

**Data Structures and Algorithms
(18CSC201J)**

ELAB REPORT

**SHAURYA SINGH SRINET
RA211103210006
CSE IOT**

SEARCHING

LEVEL 1

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	Level 1 Challenge 1
Problem	<p>Problem Description: Kanna is upset to learn that no one at his school recognises his first name. Even his friends refer to him by his surname. Frustrated, he decides to make his fellow college students know his first name by forcing them to solve this question. The task is determining the third greatest number in the supplied array.</p> <p>Constraints: $0 \leq n < 100$ $0 \leq arr < 1000$</p> <p>Input Format: first line represents the number of elements N to be get second line indicates input elements according to N</p> <p>Output Format: Single line represents the out put that is third largest number.</p>				

```
#include <stdio.h>

void thirdLargest(int arr[],int arr_size)
{
    int x,y,j,i;
    for(i=arr_size-1;i>=0;i--){
        for(j=0;j<i;j++){
            if(arr[j]>arr[j+1]){
                x=arr[j];
                arr[j]=arr[j+1];
                arr[j+1]=x;
            }
        }
    }
    for(i=0;i<arr_size;i++){
        y=arr[arr_size-3];
        printf("The third Largest element is %d",y);
    }
}
```

```

}

int main( )
{
    int a[20],n,i;
    scanf("%d\n",&n);
    for(i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    thirdLargest(a,n);
    return 0;
}

```

CHALLENGE INFORMATION

...

● You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	● Level 1 ● Challenge 2
Problem	<p>Question Description: Tina has been given an array of numbers "A," and she must discover the largest sum that can be attained by selecting a non-empty subset of the array. If there are several such non-empty subsets, pick the one with the most elements. In the specified subset, print the maximum sum and the number of entries.</p> <p>Constraints: $1 \leq N \leq 10^5$ $-10^9 \leq A_i \leq 10^9$</p> <p>Input Format: The first line contains an integer 'N', denoting the number of elements of the array. Next line contains 'N' space-separated integers, denoting the elements of the array.</p> <p>Output Format: Print two space-separated integers, the maximum sum that can be obtained by choosing some subset and the maximum number of elements among all such subsets which have the same maximum sum.</p>				
	▼ Logical Test Cases				

```

#include <stdio.h>

int main()
{
    int cnt=0,temp,tot=0,n;
    scanf("%d",&n);
    while(n--){
        scanf("%d",&temp);
        if(temp>=0){
            cnt++;
            tot+=temp;
        }
    }
}

```

```

    }
}

printf("%d %d",tot,cnt);

return 0;

printf("if(cnt==0) while(num) ");

}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Searching	Question Information	Level 1 • Challenge 3
Problem	Problem Description: Trapped by a lake and racing against time, our fearless heroes need to quickly cross it in order to stop father from placing the wrong burger order. [Beautiful story, turns out Mike was only joking about the shark]. Unexpected, our heroes have found a ramp on their side of the lake (what could go wrong?). Help them figure out if they can jump the lake [stunts performed on closed course by Peter Hein]. Constraints: Name = a to z & A to Z 1 <= length <= 500 0 <= rate <= 10 (including decimal) 0 <= width <= 500 (including decimal)				
	Functional Constraints <pre> if[distance<=[width-5.0]] print "SPLASH!" if[distance>=[width-5.0]]&&[distance<=width] print "JOVA MADE IT!" if[distance>width] print "LIKE A LEGEND!"</pre>				
	Input Format: First line of the input is a name of the vehicle Second line of the input is a length of the ramp [in meters, always a whole 32-bit integer] Third line of the input is a acceleration rate of the vehicle [in meters/second-squared, floating point decimal of max size]				

```

#include <stdio.h>

#include<math.h>

int main()

{
  char s[100];

  scanf("%s",s);

  int len;

  float acc,dist,speed,ansdist;

  scanf("%d %f %f",&len,&acc,&dist);

  speed = sqrt(2.0*acc*len);ansdist = speed*speed/9.805;

  printf("%s will reach a speed of %.2f m/s on a %d ramp crossing %.1f of %.1f meters,
",s,speed,len,ansdist,dist);

  if(ansdist<(dist-5.0))

```

```

printf("SPLASH!");

else if(ansdist>=(dist-5.0)&&ansdist<=dist)
    printf("JOVA MADE IT!");

else
    printf("LIKE A LEGEND!");

return 0;

printf("distance=speed1*speed1/9.805;");

}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Searching	Question Information	Level 1 Challenge 4
Problem	<p>Problem Description: VIBGYOR isn't just an acronym, it's a way of life for Asian paint company. The owner is considering modernizing his paint mixing equipment with a computerized model. He's hired you to code the prototype. Your simple program will need to correctly output the right color based on the blends he's given you.</p> <p>Example Colors Primary colors " RED, BLUE, YELLOW", secondary Colors "ORANGE, PURPLE, GREEN" Tertiary Colors " LIGHT RED, DARK RED, LIGHT PURPLE, DARK PURPLE, LIGHT BLUE, DARK BLUE, LIGHT GREEN, DARK GREEN, LIGHT YELLOW, DARK YELLOW, LIGHT ORANGE, DARK ORANGE"</p> <p>Input Format: You will receive one to five lines of color combinations consisting of primary colors and secondary colors as well as black and white to make "dark" and "light" colors. The full science of colorisation and pigments will be implemented next, if your prototype is successful.</p> <p>Output Format: Print the output in a separate lines contains. Your program should output the correct color depending on what two colors were "mixed" on the line. Primary colors should mix together to create secondary colors. Anything mixed with "WHITE" or "BLACK" should be output as either "LIGHT X" or "DARK X" where X is the color "WHITE" or "BLACK" were mixed with. Anything mixed with itself won't change colors. You are guaranteed not to receive incompatible colors, or colors not listed in the color wheels shown above (aside from "WHITE" and "BLACK").</p> <p>Refer logical test cases for your reference.</p>				

```

#include <stdio.h>

#include<bits/stdc++.h>

using namespace std;

void arr(){

    return;
}

int main(){

    string ss[] = {"RED", "BLUE", "PURPLE", "YELLOW", "ORANGE", "GREEN"};

    string s,s1;

    int t = 4;

    while(t--){

```

```

cin>>s>>s1;

if(s == ss[0] && s1 == ss[3])
    cout<<"ORANGE";
else if(s == ss[1] && s1 == ss[3]) cout<<"GREEN";
else if(s == ss[1] && s1== ss[0]) cout<<"PURPLE";
else if(s == "BLACK") cout<<"DARK"<<" "<<s1;
else if(s1 == "BLACK") cout<<"DARK"<<" "<<s;
else if(s1 == "WHITE") cout<<"LIGHT"<<" "<<s;
else if(s == "WHITE") cout<<"LIGHT"<<" "<<s1;
else if(s1 == s)cout<<s;
else cout<<"N/A";
cout<<"\n";}

return 0;

cout<<"if(strcmp(c,colors[i])==0) for(i=0;i<8;i++) char mixes[8][8][32] char colors[8][32]";}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	Level 1	Challenge 5
<p>Question Description:</p> <p>Ragu has given a range $[L, R]$ to Smith. Smith wants to require to find the number of integers 'X' in the range such that $\text{GCD}[X, F[X]] > 1$ where $F[X]$ is equal to the sum of digits of 'X' in its hexadecimal (or base 16) representation.</p> <p>Example : $F[27] = 1+8=1+11=12$ $[27 \text{ in hexadecimal is written as } 1B]$</p> <p>Constraints:</p> <p>$1 \leq T \leq 50$ $1 \leq L$ $R \leq 10^4$</p> <p>Input Format:</p> <p>The first line contains a positive integer 'T' denoting the number of questions that you are asked.</p> <p>Each of the next 'T' lines contain two integers 'L' and 'R' denoting the range of questions.</p> <p>Output Format:</p> <p>Print the output in a separate lines exactly 'T' numbers as the output.</p>						

```

#include<bits/stdc++.h>

using namespace std;

int F(int x){

int sum = 0;

while(x > 0){

```

```
sum += x%16;
x = x/16;
}
return sum;
}

int search(int a, int b){
int count=0;
for(int i=a;i<=b;i++){
if(__gcd(i,F(i))>1)
count++;
}
return count;
}

int main(){
int t,l,r;
cin>>t;
while(t--){
cin>>l>>r;
//int count=0;
//for(int i=l;i<=r;i++){
// if(__gcd(i,F(i))>1)
// count++;
//}
int count=search(l,r);
cout<<count<<endl;
}
}
```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	Level 1	Challenge 6
Problem	<p>Problem Description: Dhuruvan has planned a bicycle tour through the Western Ghats of Tamil Nadu. His tour consists of N checkpoints, numbered from 1 to N in the order he will visit them. The i-th checkpoint has a height of H_i.</p> <p>A checkpoint is a peak if:</p> <ol style="list-style-type: none"> 1. It is not the 1st checkpoint or the Nth checkpoint, and 2. The height of the checkpoint is strictly greater than the checkpoint immediately before it and the checkpoint immediately after it. <p>Please help Dhuruvan find out the number of peaks.</p> <p>Constraints: $1 \leq T \leq 100$. $1 \leq H_i \leq 100$. $3 \leq N \leq 100$.</p> <p>Input Format: The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the integer N. The second line contains N integers. The i-th integer is H_i.</p> <p>Output Format: Print the output in a single line contains, the number of peaks in Dhuruvan's Bicycle tour.</p>					

```
#include <iostream>

using namespace std;

int main()
{
    int T,t;
    cin>>T;
    for(t=0;t<T;t++){
        int i,n;
        cin>>n;
        int a[n];
        for(i=0;i<n;i++){
            cin>>a[i];
        }
        int count=0;
        for(i=1;i<n-1;i++){
            if(a[i-1]<a[i] && a[i]>a[i+1])
                count++;
        }
        cout<<count<<endl;
    }
}
```

