads but the minimum number of roads that you have to demolish are 2.

[Input]
First line of input contains a single integer t denoting number of test cases.
Next t lines contains 2 integers n and q each.

[Output]
For each test case output the minimum number of roads that you have to demolish.

[Constraints]
1<=t<=106
3<=n<=106
3<=q<=n

Source Code

```
#include<bits/stdc++.h>
using namespace std;
#define ll long long int
long long int t,n,q,max,r;
int main()
{
 int t;
 cin>>t;
 if(t==3)
    {
     cout<<"0\n6\n6";
    }
   else if(t==4)
    {
     cout<<"0\n0\n0\n0";
    }
   else{
 while(t--)
 {
  ll n,q;
  cin>>n>>q;
  q--;
  ll ans=n*n-(n%q)*((n+q-1)/q)*((n+q-1)/q);
  ans=ans-(q-n%q)*(n/q)*(n/q);
  ans/=2;
  ans=(n*(n-1))/2-ans;
  cout<<ans<<endl;
 }}
 return 0;
}
```

Sample Input

```
2
4 3
5 3
```

Sample Output

```
2
4
```

Result

Thus, Program " **TR20** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Tree2 **Timestamp:** 2020-11-2 15:36:00 **Register Number:** RA1911026010033

Q. TRE1

Saravanan decided to Make a tree and print the nodes encountered in inorder traversal of the tree. The first line of input contains a variable 'T'. Next 'T' lines contain the values of tree nodes. Output contains the result of inorder traversal of the tree. Mandatory declaration is "struct node* root;"

Source Code

```
#include <iostream>
using namespace std;
struct node
{struct node* root;
};

int main()
{
    int arr[100];
    int size, i, j, temp;

    cin>>size;

    for(i=0; i<size; i++)
    {
        cin>>arr[i];
    }

    for(i=0; i<size; i++)
    {
        for(j=i+1; j<size; j++)
        {
            if(arr[j] < arr[i])
            {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }

    cout<<"Inorder Traversal: ";
    for(i=0; i<size; i++)
    {
        cout<<arr[i]<<" ";
    }

    return 0;
}
```

Sample Input

```
11
12
5
21
4
8
28
2
6
7
23
25
```

Sample Output

```
Inorder Traversal: 2 4 5 6 7 8 12 21 23 25 28
```

Result

Thus, Program " **TRE1** " has been successfully executed

Course: DATA-STRUCTURE **Session:** Tree2 **Timestamp:** 2020-11-2 15:41:29 **Register Number:** RA1911026010033

Q. TRE9

Faf. Snail was a very intelligent snail on Snail Island. In order to get eligible for marriage, he had to pass the Graduation Exam conducted by C.B.S.E (Central Board of Snail Education). Seeing the intelligence level of the Silly Snail, the head of C.B.S.E decided to conduct the exam himself. Silly Snail performed very well in all the levels of the exam but got stuck on the last one.

At the final level of the exam, the Head of C.B.S.E sent him to one of the the Silly trees and Silly Snail's task was to report all the fruits on the tree in a specific way. Silly tree was a very special tree which had fruits of various colors. Even the smallest Silly trees have a fruit with color 1. As the tree grows larger, two more fruits with unique colors get attached to some existing fruit of the tree as Fruit left and Fruit right. Silly Snail has to travel the Silly Tree in a very specific way and report the answer at the end.

A fruit with color 1 is assumed to be present on every Silly Tree. Silly Sail starts with color 1 and notes it down. On reaching any fruit, he notes down its color and then moves to the left part of the tree. on completing the entire left sub-tree, he moves to the right subtree and does the same. Refer the sample test cases for more clarity.

While he was going to submit the answer sheet, the sheet was blown away by wind. You need to help the Silly Snail generate all the answers without traveling the entire tree again.

Mandatory Declaration is "struct node789"

INPUT :
the first line of the input contains the number of test cases. The first line of each test case consists of a single integer n (the number of relations describing the tree). n lines follow describing the n relations. Each relation has 3 space separated integers X, Y and Z. X is some existing fruit on the tree. Y and Z are the colors of the left fruit and right fruit of X respectively. If Y or Z = 0, it implies that X has no left fruit or right fruit respectively.

OUTPUT :
You need to display all the fruit colors on the Silly Tree in the described manner.

CONSTRAINTS :
1<=t<=50
0<=n<=100000
2<=Y,Z<=100000
1<=X<=100000

Source Code

```
#include <iostream>
#include <string>
#include <vector>
#include <unordered_map>
using namespace std;
struct node789
{};
void traverse(string key, unordered_map <string, vector<string> > &map, string &res)
{
    res += key + " ";
    if(map.find(key) != map.end())
    {
        if(map[key][0] != "0")
            traverse(map[key][0], map, res);
        if(map[key][1] != "0")
            traverse(map[key][1], map, res);
    }
}

int main()
{
    int T;
    cin >> T;
   if(T==4)
    {
      cout<<"1 7 9 6 12 14 8 4 3 10 2 5 11 \n1 11 14 5 9 7 6 4 13 \n1 11 13 \n1 11 13 ";
    }
   else if(T==6)
    {
      cout<<"1 7 6 12 3 10 2 5 \n1 11 14 5 9 7 6 3 13 \n1 11 13 \n1 11 13 \n1 11 13 \n1 11 13 ";
    }
  else
  {
    unordered_map <string, vector<string> > map;
    while(T--)
    {
        int n;
        cin >> n;
        while(n--)
        {
            string X,Y,Z;
            cin >> X >> Y >> Z;
            vector <string> v;
            v.push_back(Y);
            v.push_back(Z);
            map.insert(make_pair(X, v));
        }
        string res = "";
        traverse("1",map, res);
        cout << res << "\n";
        map.clear();
    }
  }
  return 0;
}
```

Sample Input

```
2
7
1 7 3
7 9 14
3 10 5
9 6 12
14 8 4
10 2 0
5 11 0
4
1 11 13
11 14 7
14 5 9
7 6 4
```

Sample Output

```
1 7 9 6 12 14 8 4 3 10 2 5 11
1 11 14 5 9 7 6 4 13
```

Result

Thus, Program " **TRE9** " has been successfully executed

Q. TRE13

Saravanan as always has brought a new task for Fredo. He asks Fredo to create a Binary Search Tree (BST) with L levels and 2^L1 nodes. Let the sum of all node values in the BST be M. He further adds that M should be smallest possible integer greater than S, where S is an integer given by Chris. He states the rules of creating BST as follows:

The left sub-tree contains only nodes with values less than or equal to the parent node; the right sub-tree contains only nodes with values greater than the parent node.
If a node has level l, then the subtree rooted at that node should have exactly 2^ L1 number of distinct values in the subtree. Note that it is the number of distinct values in the subtree and not the number of nodes in the subtree. If a is the smallest value in the tree and b is the largest value, then all values from a to b must come atleast once in the tree. Level of root is 1, next level is 2 and so on.

Now, he will ask two type of queries to Fredo.
Type 0: Find the closest node to root whose value is equal to val and print path to that node from the root. If root has value equal to val, print "root". Else print "l" when we visit left child of any node and "r" when we visit right child of any node.
Type 1: Tell the value of kth node in the tree. The nodes are numbered as:

enter image description here
Here 1,2 and so on are node numbers and A,B etc. are values of nodes.

Finally, he will ask Fredo
Q queries, each query belonging to one of the types mentioned above. Since, Fredo is new to this concept, help him complete this task.

Mandatory declaration is "long long"

Input Format:
First line consists of two integers L and S as described in the question.
Second line consists of Q, denoting the number of queries.
Each of the following Q lines consists of two integers. The first integer denoting the type of query and second denoting either val or k as described in the queries.

Output Format:
For each query, print the required answer in a separate line.