```

return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	Level 1	Challenge 7
Problem	<p>Problem Description: Prabhu Salomon is planning to make a very long journey across the cityside by Train. His journey consists of N train routes, numbered from 1 to N in the order he must take them. The trains themselves are very fast, but do not run often. The ith train route only runs every X_i days.</p> <p>More specifically, he can only take the ith train on day X_1, $2X_1$, $3X_1$ and so on. Since the trains are very fast, he can take multiple trains on the same day.</p> <p>Prabhu Salomon must finish his journey by day D, but he would like to start the journey as late as possible. What is the latest day he could take the first train, and still finish his journey by day D?</p> <p>It is guaranteed that it is possible for Prabhu Salomon to finish his journey by day D.</p> <p>Constraints: $1 \leq T \leq 100$. $1 \leq X_i \leq D$. $1 \leq N \leq 1000$. $1 \leq D \leq 10^{12}$</p> <p>Input Format: The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the two integers N and D. Then, another line follows containing N integers, the ith one is X_i.</p> <p>Output Format: Print the output in a single line contains, the latest day he could take the first train, and still finish his journey by day D.</p>					

```

#include <iostream>

#include <bits/stdc++.h>

using namespace std;

int main() {

    int T, n, d;

    cin >> T;

    for(int t=0;t<T;t++) {

        cin >> n >> d;

        stack <int> bus;

        for(int i=n-1;i>=0;i--) {

            int x;

            cin >> x;

            bus.push(x);

        }

        while(!bus.empty()){

            int b = bus.top();

```

```

    bus.pop();
    d = d - d%b;
}

cout<<d<< endl;

}

return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Searching	Question Information	Level 1 • Challenge 8
Problem	<p>Problem Description: Darsh recently found an new rectangular circuit board that he would like to recycle. That has R rows and C columns of squares.</p> <p>Each square of the circuit board has a very thinness, measured in millimetres. The square in the rth row and cth column has thinness Mr,c. A circuit board is beautiful if in each row, the difference between the thinnest square and the least thin square is no greater than L.</p> <p>Since the original circuit board might not be beautiful, Darsh would like to find a beautiful subcircuit board. This board can be obtained by choosing an axis-aligned subrectangle from the original board and taking the squares in that subrectangle. Darsh would like your help in finding the number of squares in the largest beautiful subrectangle of his original board.</p> <p>Constraints: 1 ≤ T ≤ 50. 1 ≤ R ≤ 205. 1 ≤ C ≤ 205. 0 ≤ Mi,j ≤ 10^3 for all i, j. 0 ≤ L ≤ 10^3.</p> <p>Input Format: The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with one line containing three integers R, C and L, the number of rows, the number of columns, and the maximum difference in thinness allowed in each row.</p> <p>Then, there are R more lines containing C integers each. The cth integer on the rth line is Mr,c, the thinness of the square in the rth row and cth column.</p> <p>Output Format: Print the output in a separate lines contains the maximum number of squares in a beautiful subrectangle.</p>				

```

#include <stdio.h>

void a(){

printf(" bool ok[309][309][309]; ");

}

int main()

{

int t;

scanf("%d",&t);

while(t--){

    int A[309][309];

    int B[309][309]={};

    int r,c,l;
}

```

```

int i,j,k;
scanf("%d %d %d",&r,&c,&l);
for(i=0;i<r;i++){
    for(j=0;j<c;j++){
        scanf("%d",&A[i][j]);
        if(j==0 || A[i][j]!=A[i][j-1])
            B[i][j]=1;
        else
            B[i][j]=B[i][j-1]+1;
    }
}
int res = 0;
for(i=0;i<r;i++){
    for(j=0;j<c;j++){
        int target = B[i][j],res1=0;
        for(k=i;k<r;k++){
            if(B[k][j]>=target)
                res1+=target;
            else
                break;
        }
        if(res1 > res)
            res=res1;
    }
}
printf("%d\n",res);
}
return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Searching	Question Information	Level 1	Challenge 9
Problem	<p>Question Description: Suresh have "N" rectangles. A rectangle is Silver if the ratio of its sides is in between [1.6, 1.7], both inclusive. Your task is to find the number of silver rectangles.</p> <p>Constraints: $1 \leq N \leq 10^5$ $1 \leq W,$ $H \leq 10^9$</p> <p>Input Format: First line: Integer "N" denoting the number of rectangles Each of the "N" following lines: Two integers W, H denoting the width and height of a rectangle</p> <p>Output Format: Print the output in a single line contains find the number of Silver rectangles.</p> <p>Sample Input: 5 10 1</p>					

```
#include <stdio.h>

#include<math.h>

int main()

{

    float n,i,width,height;

    scanf("%f",&n);

    int count=0;

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

        scanf("%f %f",&width,&height);

        if(width/height>=1.6 && width/height<=1.7)

            ++count;

        else if(height/width >=1.6 && height/width<=1.7)

            ++count; }

    printf("%d",count+1);

    return 0;}
```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Searching	Question Information	Level 1 • Challenge 10
Problem	<p>Question Description:</p> <p>Sakthi has been acting strangely for a few days now. Finally, you [his best friend] found out that it was because his project proposal was turned down (rejected).</p> <p>He is working hard to solve the problem, but he is unable to concentrate due to the rejection. Are you able to assist him?</p> <p>Find if n can be expressed as the sum of two desperate numbers (not necessarily dissimilar) given a number n.</p> <p>where desperate numbers are those which can be written in the form of $(a * (a+1))/2$ where $a > 0$.</p> <p>Constraints:</p> <p>$1 \leq n \leq 10^9$</p> <p>Input :</p> <p>The first input line contains an integer n</p> <p>Output :</p> <p>Print "YES" (without the quotes), if n can be represented as a sum of two desperate numbers, otherwise print "NO" (without the quotes).</p>				

```
#include <stdio.h>

int check(int s){

    int n,sum = 0;

    for (n = 1; sum < s; n++) {

        sum += n;

        if (sum == s)

            return 1;

    }

    return -1;
}

int binarySearch(int low,int high,int key){

    return 1;
}

int main() {

    int n, i, flag = 0;

    scanf("%d", &n);

    for (i = 2; i <= n / 2; ++i) {

        if (check(i) == 1) {

            if (check(n - i) == 1) {

                flag = 1;
            }
        }
    }

    binarySearch(1,1,1);

    if (flag == 0)
}
```

```

printf("NO");

else

printf("YES");

return 0;

}

```

SORTING

LEVEL 1

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Sorting	Question Information	Level 1 Challenge 11
Problem	<p>Problem Description: Banana leaf platter is a traditional method of serving rice dishes in South Indian cuisine. Due to the migration of South Indians, banana leaf rice can also be found in areas with significant ethnic South Indian diaspora such as Malaysia and Singapore.</p> <p>Irfan is a banana leaf sales person. he has N stacks of banana leafs.</p> <p>Each stack contains K leafs.</p> <p>Each leaf has a positive beauty value, describing how attractive it looks.</p> <p>Irfan would like to take exactly P leafs to use for lunch today. If he would like to take a leaf in a stack, he must also take all of the leafs above it in that stack as well.</p> <p>Help Irfan pick the P leafs that would maximize the total sum of attractive values.</p> <p>Constraints: $1 \leq T \leq 100$. $1 \leq K \leq 30$. $1 \leq P \leq N * K$. $1 \leq N \leq 50$.</p> <p>Input Format: The first line of the input gives the number of test cases, T. T test cases follow. Each test case begins with a line containing the three integers N, K and P. Then, N lines follow. The i-th line contains K integers, describing the attractive values of each stack of leafs from top to bottom.</p>				

```

#include <bits/stdc++.h>

using namespace std;

#define ll long long

#define ar array

void dummy(){}
int n, k, p, a[50][30];
int dp[51][1501];
void solve() {
    cin >> n >> k >> p;

```

```
memset(dp, 0xc0, sizeof(dp));  
dp[0][0]=0;  
for(int i=0; i<n; ++i) {  
    memcpy(dp[i+1], dp[i], sizeof(dp[0]));  
    for(int j=0, s=0; j<k; ++j) {  
        cin >> a[i][j];  
        s+=a[i][j];  
        //use j+1 plates  
        for(int l=0; l+j+1<=p; ++l)  
            dp[i+1][l+j+1]=max(dp[i][l]+s, dp[i+1][l+j+1]);  
    }  
}  
cout << dp[n][p] << "\n";  
}  
  
int main() {  
    int n, i;  
    cin >> n;  
    for(i=0;i<n;i++) {  
        solve();  
    }  
    return 0;  
    cout<<"int max(int a,int b)";  
}
```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Sorting	Question Information	Level 1 • Challenge 12
Problem	<p>Problem Description: One of the biggest MNC has organize the programming contest for their employees. They are providing some integers and find out the longest subarray where the absolute difference between any two elements is less than or equal to 1</p> <p>Constraints: $2 \leq n \leq 100$ $0 < a[i] < 100$</p> <p>Input Format: The first line contains a single integer 'n', the size of the array 'a'. The second line contains 'n' space-separated integers, each an $a[i]$.</p> <p>Output Format: Print the output in a single line contains display the longest subarray where the absolute difference between any two elements is less than or equal to 1</p>				

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

```
using namespace std;
```

```
int main()
```

```
{
```

```
int n,temp;cin>>n;
```

```
map<int,int> mp;
```

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

```
    cin>>temp;
```

```
    mp[temp]++;
}
```

```
int curr,mx=0;
```

```
for(auto pr:mp){
```

```
    curr = mp.find(pr.first+1)==mp.end()?0:mp[pr.first+1];
```

```
    mx = max(mx,pr.second+curr);
```

```
}
```

```
cout<<mx;
```

```
return 0;
```

```
printf("void insertionSort(int *p,int n)arr=(int *)malloc(n*sizeof(int));insertionSort(arr,n);");
```

```
}
```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Sorting	Question Information	Level 1 Challenge 13
Problem	<p>Question description</p> <p>In India, the real estate sector is the second-highest employment generator, after the agriculture sector.</p> <p>It is also expected that this sector will incur more non-resident Indian (NRIL) investment, both in the short term and the long term.</p> <p>Bengaluru is expected to be the most favoured property investment destination for NRILs, followed by Ahmedabad, Pune, Chennai, Goa, Delhi and Dehradun.</p> <p>Ramesh is residing in England. he is willing to invest money in real estate.</p> <p>So he has chosen Bengaluru for good investment.</p> <p>There are N flats for sale in Bengaluru main city.</p> <p>The ith flat costs A_i rupees to buy.</p> <p>Ramesh has a budget of B rupees to spend.</p> <p>What is the maximum number of flats Ramesh can buy?</p> <p>Constraints:</p> <p>1 ≤ T ≤ 100. 1 ≤ B ≤ 10⁵. 1 ≤ A_i ≤ 1000, for all i. 1 ≤ N ≤ 10⁵.</p> <p>Input Format:</p>				