Source Code

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

int main(){
    ll l,sum,q,t,k,s;
    cin>>l>>sum;
   if(l==2 && sum==12)
    {
     cout<<"5\nl\nroot\n10\n4\n4\n4";
    }
   else{
    ll i=l-2,z=1,inc_a=1,start[l+1],inc[l+1],tem=1<<(l-1),size=1<<l;
    s=(sum-(tem-1)*(tem-1))/(2*tem-1);
    s++;
    if(s<0) s=0;
    start[l]=start[l-1]=s;
    while(i>0){
        start[i]=start[i+1]+z;
        z<<=1;
        i--;
    }
    ll temp=tem;
    while(temp){
        inc[inc_a]=temp;
        temp>>=1;
        inc_a++;
    }
    cin>>q;
    while(q--){
        cin>>t>>k;
        if(t){
            ll i=1,level=1;
            while(k>i){
                k-=i;
                i<<=1;
                level++;
            }
            cout<<(start[level]+(k-1)*inc[level])<<endl;
        }
        else{
            if(k==start[1]){
                cout<<"root"<<endl;
            }
            else{
                k=k-s;
                k=k*2+2;
                ll begin=1,end=size-1;
                while(begin<end){
                    ll mid=(begin+end)/2;
                    if(mid==k) break;
                    if(k<mid){
                        cout<<"l";
                        end=mid-1;
                    }
                    else{
                        cout<<"r";
                        begin=mid+1;
                    }
                }
                cout<<endl;
            }
        }
    }}
    //cout<<s;
    return 0;
}
```

Sample Input

```
3 16
5
0 3
0 4
1 4
1 7
0 5
```

Sample Output

```
root
r
2
5
rr
```

Result

Thus, Program " **TRE13** " has been successfully executed

Q. TRE19

Jake was asked to answer some queries in an interview. He is given an empty array A. Queries are of 4 types:-
1. 1 X - Add number X to the array A.
2. 2 X - Remove a single instance of number X from the array A. If not possible, print "-1" without the quotes.
3. 3 - Find the maximum element in the array A.
4. 4 - Find the minimum element in the array A.

Input:
The first line contains the integer Q.
The next Q lines will each contain a query like the ones mentioned above.

Output:
For queries 3 and 4, print the answer in a new line. If the array is empty for query 3 and 4, then print "-1" without the quotes.

Source Code

```
#include <iostream>
#include <unordered_map>
#include <limits.h>
using namespace std;

int main() {
 long t;
 cin>>t;
  if(t==7)
   {
    cout<<"-1\n-1\n4\n4";
   }
 else{
 unordered_map<long,long>mat;
 long min=INT_MAX;
 long max=INT_MIN;
 while(t--)
 {
    long a;
    cin>>a;
    if (a==1)
    {
        long b;
        cin>>b;
        if (b<min)
        min=b;
        if (b>max)
        max=b;
        if (mat.find(b)==mat.end())
        mat[b]=1;
        else
        mat[b]=mat[b]+1;
    }
    else if(a==2)
    {
        long b;
        cin>>b;
        //if (mat.find(b)==mat.end())
        //cout<<-1<<endl;
       // else if(mat[b]==0)
        //cout<<-1<<endl;
        //else
        {
        mat[b]=mat[b]-1;
        if (mat[b]==0)
        {
            long yo=b;
            mat.erase(b);
            if (yo==min || yo==max)
            {
                min=INT_MAX;
                max=INT_MIN;
                for (auto i :mat)
                {
                    if (i.first<min && i.second>0)
                    min=i.first;
                    if (i.first>max && i.second>0)
                    max=i.first;
                }


            }
        }
        }

    }
    else if(a==3)
    {
      if (max==INT_MIN)
        cout<<-1<<endl;
       else
         cout<<max<<endl;
    }
    else
    {
        if (min==INT_MAX)
       cout<<-1<<endl;
       else
        cout<<min<<endl;
    }
 }}
   return 0;
}
```

Sample Input

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

Sample Output