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

using namespace std;

int main()
{
    int t;cin>>t;
    while(t--){
        int n,tot,now=0;cin>>n>>tot;
        std::vector<int>v(n);
        for (int i = 0; i < n; i++) {
            cin>>v[i];
        }
        sort(v.begin(),v.end());
        for (int i = 0; i < n; i++) {
            now+=v[i];
            if(now>tot){
                cout<<i<<endl;
                break;
            }
        }
    }
}
```

```

}

return 0;

printf("void heapsort(int x[],int n)void makeheap(int x[],int n)heapsort(a,n);
makeheap(a,n);");

}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Sorting	Question Information	Level 1	Challenge 14
Problem	<p>Question description</p> <p>Nathan won the man of the match award in the recently concluded local tournament final.</p> <p>So the friends of nathan have asked him to take them to cinemas as a treat for winning man of the match. But Nathan is short of money to take them to cinemas so to postpone the cinema plan he tried to engage them with the programming challenge.</p> <p>The task is, Given an N-dimensional array Arr, where all elements are bigger than or equal to zero. Return the highest possible product of two numbers.</p> <p>There's no need to read or print anything. Your objective is to complete the procedure maxProduct, which accepts as parameters an array of integers Arr[] and n and returns an integer indicating the answer.</p> <p>Constraints: $2 \leq N \leq 10^7$ $0 \leq \text{Arr}[i] \leq 10^4$</p> <p>Input Format:</p> <p>The first line of input contains an integer T denoting the number of test cases.</p> <p>The first line of each test case is N, N is size of array.</p> <p>The second line of each test case contains N input Arr[i].</p> <p>Output Format:</p> <p>Single line represents the maximum product of two numbers possible.</p>					

```

#include <stdio.h>

#include <malloc.h>

void sort(int a[],int n)

{
    int i, j;

    for(i=0;i<n-1;i++)
        for(j=0;j<n-1-i;j++)
            if (a[j] < a[j + 1])
    {
        int tmp = a[j + 1];
        a[j + 1] = a[j];
        a[j] = tmp;
    }
}

int main()

```

```

{
    int t;
    scanf("%d", &t);
    while (t--)
    {
        int n, i, *arr;
        scanf("%d", &n);
        arr = (int *)malloc(sizeof(int) * n);
        for (i = 0; i < n; i++) scanf("%d", &arr[i]);
        sort(arr, n);
        printf("%d\n", arr[0] * arr[1]);
    }
    return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Sorting	Question Information	Level 1 • Challenge 15
Problem	<p>Question description Selvan studies engineering as per his father's wishes, while Aaron, whose family is poor, studies engineering to improve his family's financial situation. Sumanth, however, studies engineering of his simple passion for developing data structure applications.</p> <p>Sumanth is participating in a hackathon for data structure application development.</p> <p>Sumanth's task is to use Insertion Sort to sort the supplied set of numbers.</p> <p>As a result, the input provides the number of components on the first line and the numbers to be sorted on the second line. Print the array's state at the third iteration and the final sorted array in the supplied format in the output.</p> <p>Judge will determine whether the outcome is correct or not.</p> <p>Can you help him?</p> <p>Constraints</p> <p>$1 \leq N \leq 10^5$</p> <p>$1 \leq A_i \leq 10^9$</p> <p>Input Format:</p> <p>The first line of the input contains the number of elements</p> <p>The second line of the input contains the numbers to be sorted.</p> <p>Output Format:</p>				

```

#include <iostream>

#define f(i,a,n) for(i=a;i<n;i++)
using namespace std;

void insertionSort(int arr[], int n)

```

```
{  
for(int i=1;i<n;i++){  
    int curr = arr[i];  
    for(int j=i-1;j>=0;j--){  
        if(arr[j]>curr){  
            arr[j+1]=arr[j];  
            if(j==0)  
                arr[j]=curr;  
        }  
        else{  
            arr[j+1]=curr;  
            j=-1;  
        }  
    }  
    int k;  
    if(i==2){  
        f(k,0,n)  
        cout<<arr[k]<<" ";  
        cout<<endl;  
    }  
}  
void printArray(int arr[],int n)  
{  
    int i;  
    f(i,0,n)  
    cout << arr[i] <<" ";  
}  
int main()
```

```
{
int n;
cin>>n;
int arr[n];
for(int i=0;i<n;i++)
cin>>arr[i];
insertionSort(arr, n);
printArray(arr, n);
return 0;
}
```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Sorting	Question Information	Level 1	Challenge 16
Problem	<p>Problem Description: Let's call an integer array a_1, a_2, \dots, a_n good if $a_i \neq i$ for each i. Let $P(a)$ be the number of pairs $\{i, j\}$ ($1 \leq i < j \leq n$) such that $a_i + a_j = i + j$. Let's say that an array a_1, a_2, \dots, a_n is excellent if:</p> <ol style="list-style-type: none"> 1. a is good; 2. $l \leq a_i \leq r$ for each i; 3. $P(a)$ is the maximum possible among all good arrays of size n. <p>Given n, l and r, calculate the number of excellent arrays modulo 10^{9+7}.</p> <p>Constraints: $1 \leq t \leq 1000$ $2 \leq n \leq 2 \cdot 10^4$ $-10^9 \leq l \leq 1$ $n \leq r \leq 10^9$</p> <p>Input Format: The first line contains a single integer t — the number of test cases. The first and only line of each test case contains three integers n, l, and r.</p>					

```
#include <stdio.h>

#define N 200000
#define MD 1000000007

int min(int a, int b) { return a < b ? a : b; }

int max(int a, int b) { return a > b ? a : b; }

int vv[N + 1], ff[N + 1], gg[N + 1];

void init() {

    int i;
```

```

ff[0] = gg[0] = 1;
for(i=1;i<=N;i++){
    vv[i] = i == 1 ? 1 : (long long) vv[i - MD % i] * (MD / i + 1) % MD;
    ff[i] = (long long) ff[i - 1] * i % MD;
    gg[i] = (long long) gg[i - 1] * vv[i] % MD;
}
int choose(int n, int k) {
    return k < 0 || k > n ? 0 : (long long) ff[n] * gg[k] % MD * gg[n - k] % MD;
}
int main() {
    int t;
    init();
    scanf("%d", &t);
    while(t--) {
        int n, l, r, i, j, k, d, ans;
        scanf("%d%d%d", &n, &l, &r);
        d = min(1 - l, r - n);
        if (n % 2 == 0)
            ans = (long long) choose(n, n / 2) * d % MD;
        else
            ans = (long long) (choose(n, n / 2) + choose(n, n / 2 + 1)) * d % MD;
        while (1) {
            d++;
            i = max(l + d, 1), j = min(r - d, n);
            if (i - j > 1)
                break;
            k = j - i + 1;
            if (n % 2 == 0)
                ans = (ans + choose(k, n / 2 - (i - 1))) % MD;
        }
    }
}

```

```

else

ans = ((long long) ans + choose(k, n / 2 - (i - 1)) + choose(k, n / 2 + 1 - (i - 1))) % MD;

}

printf("%d\n", ans);
}return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Sorting	Question Information	• Level 1 • Challenge 17	
Problem	<p>Problem Description: One day Priya was jogging and realized that her life was boring. Everything was red, even the streets in the best park were red.</p> <p>Therefore she decided to make streets a little bit brighter. She know that every street in the park is a segment laying on the X axis with coordinates X_l, X_r ($X_l \leq X_r$). Streets may intersect or overlap.</p> <p>She chooses any subset of streets and paints them in green. After that she wants to get one continuous green segment. As she really likes number L the length of this segment has to be equal to L.</p> <p>Your task is to determine if it is possible to choose some subset of streets and paint them to get one green segment with the length equal to L.</p> <p>Constraints: $1 \leq \text{sum of all } N \leq 2 * 10^3$ $1 \leq L \leq 10^6$ $1 \leq X_l \leq X_r \leq 10^6$ $1 \leq N \leq 20, 1 \leq X_l \leq X_r \leq 200$, holds for test cases worth 10% of the problem's score. $1 \leq N \leq 100, 1 \leq X_l \leq X_r \leq 200$, holds for test cases worth 20% of the problem's score. </p> <p>Input Format: The first line contains one integer T - the number of test cases. Each test case starts with two integers N and L, denoting the number of streets and Priya's favorite number L. The next N lines contain two integers X_l, X_r, denoting the left and right borders of the street.</p> <p>Output Format: Print the output in a separate lines "Yes" if it is possible to paint some street and "No" otherwise.</p>					• Level 1 • Challenge 17

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

```
using namespace std;
```

```
#define ll long long int
```

```
#define F first
```

```
#define S second
```

```
const int N = 2021;
```

```
pair<int, int> a[N];
```

```
int main() {
```

```
    ll t;
```

```
    cin >> t;
```

```
    while(t--){
```

```

int n,L;
scanf("%d %d", &n, &L);
for(;; i=0;i<n;i++) {
    scanf("%d %d", &a[i].F, &a[i].S);
}
sort(a , a + n);
int possible=0;
for (;; i=0;i<n;i++) {
    int maxright = a[i].F + L;
    int cur_right = a[i].S;
    for(;; j=0;j<n;j++) {
        if ((a[j].F <= cur_right) && (a[j].F > a[i].F) && (a[j].S <= maxright)) {
            cur_right = max(cur_right, a[j].S);
        }
    }
    if(cur_right==maxright){
        possible = 1;
        break;
    }
}
if(possible == 1) {
    cout<<"Yes"<<endl;
} else {
    cout<<"No"<<endl;
}
}

return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Sorting	Question Information	Level 1	Challenge 18
Problem	<p>Problem Description: Tina owns a match making company, which even to her surprise is an extreme hit. She says that her success rate cannot be matched [Yes, letterplay!] in the entire match-making industry. She follows an extremely simple algorithm to determine if two people are matches for each other. Her algorithm is not at all complex, and makes no sense - not even to her. But she uses it anyway.</p> <p>Let's say that on a given day she decides to select n people - that is, n boys and n girls. She gets the list of n boys and n girls in a random order initially. Then, she arranges the list of girls in ascending order on the basis of their height and boys in descending order of their heights. A girl A_i can be matched to a boy on the same index only, that is, B_i and no one else. Likewise, a girl standing on A_k can be only matched to a boy on the same index B_k and no one else.</p> <p>Now to determine if the pair would make an ideal pair, she checks if the modulo of their heights is 0, i.e., $A_i \% B_i == 0$ or $B_i \% A_i == 0$. Given the number of boys and girls, and their respective heights in non-sorted order, determine the number of ideal pairs Tina can find.</p> <p>Constraints: $1 \leq \text{Test Cases} \leq 10^2$ $1 \leq N \leq 10^4$ $1 \leq A_i, B_i \leq 10^8$</p> <p>Input Format: The first line contains number of test cases. Then, the next line contains an integer, n, saying the number of boys and girls. The next line contains the height of girls, followed by the height of boys.</p> <p>Output Format: Print the number of ideal pairs in a separate lines</p>					

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

using namespace std;

void fun()
{
    int n,i;cin>>n;
    vector<int> a(n),b(n) ;
    for(i=0;i<n;i++)
        cin>>a[i];
    for(i=0;i<n;i++)
        cin>>b[i];
    sort(a.begin(),a.end());sort(b.begin(),b.end());
    int ans = 0;
    for(i=0;i<n;i++) {
        if(a[i]%b[n-1-i]==0 || b[n-1-i]%a[i]==0)
            ans++;
    }
    cout<<ans<<endl;
}

int main(){
    int t;cin>>t;
}
```

```

while(t--) fun();

return 0;

}