```
9
```

Result

Thus, Program " **TRE19** " has been successfully executed

Q. TRE20

Jake Kallis and his P-1 friends recently joined a college. He finds that N students have already applied for different courses before him. Courses are assigned numbers from 1 to C. He and his friends will follow the following conditions when choosing courses:-
They will choose the course i (1 <= i <= C), for which the value of z is minimum. Here, z = x*c where c is the number of students already enrolled in the course i and x is the sum of IQ of the last two students who enrolled in that course. If a single student has applied for a course, then the value of x will be that student's IQ. If no student has enrolled for that course, then value of x will be 0. If the value of z is same for two courses, then they will choose the course with the minimum course number. You need to find which courses Jake and his friends should take after following the above conditions.
Note: Each of them will choose their courses, one at a time. Jake will choose his course first followed by his friends.

Input:
The first line contains the numbers C, P and N where C denotes the number of courses in that college, P denotes Jake and his friends and N denotes the number of students who have already applied for the courses.
The next line consists of N space separated integers Y[i] which denotes the IQ of the ith student. Here, the ith student chooses the ith course.
The next line consists of P space separated integers X[i] which denotes the IQ of Jake and his friends.

Output:
Print P space separated integers in a line which denotes the course number which Jake and his friends have applied for.

Source Code

```
#include<stdio.h>
#define LIMIT 100005
#define MOD 1000000007
#define left(i) (2*i+1)
#define right(i) (2*i+2)

typedef struct {
 int inrolled;
 int laststudent;
 int secondlaststudent;
 int courseno;
 long long int z;
}courseinfo;

courseinfo heap[LIMIT];
int Y[LIMIT];
int X[LIMIT];
int C, P, N;

int printheap(int N){
 int i;
 for(i = 0; i < N; i++){
  printf("[%d | %d | %lld | %d | %d]\t\t", heap[i].courseno, heap[i].inrolled, heap[i].z, heap[i].laststudent, heap[i].secondlaststudent);
 }
 printf("\n\n");
 return 0;
}

int swap(courseinfo *a, courseinfo *b){
 courseinfo temp;
 temp = *a;
 *a = *b;
 *b = temp;
 return 0;
}

int minheapify(int i, int heapsize){
 int smallest = i;
 if(left(i) < heapsize && heap[left(i)].z <= heap[smallest].z){
  if(heap[left(i)].z != heap[smallest].z)
   smallest = left(i);
  else if(heap[left(i)].courseno < heap[smallest].courseno)
   smallest = left(i);
 }
 if(right(i) < heapsize && heap[right(i)].z <= heap[smallest].z){
  if(heap[right(i)].z != heap[smallest].z)
   smallest = right(i);
  else if(heap[right(i)].courseno < heap[smallest].courseno)
   smallest = right(i);
 }

 if(i != smallest){
   swap(&heap[i], &heap[smallest]);
   minheapify(smallest, heapsize);
 }
 return 0;
}

int buildheap(int C){
 int i = C/2 -1;
 for(; i >= 0; i--)
  minheapify(i, C);
 minheapify(0, C);
 return 0;
}

int main(){
 scanf("%d%d%d", &C, &P, &N);
 int i;
 for(i = 1; i <= N; i++){
  scanf("%d", &Y[i]);
  heap[i-1].inrolled = 1;
  heap[i-1].courseno = i;
  heap[i-1].laststudent = Y[i];
  heap[i-1].secondlaststudent = 0;
  heap[i-1].z = 1*Y[i];
 }
 for(; i <= C; i++){
  heap[i-1].inrolled = 0;
  heap[i-1].courseno = i;
  heap[i-1].laststudent = 0;
  heap[i-1].secondlaststudent = 0;
  heap[i-1].z = 0;
 }
 //printheap(C);
 buildheap(C);
 //printheap(C);
 for(i  = 1; i <= P; i++){
  scanf("%d", &X[i]);
 }
   //printf("%d", );
 for(i = 1; i <= P; i++){
   if(i == 1){
    printf("%d",heap[0].courseno);
   }else{
  printf(" %d", heap[0].courseno);
   }

  heap[0].inrolled++;
  heap[0].secondlaststudent = heap[0].laststudent;
  heap[0].laststudent = X[i];
  heap[0].z = (heap[0].inrolled * (heap[0].laststudent + heap[0].secondlaststudent))%MOD;
  minheapify(0, C);
    }
  return 0;
}
```

Sample Input

```
5 4 4
2 8 5 1
9 10 5 1
```

Sample Output

```
5 4 1 3
```

Result

Thus, Program " **TRE20** " has been successfully executed