```

CHALLENGE INFORMATION



● You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Sorting	Question Information	● Level 1 ● Challenge 19
Problem Description:					
In mathematics , a permutation of a set is, loosely speaking, an arrangement of its members into a sequence or linear order , or if the set is already ordered, a rearrangement of its elements. The word "permutation" also refers to the act or process of changing the linear order of an ordered set.					
Mariappan[M] is alone too and has a permutation p1,p2,...,pn of numbers from 1 to n.					
M thinks that a permutation p1,p2,...,pn beautifulness is defined as value of $\sum p_i - i $, $1 \leq i \leq n$.					
M can swap two elements of the permutation at most once.					
Problem					
Constraints: $1 \leq n \leq 10^5$ $1 \leq p_i \leq n$ all p_i are distinct					
Input Format: First line contains only 'n'. Second line contains the permutation p1, p2...pn separated by space.					
Output Format: Print the output in a single line contains maximum beautifulness that M can get					

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

```
using namespace std;
```

```
int main(){
```

```
int n,i,sum=0;
```

```
cin>>n;
```

```
int arr[n];
```

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

```
cin>>arr[i];
```

```
sort(arr,arr+n);
```

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

```
{
```

```
int z= arr[n-i-1]-(i+1);
```

```
//cout<<z<<" ";
```

```
//cout<<abs(z);
```

```
sum=sum+abs(z);
```

```

}

cout<<sum;

return 0;

cout<<"swap(l,r);";}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Sorting	Question Information	Level 1 Challenge 20
Problem	<p>Question description Sajid is an third year student in a reputed institution. Although he scored well in many subjects, he did not an expert in computer programming languages. But Sajid's computer examination is scheduled for next week. As per the blueprint, many questions would come from the sorting topic. He collected previous year's questions. one of the repeated questions is to sort the given set of numbers using Selection Sort The first line of the input contains the number of elements N, the second line of the input contains the numbers A_i to be sorted. In the output print the the final sorted array in the given format.</p> <p>Can you help him ? Constraints $1 \leq N \leq 10^5$ $1 \leq A_i \leq 10^9$</p> <p>Input: The first line of the input contains the number of elements the second line of the input contains the numbers to be sorted.</p>				

```

#include <bits/stdc++.h>

using namespace std;

void dothis(){

printf("void selectionSort(int arr[],int n)void swap(int *xp,int *yp)void printArray(int arr[],int size)");
}

int main()

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

```

```

sort(v.begin(),v.end());
for (int i = 0; i < n; i++) {
    cout<<v[i]<<' ';
}
cout<<endl;
return 0;
}

```

ARRAYS

LEVEL 1

CHALLENGE INFORMATION

● You have already solved this challenge! Though you can run the code with different logic! X

Course	DS	Session	Arrays	Question Information	● Level 1 ● Challenge 2!
<p>Problem Description: Simon work with Greek squares and matrix traces.</p> <p>The trace of a square matrix is the sum of the values on the main diagonal [which runs from the upper left to the lower right].</p> <p>An $B \times B$ square matrix is a Greek square if each cell contains one of B different values, and no value is repeated within a row or a column. In this problem, we will deal only with "beautiful Greek squares" in which the B values are the integers between 1 and B.</p> <p>Given a matrix that contains only integers between 1 and B, we want to compute its trace and check whether it is a beautiful Greek square. To give some additional information, instead of simply telling us whether the matrix is a beautiful Greek square or not, show the number of rows and the number of columns that contain repeated values.</p> <p>Constraints: $1 \leq T \leq 100$. $2 \leq B \leq 100$. $1 \leq A_{i,j} \leq B$, for all i, j.</p> <p>Input Format: The first line of the input gives the number of test cases, T. T test cases follow. Each starts with a line containing a single integer B: the size of the matrix to explore. Then, B lines follow. The i-th of these lines contains B integers $A_{i,1}, A_{i,2}, \dots, A_{i,B}$. $A_{i,j}$ is the integer in the i-th row and j-th column of the matrix.</p> <p>Output Format: Print the output in a single lines contains the number of rows and the number of columns that contain repeated values.</p>					

```

#include <bits/stdc++.h>

using namespace std;

int t,i,j,tes,n,x,y,sum;

int a[1007][1007];

map<int,bool> udah;

void solve(){}
int main() {
    solve();
}

```

```
scanf("%d",&t);
for (tes=1 ; tes<=t ; tes++) {
    scanf("%d",&n);
    for (i=1 ; i<=n ; i++) {
        for (j=1 ; j<=n ; j++) {
            scanf("%d",&a[i][j]);
        }
    }
    sum = 0;
    x = 0;
    y = 0;
    for (i=1 ; i<=n ; i++) {
        udah.clear();
        for (j=1 ; j<=n ; j++) {
            if (udah[a[i][j]]) x++, j = n;
            udah[a[i][j]] = true;
        }
    }
    for (j=1 ; j<=n ; j++) {
        udah.clear();
        for (i=1 ; i<=n ; i++) {
            if (udah[a[i][j]]) y++, i = n;
            udah[a[i][j]] = true;
        }
    }
    for (i=1 ; i<=n ; i++) sum += a[i][i];
    printf("%d %d %d\n",sum,x,y);
}
return 0;
```

```

cout<<"for(i=0;i<n;i++); int g[105][105];";
}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Arrays	Question Information	Level 1 Challenge 22
Problem	<p>Problem Description: For some reason, your school's football team has chosen to spell out the numbers on their jerseys instead of using the usual digits. Being great fans, you're going to be ready to cheer for your favorite players by bringing letter cards so you can spell out their number. Each fan has different favorites, so they each need to bring different sets of letters.</p> <p>The English spellings for the numbers 0 to 12 are: ZERO ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE TEN ELEVEN TWELVE</p> <p>Input Format: Read a set of integers from 0 to 12, separated by spaces, representing one fan's favorite players. The last integer will be 999, marking the end of the line.</p> <p>Output Format: Print the same numbers, then a period and a space. Then, in alphabetical order, print all the letters the fan needs to be able to spell any one of the jersey numbers provided</p>				

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int curr;
    multiset<char> mp;
    string names[] =
    {"", "ONE", "TWO", "THREE", "FOUR", "FIVE", "SIX", "SEVEN", "EIGHT", "NINE", "TEN", "ELEVEN", "TWELVE"};
    while(cin>>curr){
        if(curr==999){
            cout<<"0999"<<'.'<<' ';
            break;
        }
        cout<<curr<<' ';
        if(curr>12)continue;
        string now = names[curr];
        for(auto ch:now){
            mp.insert(ch);
        }
    }
}

```

```

}

for (auto ch : mp) {

cout<<ch<<' ';

}

return 0;

printf("char nums[13][256]for(n=0;n<26;n++)");

}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Arrays	Question Information	● Level 1 ● Challenge 23
<p>Problem Description: Umesh has n mixtures in front of him, arranged in a row. Each mixture has one of 100 different colors (colors have numbers from 0 to 99). He wants to mix all these mixtures together. At each step, he is going to take two mixtures that stand next to each other and mix them together, and put the resulting mixture in their place. Functional Description: When mixing two mixtures of colors a and b, the resulting mixture will have the color $(a+b) \bmod 100$. Also, there will be some smoke in the process. The amount of smoke generated when mixing two mixtures of colors a and b is $a \cdot b$. Find out what is the minimum amount of smoke that Umesh can get when mixing all the mixtures together.</p> <p>Problem</p> <p>Constraints: $1 \leq n \leq 100$</p> <p>Input Format: There will be a number of test cases in the input. The first line of each test case will contain n, the number of mixtures, The second line will contain n integers representing the initial colors of the mixtures.</p> <p>Output Format: For each test case, output the minimum amount of smoke.</p>					

```

#include<stdio.h>

typedef long long unsigned LLU;

LLU min_smoke[100][100];

int color[100][100];

LLU smoke(int n){

int i,j,l;

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        min_smoke[i][j]=10000000000000000;
    }
}

```

```

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

for(l=2;l<=n;l++){
    int e = n-l;
    for(i=0;i<=e;i++){
        int k = i+l-1;
        for(j=i;j<k;j++){
            LLU sm = min_smoke[i][j] + min_smoke[j+1][k] + color[i][j]*color[j+1][k];
            int cl = (color[i][j]+color[j+1][k])%100;
            if(sm<min_smoke[i][k]){
                min_smoke[i][k] = sm;
                color[i][k] = cl;
            }
        }
    }
}

return min_smoke[0][n-1];
}

int main(void){
    int n;
    while(scanf("%d",&n)!=EOF){
        int i;
        for(i=0;i<n;i++){
            scanf("%d",&(color[i][i]));
        }
        printf("%llu\n",smoke(n));
    }
    return 0;
}

```

```

printf("scount[100][100]colours[100]");
}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Arrays	Question Information	Level 1	Challenge 24
Problem	<p>Problem Description: Caleb likes to challenge Selvan's math ability.</p> <p>He will provide a starting and ending value that describes a range of integers, inclusive of the endpoints.</p> <p>Selvan must determine the number of square integers within that range.</p> <p>Note: A square integer is an integer which is the square of an integer, e.g. 1, 4, 9, 16, 25</p> <p>Constraints: 1 <= q <= 100 1 <= start <= 10^9 1 <= end <= 10^9</p> <p>Input Format: The first line contains 'q', number of test cases. Each of the next 'q' lines contains two space-separated integers, representing 'start' and 'end'.</p> <p>Output Format: Print the number of square integers within that range.</p>					

```

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

void Calc_square(long int start,long int end){

    int i,count=0;

    for(i=start;i<=end;i++){
        int n=sqrt(i);

        if(n==sqrt(i))
            count++;

    }

    printf("%d\n",count);
}

int main()
{
    long int q,start,end;
    scanf("%ld",&q);

    while(q--){
        scanf("%ld %ld",&start,&end);

```

```

    Calc_square(start,end);

}

return 0;

}

```

CHALLENGE INFORMATION

>You have already solved this challenge! Though you can run the code with different logic.

Course	DS	Session	Arrays	Question Information	• Level 1 • Challenge 25
Problem	<p>Problem Description:</p> <p>saravanan with his friends going to the theatre for a movie.</p> <p>The seating arrangement is triangular in size.</p> <p>Theatre staffs insisted the audience to sit in odd row if the seat number is odd and in even row if the seat number is even.</p> <p>But the instruction is very confusing for saravanan and his friends.</p> <p>So help them with the seating layout so that they can sit in correct seats.</p> <p>Constraints:</p> <p>$4 \leq N \leq 20$</p> <p>Input Format:</p> <p>Only line of input has single integer value representing the number of rows in the theatre.</p> <p>Output Format:</p> <p>Print the layout based on the number of rows specified in input.</p> <p>Refer sample testcases for format specification.</p>				

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

```

int main()
{
    int i,j,k,N;
    scanf("%d",&N);
    for(i=1;i<=N;i++)
    {
        if(i%2==0)
        {
            k=2;
        }
        else
        {
            k=1;
        }
    }
}
```

```

for(j=1; j<=i; j++,k+=2)

{
    printf("%d ", k);

}

printf("\n");

}

return 0;

}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Arrays	Question Information	● Level 1 ● Challenge 26
<p>Problem Description:</p> <p>There are K nuclear reactor chambers labelled from 0 to K-1. Particles are bombarded onto chamber 0. The particles keep collecting in the chamber 0.</p> <p>However if at any time, there are more than N particles in a chamber, a reaction will cause 1 particle to move to the immediate next chamber(if current chamber is 0, then to chamber number 1), and all the particles in the current chamber will be destroyed and same continues till no chamber has number of particles greater than N.</p> <p>Given K,N and the total number of particles bombarded [A], find the final distribution of particles in the K chambers. Particles are bombarded one at a time. After one particle is bombarded, the set of reactions, as described, take place.</p> <p>After all reactions are over, the next particle is bombarded. If a particle is going out from the last chamber, it has nowhere to go and is lost.</p> <p>Constraints:</p> <p>A will be between 0 and 1000000000 inclusive.</p> <p>N will be between 0 and 100 inclusive.</p> <p>K will be between 1 and 100 inclusive.</p> <p>Input Format:</p> <p>The input will consist of one line containing three numbers A,N and K separated by spaces.</p> <p>All chambers start off with zero particles initially.</p> <p>Output Format:</p> <p>Consists of K numbers on one line followed by a newline.</p> <p>The first number is the number of particles in chamber 0,</p> <p>The second number is the number of particles in chamber 1 and so on.</p>					

```

#include <stdio.h>

int main()

{int a[100],r=0,c[r],k,i,b;

scanf("%d%d%d",&a[r],&c[r],&k);

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

    b=a[r]%(c[r]+1);

    printf("%d ",b);

    a[r]=a[r]/(c[r]+1);
}

```

```

    }
//while(a>0){}
    return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Arrays	Question Information	● Level 1 ● Challenge 27
				<p>Question description</p> <p>Simon is studying B.Tech.-Mechanical Engineering.</p> <p>He's going to attend a computer science-based subject exam this semester.</p> <p>Due to the less preparation in the previous monthly tests, his internal mark decreased.</p> <p>His computer science Professor made an offer one more chance to boost up his internal marks.</p> <p>Professor assigns a program to Simon for the internal mark boostup.</p> <p>So Simon wants to identify the element of array which occurs most time in the array</p> <p>Can you help him ?</p> <p>Function Description</p>  <p style="text-align: center;">1 Occurs 4 times</p>	

```

#include <stdio.h>

int main()
{
    int n,i,j;
    scanf("%d",&n);
    int a[n],ctrl=0,f;
    for(i= 0;i< n;i++)
    {
        scanf("%d",&a[i]);
        int max=0;
        for(j=0;j<n;j++)

```

```
{  
    if(a[i]==a[j])  
        max++;  
}  
if(max>ctrl)  
{  
    ctrl=max;  
    f=a[i];  
}  
}  
printf("%d",f);  
return 0;  
}
```

CHALLENGE INFORMATION

• • •

✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course

DS

Session

Arrays

Question Information

• Level 1 • Challenge 28

Question description

Sajid is a First year student in reputed institution.

Although he scored well in many subjects, he did not an expert in Algorithms.

But Sajid's computer examination is scheduled for next week.

As per the blueprint, many questions would come from the Arrays topic.

He collected previous year's questions. one of the repeated questions is you need to reverse the array in C Programming Language.

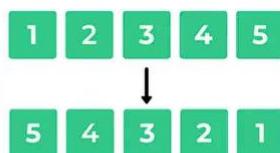
Can you help him ?

Function Description

Algorithm

Problem

```
Start
Input -> n
Input -> elements of array
Start loop (i) for 0 to n/2
    exchange
        array[i] -> array[n-1-i]
    print array (space)
```

Sample Test Case : 1**Sample Test Case : 2**

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int n,i;
```

```
cin>>n;
```

```
int arr[n];
```

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

```
    cin>>arr[i];
```

```
for(i=0;i<n/2;i++)
```

```
{
```

```
    int temp;
```

```
    temp=arr[i];
```

```
    arr[i]=arr[n-1-i];
```

```
    arr[n-1-i]=temp;
```

```

}

for(int i=0;i<n;i++)
cout<<arr[i]<<" ";
return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Arrays	Question Information	• Level 1 • Challenge 29							
Problem	<p>Problem Description: Ram has provide inputs two numbers 'p' and 'q' to Sakthi. He wants to creates a matrix of size $p \times q$ [p rows and q columns] in which every elements is either Y or 0. The Ys and 0s must be filled alternatively, the matrix should have outermost rectangle of Ys, then a rectangle of 0s, then a rectangle of Ys, and so on..</p> <p>Constraints:</p> <p>$1 \leq p, q \leq 1000$</p> <p>Input Format:</p> <p>Input lines must be how many rows and columns in that matrix, also values must be separate space.</p> <p>Output Format:</p> <p>Print the output in a separate lines.</p>											
	<p>✓ Logical Test Cases</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 50%;">Test Case 1</th> <th style="width: 50%;">Test Case 2</th> </tr> <tr> <td>INPUT (STDIN)</td> <td>INPUT (STDIN)</td> </tr> <tr> <td style="text-align: center;">6 7</td> <td style="text-align: center;">5 8</td> </tr> </table>					Test Case 1	Test Case 2	INPUT (STDIN)	INPUT (STDIN)	6 7	5 8	
	Test Case 1	Test Case 2										
	INPUT (STDIN)	INPUT (STDIN)										
	6 7	5 8										

```

#include <bits/stdc++.h>

using namespace std;

void ss(){
    cout<<"while(top<=bottom && right>=left)";
}

void fillOX(int m, int n){

int i, k = 0, l = 0, r = m, c = n;

char a[m][n], x = 'Y';

while (k < m && l < n) {
    for (i = l; i < n; ++i)
        a[k][i] = x;
}

```

```
k++,i=k;  
while(i < m)  
    a[i][n-1] = x,i++;  
  
n--;  
if (k < m)  
    for (i = n; i >= l; --i)  
        a[m-1][i] = x;  
  
m--;  
if (l < n)  
    for (i = m; i >= k; --i)  
        a[i][l] = x;  
  
l++;  
x = (x == '0')? 'Y': '0';  
}  
  
for (i = 0; i < r; i++) {  
    for (int j = 0; j < c; j++) {  
        cout << a[i][j];  
        if(j < c-1)  
            cout<<" ";  
    }  
    cout <<"\n";  
}  
}  
}  
  
int main()  
{  
    int m,n;  
    cin>>m>>n;  
    fill0X(m, n);  
}
```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Arrays	Question Information	Level 1 Challenge 30
				<p>Question description</p> <p>Malar is a First year student in reputed institution.</p> <p>Although he scored well in many subjects, he did not an expert in Algorithms.</p> <p>But malar's computer examination is scheduled for next week.</p> <p>As per the blueprint, many questions would come from the Arrays topic.</p> <p>He collected previous year's questions. one of the repeated questions is you need to find the pairs in Array with given sum.</p> <p>Can you help him ?</p> <p>Function Description</p>	

Sample Test Case



Sum to be checked : 7

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

using namespace std;

/*int getPairsCount(int arr[], int n, int sum)
{
    return 1;
}*/
```

```
int main()
{
    int n;
    cin>>n;
    int array[n];
    for(int i=0;i<n;i++)
        cin>>array[i];
    int sum;
    cin>>sum;
    int count = 0,i,j;
```

```
for(i=0;i<n;i++){
    for(j=i+1;j<n;j++){
        if (array[i] + array[j] == sum){
            count++;
            cout<<"["<<array[i]<<" "<<array[j]<<"]"<<endl;
        }
    }
}
cout<<"Total Number of Pairs:"<<count;
//getPairsCount(arr, n, sum);
//return 0;
}:
```

LINKED LISTS

LEVEL 1

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Linked List	Question Information	• Level 1 • Challenge 32
<p>Question description</p> <p>Once upon a time, in French Canada, there lived a fat old woman named Tante Adela.</p> <p>She lived alone in her barn with her large grey cat and her cows.</p> <p>She got up quite early one morning since it was baking day and she had a lot to accomplish.</p> <p>She carried a pile of wood to her oven outdoors.</p> <p>she ran across some old school classmates, with whom she reminisced about their school days and a mental exam competition.</p> <p>One of the competition's requirements was to write a C function that searches a singly linked list for a given key "x." (Iterative).</p> <p>If x is contained in the linked list, the function should return true; otherwise, it should return false.</p> <p>For example,</p> <p>if the key to be searched is 15 and linked list is 14->21->11->30->10,</p> <p>then function should return false.</p> <p>If key to be searched is 14, then the function should return true.</p>					
<p>Problem</p> <p>Constraints</p> <p>1 < N < 1000</p> <p>1 < X < 1000</p>					
<p>Input Format</p>					

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

using namespace std;

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

void push(struct node** head_ref, int new_key)
{
    struct node* new_node = new node();
    new_node->key = new_key;
    new_node->next = (*head_ref);
    (*head_ref) = new_node;
}
```

```
}

bool search(struct node* head,int x)

{

node* current = head;

while (current != NULL)

{



if (current->key == x)

return true;

current = current->next;

}

return false;

}

int main()

{

struct node* head = NULL;

int x,n,t;

cin>>n;

while(n--){

cin>>t;

push(&head,t);

}

cin>>x;

search(head, x)? cout<<"Yes" : cout<<"No";

return 0;

}
```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Linked List	Question Information	Level 1 Challenge 33
Problem	<p>Question description</p> <p>Kapildev works in the mobile phone marketing industry.</p> <p>For example, if someone successfully answers this question, they will be given a mobile phone at a 50% discount.</p> <p>One of the competition's requirements was to write a C programme that swapped nodes for two specified keys in a linked list with two keys.</p> <p>By altering linkages, nodes should be switched.</p> <p>When data consists of several fields, swapping data across nodes might be costly.</p> <p>It is reasonable to presume that all keys in a linked list are unique.</p> <p>example :</p> <p>Given linked list : 10->15->12->13->20->14 and swap keys X=12 and Y=20.</p> <p>Linked list after swapping : 10->15->20->13->12->14 [if X or Y or Both are not present in Linked List, ABORT the Swapping]</p>				

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

using namespace std;

int main()

{
    int n,x,y,indx=-1,indy=-1;cin>>n;
    int arr[n];
    for (int i = 0; i < n; i++)
        cin>>arr[i];
    cin>>x>>y;
    for (int i = 0; i < n; i++) {
        if(arr[i]==x){
            indx = i;
            break;
        }
    }
    for (int i = 0; i < n; i++) {
        if(arr[i]==y){
            indy = i;
            break;
        }
    }
```

```

}

cout<<"before Swapping:";

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

cout<<"-->"<<arr[n-1-i];

}

if(indy!=-1&&indx!=-1){

swap(arr[indx],arr[indy]);

}

cout<<endl<<"after Swapping:";

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

cout<<"-->"<<arr[n-1-i];

}

return 0;

printf("struct node{ struct node *next; void swapNodes(struct node **head_ref,int x,int y)"); 

}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Linked List	Question Information	Level 1 Challenge 34
Problem	<p>Question description</p> <p>Arulmozhivarman's Dream came true after he got an Appointment order from Google.Simon's family was very happy of his achievement.</p> <p>The company mentioned Basic Salary, DA, HRA with some other benefits.</p> <p>But not highlighted the Gross salary in the order.</p> <p>Arulmozhivarman's father wanted to know the Gross salary of his son.</p> <p>Arulmozhivarman try to his gross salary from HR department, they informed that you have to get pass grade in first month entry test. the entry test has 5 questions. one of the question was, to delete alternate nodes of a linked list. In this program we want to remove alternate nodes from the singly linked list which is start from the second node of the linked list and remove all the alternate nodes of it.</p> <p>Can you help Arulmozhivarman?</p> <p>Function Description</p> <ul style="list-style-type: none"> First Take two pointers a and b. Let initially a points to head and b points to pointer of a ,that is second node . Than make the link of a to point to pointer of b and free b. Next is move to a to its pointer that is next node. Next is move b to next node of a. Continue loop until a and b becomes NULL. <p>Constraints</p> <p>0<n<100</p>				

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```

{
    int n;
    cin>>n;
    for(int i=1;i<=n;i+=2)
        cout<<i<<" ";
    return 0;
    cout<<"void insert_Data(struct node **head) void delete_Alt(struct node **head)";
}

```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Linked List	Question Information	Level 1	Challenge 35
				<p>Question description</p> <p>Admission for the current Academic year is happening in Most of the Universities across the Country.</p> <p>Once the Students got admitted they are assigned a unique Registration Number.</p> <p>Admission in charges used to assign give these details in some order.</p> <p>But during enrolment of the student there is a specific entrance test for admitted students to get scholarship.</p> <p>now admission cell conducting a test. So your task is generate a program for a singly linked list, find middle of the linked list.</p> <p>If there are even nodes, then print second middle element.</p> <p>For example,</p> <p>if given linked list is 1>2>3>4>5 then output should be 3.</p> <p>Problem</p> <p>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.</p>		

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

using namespace std;

void MandatoriesSuck(){

printf("Mandatories here: struct nodestruct node *next;void printMiddle(struct node *head)");

}

class Node {

public:

int data;

Node* next;

Node(int dat){


```

```
data = dat;
next = NULL;
}
};

Node* insertNode(Node* head, int data){
if(head==NULL){
return new Node(data);
}
if(head->next==NULL){
head->next = new Node(data);
return head;
}
insertNode(head->next,data);
return head;
}
void printNode(Node* head){
if(head==NULL){
return;
}
printNode(head->next);
cout<<"-->"<<head->data;
}
int main()
{
int n,temp,mid;cin>>n;
Node* head = NULL;
for (int i = 0; i < n; i++) {
cin>>temp;
if(i==(n/2 -(n%2==0?1:0)))mid = temp;
```

```

head = insertNode(head,temp);

}

cout<<"Linked list:";

printNode(head);

cout<<endl<<"The middle element is ["<<mid<<']';

return 0;

}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Linked List	Question Information	Level 1	Challenge 38
Problem	<p>Question description</p> <p>Lalitha is a IT expert who training youngsters struggling in coding to make them better.</p> <p>Lalitha usually gives interesting problems to the youngsters to make them love the coding. One such day Lalitha provided the youngsters to solve that Add a node at the end.</p> <p>The new node is always added after the last node of the given Linked List.</p> <p>For example if the given Linked List is 5->10->15->20->25 and</p> <p>we add an item 30 at the end,</p> <p>then the Linked List becomes 5->10->15->20->25->30.</p> <p>Since a Linked List is typically represented by the head of it,</p> <p>we have to traverse the list till end and then change the next of last node to new node.</p> <p>Constraints:</p> <p>1 < arr <100</p> <p>INPUT</p> <p>First line contains the number of datas- N. Second line contains N integers(i.e, the datas to be inserted).</p> <p>OUTPUT</p>					

```

#include <stdio.h>

#include<stdlib.h>

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

void display();

int main() {
    int n;
    scanf("%d",&n);
    struct node *temp, *p2;
    start=NULL;

```

```
while(n) {  
    temp=(struct node *)malloc(sizeof(struct node));  
    scanf("%d", &temp->data);  
    temp->next = NULL;  
    if(start == NULL){  
        start= temp;  
        p2 = temp;  
    }  
    else  
    {  
        p2->next=temp;  
        // while(p2 != NULL && p2 -> next != NULL p2=p2->next;  
        p2=p2->next;  
    }--n;  
}  
display();  
return 0;  
}  
  
void display() {  
    struct node *temp;  
    temp = start;  
    printf("Linked List:");  
    while(temp != NULL)  
    {  
        printf("->%d",temp->data);  
        temp = temp->next;  
    }  
}
```

CHALLENGE INFORMATION

You have already solved this challenge! Though you can run the code with different logic!

Course	DS	Session	Linked List	Question Information	Level 1 Challenge 39
				Question description Professor Shiva decided to conduct an industrial visit for final year students, but he set a condition that if students received a passing grade in the surprise test, they would be eligible to go on the industrial visit. He asked the students to study a topic linked list for 10 minutes before deciding to conduct a surprise test. Professor-mandated questions, such as the deletion of nodes with a certain data D, are now being asked. For example if the given Linked List is 5->10->15->10->25 and delete after 10 then the Linked List becomes 5->15->25.	
Problem				Constraints 1 < N < 100 1 < D < 1000 Input Format First line contains the number of datas- N. Second line contains N integers[the given linked list].	

```
#include<iostream>

using namespace std;

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

void display();

void deleteNode(node*& head, int val)
{
    if (head == NULL) {
        return;
    }
    if (head->data == val) {
        node* t = head;
        head = head->next;
        delete (t);
        return;
    }
    deleteNode(head->next, val);
}
```

```
int main() {
    int n;
    scanf("%d",&n);
    struct node *temp, *p2;
    start=NULL;
    for(int i=0;i<n;i++){
        temp=(struct node *)malloc(sizeof(struct node));
        scanf("%d", &temp -> data);
        temp->next = NULL;
        if(start == NULL){
            start= temp;
            p2 = temp;
        }
        else
        {
            p2->next=temp;
            p2=p2->next;
        }
    }
    int x;
    cin>>x;
    //display();
    for(int i=0;i<n;i++)
        deleteNode(start,x);
    display();
    return 0;
    cout<<"void del()void create() ";
}
void display() {
```

```

struct node *temp;

temp = start;

printf("Linked List:");

while(temp != NULL)

{

printf("->%d",temp->data);

temp = temp->next;

}

}

```

STACKS

LEVEL 1

CHALLENGE INFORMATION



● You have already solved this challenge! Though you can run the code with different logic!

X

Course	DS	Session	Stack	Question Information	● Level 1 ● Challenge 41
Problem Description:					
Dr. Malar was booking a tour package of IRCTC from Chennai to Delhi for his family.					
Two of the relatives was interested in joining to this tour.					
these two persons are studying engineering in computing technology. only one tickets are remaining in the IRCTC portal.					
So, Dr. Malar decided to book one ticket for out of those persons also along with his family members.					
she wants to identify the one person out of these persons. he decided to conduct a technical task to identify the right person to travel.					
the task was that, implement two stack operations in an array					
Can you help them to complete the task?					
Constraints					
0<n<5 only five elements has to be practiced for this operation					
first element pushed into stack1					
second element pushed into stack2, likewise elements pushed into alternative stacks vice versa.					
Function Description					
<ul style="list-style-type: none"> • Create a data structure <code>twoStacks</code> that represents two stacks. • Implementation of <code>twoStacks</code> should use only one array, i.e., both stacks should use the same array for storing elements. • Following functions must be supported by <code>twoStacks</code>. 					

```

#include <bits/stdc++.h>

using namespace std;

void non(){printf("void push1(int x)void push2(int x)int pop1()int pop2();")}

int main()

{
    int n,k;
    for (int i = 0; i < 5; i++) {

```

```

k=n;

cin>>n;

}

cout<<"Popped element from stack1 is:"<<n<<endl;

cout<<"Popped element from stack2 is:"<<k;

return 0;

}

```

CHALLENGE INFORMATION



● You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Stack	Question Information	● Level 1 ● Challenge 43
Question description					
Given a permutation of numbers from 1 to N . Among all the subarrays, find the number of unique pairs $\{a,b\}$ such that $a \neq b$ and a is maximum and b is the second maximum in that subarray.					
Input: First line contains an integer, N ($1 \leq N \leq 105$). the Second line contains N space-separated distinct integers, A_i ($1 \leq A_i \leq N$), denoting the permutation.					
Output: Print the required answer.					
Explanation:					
Sample Input					
5					
1 2 3 4 5					
Sample output					
4					
Problem All the possible subarrays are:					

```

#include <stdio.h>

int main(){

int num,i,count=0,a[100001],stck[100001],top=-1;

scanf("%d", &num);

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

scanf("%d",&a[i]);

while(top!=-1 && stck[top]<a[i]) {

top--;

count++;

}

```

```

}

if (top!=-1) {

count++;

}

stck[++top]=a[i];

}

printf("%d",count);

return 0;

}

```

CHALLENGE INFORMATION

✔ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Stack	Question Information	● Level 1 ● Challenge 45
Problem	<p>Question description</p> <p>Rajinikanth organized a technical round interview in an Animation company for the set of computer science candidates.</p> <p>the task is to implement stack operations for two stacks and merge the stacks into one.</p> <p>Get two sets of stack elements and reverse them, then merge them into one stack.</p> <p>Rajinikanth has given the deadline of only 15 minutes to complete the problem.</p> <p>Can you Help the candidates to complete the problem within the specified time limit?</p> <p>Function Description</p> <p>a) push(): Adds the new item at the beginning of the linked list using the first pointer. b) pop(): Removes an item from the beginning using the first pointer. c) merge(): Links the first pointer second stack as next of the last pointer of the first list.</p> <p>Constraints</p> <p>$0 < n, m < N$</p> <p>$1 < arr[i] < 1000$</p> <p>Input Format:</p> <p>First-line indicates n & m, where n is the number of elements to be pushed into the stack and m is the number of pop operations that need to be performed</p> <p>The next line indicates the n number stack elements</p> <p>Output Format:</p> <p>First-line indicates the top of the element of the stack</p>				

```
#include <iostream>
```

```
using namespace std;
```

```
class node {
```

```
public:
```

```
int data;
```

```
node* next;
};

class mystack {
public:
node* head;
node* tail;
mystack()
{
head = NULL;
tail = NULL;
}
};

mystack* create()
{
mystack* ms = new mystack();
return ms;
}

void push(int data,mystack* ms)
{
node* temp = new node();
temp->data = data;
temp->next = ms->head;
if (ms->head == NULL)
ms->tail = temp;
ms->head = temp;
}

int pop(mystack* ms)
{
if (ms->head == NULL) {
```

```
cout << "stack underflow" << endl;
return 0;
}
else {
node* temp = ms->head;
ms->head = ms->head->next;
int popped = temp->data;
delete temp;
return popped;
}
}

void merge(mystack* ms1,mystack* ms2)
{
if (ms1->head == NULL)
{
ms1->head = ms2->head;
ms1->tail = ms2->tail;
return;
}
ms1->tail->next = ms2->head;
ms1->tail = ms2->tail;
}

void display(mystack* ms)
{
node* temp = ms->head;
while (temp != NULL) {
cout << temp->data << " ";
temp = temp->next;
}
```

```
}

int main()
{
mystack* ms1 = create();
mystack* ms2 = create();

int n,m,t;

cin>>n>>m;

for(int i=0;i<n;i++)
{
    cin>>t;
    push(t,ms1);
}

for(int i=0;i<m;i++)
{
    cin>>t;
    push(t,ms2);
}

merge(ms1, ms2);

for(int i=0;i<n+m;i++)
cout<<pop(ms1)<<" ";

}
```

CHALLENGE INFORMATION

• • •

✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Stack	Question Information	• Level 1 • Challenge 46
<p>Question description</p> <p>Hassan enjoys jumping from one building to the next. However, he merely jumps to the next higher building and stops when there are none accessible. The amount of stamina necessary for a voyage is equal to the xor of all the heights Hassan leaps till he comes to a halt.</p> <p>If heights are [1 2 4], and he starts from 1, goes to 2 stamina required is $1 \oplus 2 = 3$, then from 2 to 3. Stamina for the entire journey is $1 \oplus 2 \oplus 4 = 7$. Find the maximum stamina required if can start his journey from any building.</p> <p>Constraints</p> <p>$1 \leq N \leq 10^5$</p> <p>$1 \leq \text{Height} \leq 10^9$</p> <p>Input</p> <p>First line: N, no of buildings.</p> <p>Second line: N integers, defining heights of buildings.</p> <p>Output</p> <p>Single Integer is the maximum stamina required for any journey.</p> <p>Explanation:</p> <p>8</p> <p>1 2 3 8 6 4 7 9</p> <p>Considering the input given by you, 1st starting point is 1. From this building, Hassan will try to find the next higher building that is building 2 here and so on. Likewise he will complete all pass.</p> <p>Pass 1: 1-> 2->3->8-> 9, XOR = 1</p>					
Problem					

```
#include <stdio.h>

int main() {
    int i, j, arr[1000000], n, temp=0,st[1000000]={0};

    scanf("%d",&n);

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

    st[n-1] = arr[n-1];
    temp = arr[n-1];

    for(i=n-2;i>=0;i--) {
        for(j=i+1;j<n;j++){
            if(arr[i]<arr[j]) {
                st[i]=arr[i]^st[j];
            }
        }
        break;
    }
}
```

```

}

if(st[i] == 0)

st[i] = arr[i];

if(st[i] > temp)

temp = st[i];

}

printf("%d",temp);

return 0;

}

```

CHALLENGE INFORMATION

✔ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Stack	Question Information	● Level 1 ● Challenge 48
Problem Description: <p>You are given an array A of n integers.</p> <p>You have to make a queue and stack the given integers.</p> <p>The queue should contain only prime numbers and the stack should contain only composite numbers.</p> <p>All numbers in the array will be >1.</p> <p>The rule to form the stack and queue is that you should be able to generate the array using the pop and dequeue operations.</p> <p>Note: Please read this explanation carefully</p> <p>Let the array A contains 5 integers: 7 , 21 , 18 , 3 , 12 then the content of queue and stack will be : Queue : 7 , 3 Stack : 12 , 18 , 21 Now if you follow the rules of stack and queue then you see that you can generate the array using the pop operations of stack and dequeue operation of the queue as follows :</p> <p>Problem</p> <pre> dequeue from the queue: 7 pop from stack: 7 , 21 pop from stack: 7 , 21, 18 dequeue from queue : 7 , 21 , 18 , 3 pop from stack : 7 , 21 , 18 , 3 , 12 </pre>					

```

#include<bits/stdc++.h>

using namespace std;

bool isPrime(int n)

{

if(n<=1)

return false;

for(int i=2;i<n;i++)

```

```
if(n%i==0)
return false;
return true;
}

int main(){
stack<int> stack;
int n;
cin>>n;
int a[n];
for(int i=0;i<n;i++){
cin>>a[i];
if(isPrime(a[i]))
cout<<a[i]<<" ";
else
stack.push(a[i]);
}
cout<<endl;
while(!stack.empty()){
cout<<stack.top()<<" ";
stack.pop();
}
return 0;
cout<<"int read_int() void push(int stack[],int data) top++;" ;
}
```

CHALLENGE INFORMATION

>You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Stack	Question Information	Level 1 • Challenge 49
Problem Description					
A and B are playing a game. In this game, both of them are initially provided with a list of n numbers. (Both have the same list but their own copy).					
Now, they both have a different strategy to play the game. A picks the element from start of his list. B picks from the end of his list.					
You need to generate the result in form of an output list.					
Method to be followed at each step to build the output list is:					
<ol style="list-style-type: none">1. If the number picked by A is bigger than B then this step's output is 1 . B removes the number that was picked from their list.2. If the number picked by A is smaller than B then this step's output is 2 . A removes the number that was picked from their list.3. If both have the same number then this step's output is 0 . Both A and B remove the number that was picked from their list.					
This game ends when at least one of them has no more elements to be picked i.e. when the list gets empty.					
Problem					
Output the built output list.					
Constraints					
$1 \leq N \leq 10^6$					
$1 \leq \text{num} \leq 10^9$					
Input format:					
First line consists of a number n , size of the list provided.					
Next line consists of n numbers separated by space.					

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

```
using namespace std;
```

```
int main()
```

```
{
```

```
int n;cin>>n;
```

```
vector<int>v(n);
```

```
for (int i = 0; i < n; i++)
```

```
cin>>v[i];
```

```
int a=0,b=n-1;
```

```
while(a<n&&b>=0){
```

```
if(v[a]==v[b]){


```

```
b--;a++;


```

```
cout<<"0 ";
```

```
}
```

```
else if(v[a]>v[b)){
```

```
b--;


```

```

cout<<"1 ";
}

else{
    a++;
    cout<<"2 ";
}
}

return 0;
cout<<"if(a[i]>a[j])";
}

```

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Stack	Question Information	Level 1 • Challenge 50
Problem	<p>Question description</p> <p>Hassan gets a job in a software company in Hyderabad. The training period for the first three months is 20000 salary. Then incremented to 25000 salaries.</p> <p>Training is great but they will give you a programming task every day in three months. Hassan must finish it in the allotted time. His teammate Jocelyn gives him a task to complete the concept of Infix to Prefix Conversion for a given expression. can you help him?</p> <p>Functional Description:</p> <ul style="list-style-type: none"> Step 1: Reverse the infix expression i.e A+B*C will become C*B+A. Note while reversing each '(' will become ')' and each ')' becomes '('. Step 2: Obtain the "nearly" postfix expression of the modified expression i.e CB*A+. Step 3: Reverse the postfix expression. Hence in our example prefix is +A*BC. <p>Constraints</p> <p>the input should be a expressions</p> <p>Input Format</p> <p>Single line represents the Infix expressions</p> <p>Output Format</p> <p>Single line represents the Prefix expression</p>				

▼ Logical Test Cases

Test Case 1

Test Case 2

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

```
using namespace std;
```

```
bool isOperator(char c)
```

```
{
```

```

return (!isalpha(c) && !isdigit(c));
}

int getPriority(char C)

{
if (C == '-' || C == '+')
    return 1;
else if (C == '*' || C == '/')
    return 2;
else if (C == '^')
    return 3;
return 0;
}

string infixToPostfix(string infix)

{
infix = '(' + infix + ')';
int l = infix.size();
stack<char> char_stack;
string output;
for (int i = 0; i < l; i++) {
    // If the scanned character is an
    // operand, add it to output.
    if (isalpha(infix[i]) || isdigit(infix[i]))
        output += infix[i];
    // If the scanned character is an
    // '(', push it to the stack.
    else if (infix[i] == '(')
        char_stack.push('(');
    // If the scanned character is an
    // ')', pop and output from the stack
}

```

```
// until an '(' is encountered.

else if (infix[i] == ')') {

    while (char_stack.top() != '(') {

        output += char_stack.top();

        char_stack.pop();

    }

    // Remove '(' from the stack

    char_stack.pop();

}

// Operator found

else

{

    if (isOperator(char_stack.top()))

    {

        if(infix[i] == '^')

        {

            while (getPriority(infix[i]) <= getPriority(char_stack.top()))

            {

                output += char_stack.top();

                char_stack.pop();

            }

        }

    }

    else

    {

        while (getPriority(infix[i]) < getPriority(char_stack.top()))

        {

            output += char_stack.top();

            char_stack.pop();

        }

    }

}
```

```

// Push current Operator on stack
char_stack.push(infix[i]);}}
while(!char_stack.empty()){
output += char_stack.top();
char_stack.pop();}
return output;}

string infixToPrefix(string infix)
{
/* Reverse String
 * Replace ( with ) and vice versa
 * Get Postfix
 * Reverse Postfix */
int l = infix.size();
// Reverse infix
reverse(infix.begin(), infix.end());
// Replace ( with ) and vice versa
for (int i = 0; i < l; i++) {
if (infix[i] == '(') {
infix[i] = ')';
i++;}
else if (infix[i] == ')') {
infix[i] = '(';
i++;}}
string prefix = infixToPostfix(infix);
// Reverse postfix
reverse(prefix.begin(), prefix.end());
return prefix;}

// Driver code
int main()

```

```

{
string s;
cin>>s;
cout << infixToPrefix(s) << std::endl;
return 0;
}

```

QUEUES

LEVEL 1

CHALLENGE INFORMATION



✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Queue	Question Information	Level 1 • Challenge 51
Question description					
Ramesh is an DS expert training youngsters struggling in DS to make them better.					
Ramesh usually gives interesting problems to the youngsters to make them love the DS.					
One such day Sathya provided to the youngsters to solve the task such that, delete an element in a Queue, Queue data structures work on the FIFO architecture so the element that has entered first in the list will go out from the list first.					
Youngsters were lacking the idea to solve the problem.					
Being an exciting youngster can you solve it?					
Function Description					
<ol style="list-style-type: none"> 1. Create a main function for the program. 2. In this main function accept the size and data of the queue by the user. 3. After that enqueue these element in the queue using an enqueue() function 					

```

#include <stdio.h>
#include<stdlib.h>
#define MAX 50
int queue[MAX];
int rear = - 1;
int front = - 1;
void enqueue(int data)
{

```

```
if(rear == MAX- 1)
printf("Queue Overflow n");
else
{
if(front== - 1)
front = 0;
rear = rear + 1;
queue[rear] = data;
}
}

void dequeue()
{
if(front == - 1 || front > rear)
{
printf("Queue Underflow \n");
}
else
front = front + 1;
}

void display()
{
int i;
if(front == - 1)
printf("Queue is empty \n");
else
{
for(i=front;i<=rear;i++)
printf("%d ", queue[i]);
printf("\n");
}
```

```

}

}

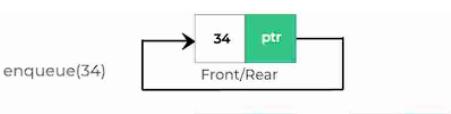
int main()
{
    int n,i,data;
    scanf("%d",&n);
    for(i=0;i<n;i++){
        scanf("%d",&data);
        enqueue(data);}
        printf("Dequeueing elements:\n");
        for(i=0;i<n-1;i++){
            dequeue();
            display();}
    return 0;
}

```

CHALLENGE INFORMATION



✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Queue	Question Information	Level 1	Challenge 52
				<p>Question description</p> <p>Anderson is a Placement trainer. he is working as CDC trainer in reputed institution that during training the youngsters are struggling in queue concept.</p> <p>Anderson usually gives interesting problems to the students to make them love the DS.</p> <p>One such day Joe Anderson provided to the final year students to solve the task such that, Circular Queue using Linked List. there is no memory waste while using Circular Queue, it is preferable than using a regular queue.</p> <p>Because linked lists allow for dynamic memory allocation, they are simple to build.</p> <p>Circular Queue implementation using linked list is identical to circular linked list except that circular Queue has two pointers front and back whereas circular linked list only has one pointer head.</p> <p>Final Year students were lacking the idea to solve the problem.</p> <p>Being an exciting youngster can you solve it?</p> <p>Function Description</p>  <pre> enqueue(34) +-----+ 34 +-----+ ptr +-----+ Front/Rear +-----+ </pre>		

#include <stdio.h>

```
#include <stdlib.h>

struct node *f = NULL;
struct node *r = NULL;

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

void enqueue(int d)
{
    struct node *n;
    n = (struct node*)malloc(sizeof(struct node));
    if(n==NULL){
        printf("Queue is Full");
    }
    else{
        n->data = d;
        n->next = NULL;
        if(f==NULL){
            f=r=n;
        }
        else{
            r->next = n;
            r=n;
        }
    }
}

int dequeue()
{
```

```
int val = -1;
struct node* t;
t = f;
if(f==NULL){
printf("Queue is Empty\n");
}
else{
f = f->next;
val = t->data;
free(t);
}
return val;
}

int main()
{
int n,i,t;
scanf("%d",&n);
for(i=0;i<n;i++)
{
scanf("%d",&t);
enqueue(t);
}
for(i=0;i<n;i++){
printf("%d\n",dequeue());
}
return 0;
}
```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Queue	Question Information	• Level 1 • Challenge 53
Problem Description					
The Monk recently learnt about priority queues and requested his teacher for a fun challenge to solve. As a result, his teacher devised a simple task. He now possesses A, which is an integer array. He wishes to discover the product of the greatest, second largest, and third largest integers in the range [1,i] for each index i.					
Note: Two numbers can be the same value-wise but they should be distinct index-wise.					
Constraints: 1 <= N <= 100000 0 <= A[i] <= 1000000					
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.					

```
#include <stdio.h>

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

```
{  
bigger=a[0];  
big=a[1];  
}  
biggest=a[2];  
if(biggest==big)  
{  
temp=bigger;  
bigger=biggest;  
biggest=temp;  
}  
if(biggest<big)  
{  
temp=big;  
big=biggest;  
biggest=bigger;  
bigger=temp;  
}  
prod=big * bigger * biggest;  
printf("%lli\n",prod);  
for(i=3;i<N;i++)  
{  
if(a[i]>biggest)  
{  
big=bigger;  
bigger=biggest;  
biggest=a[i];  
}  
else if(a[i]>bigger)
```

```

{
big=bigger;
bigger=a[i];
}
else if(a[i]>big)
{
big=a[i];
}
prod=big * bigger * biggest;
printf("%lli\n",prod);
}
return 0;
}

```

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Queue	Question Information	Level 1 Challenge 54
Question description					
				Given a string, you want to reorder its characters so that no two adjacent characters are the same. What is the lexicographically minimal such string?	
Constraints					
				<ul style="list-style-type: none"> • $1 \leq n \leq 10^6$ 	
Problem					
		Input		The only input line as a string of length n consisting of characters A–Z.	
			Output		Print the lexicographically minimal reordered string where no two adjacent characters are the same. If it is not possible to create such a string, print -1.

```
#include <stdio.h>
#include <string.h>
#define N 1000000
#define A 26
```

```
int main() {
    static char cc[N + 1];
    static int kk[A];
    int n, i, p, a, b, c;

    scanf("%s", cc);
    n = strlen(cc);
    for(i=0;i<n;i++) {
        a = cc[i] - 'A';
        kk[a]++;
    }
    for (a = 0; a < A; a++)
        if (n < kk[a] * 2 - 1) {
            printf("-1\n");
            return 0;
        }
    p = -1;
    for (i = 0; i < n; i++) {
        a = 0;
        while (a < A && (a == p || kk[a] == 0))
            a++;
        b = 0;
        for (c = 1; c < A; c++)
            if (kk[b] < kk[c])
                b = c;
        a = a != b && n - i - 1 < kk[b] * 2 - 1 ? b : a;
        kk[a]--;
        cc[i] = a + 'A';
        p = a;
    }
}
```

```

    }
    printf("%s\n", cc);
    return 0;
}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Queue	Question Information	● Level 1 ● Challenge 55	
Problem	<p>Question description You are given a string. You can remove any number of characters from it, but you cannot change the order of the remaining characters.</p> <p>How many different strings can you generate?</p> <p>Constraints</p> <ul style="list-style-type: none"> • $1 \leq n \leq 5 \cdot 10^5$ <p>Input The first input line contains a string of size n. Each character is one of a–z.</p> <p>Output Print one integer: the number of strings modulo $10^9 + 7$.</p>					

```

#include <stdio.h>

#define N 500000
#define MD 1000000007

int main() {
    static char cc[N + 1];
    static int kk[26];
    int i, k, c, kc;
    scanf("%s", cc);
    k = 0;
    for(i=0;cc[i];i++) {
        c = cc[i] - 'a';
        kc = kk[c];
        kk[c] = k + 1;
        k = (k + (kk[c] - kc) % MD) % MD;
    }
}

```

```

}

printf("%d\n", (k + MD) % MD);

return 0;

}

```

CHALLENGE INFORMATION

You have already solved this challenge ! Though you can run the code with different logic !

Course	DS	Session	Queue	Question Information	Level 1 • Challenge 57
				<p>Question description</p> <p>There is a bit string consisting of n bits. Then, there are some changes that invert one given bit. Your task is to report, after each change, the length of the longest substring whose each bit is the same.</p> <p>Constraints</p> <ul style="list-style-type: none"> • $1 \leq n \leq 2 \cdot 10^5$ • $1 \leq m \leq 2 \cdot 10^5$ • $1 \leq x_i \leq n$ <p>Problem</p> <p>Input Format</p> <p>The first input line has a bit string consisting of n bits. The bits are numbered $1, 2, \dots, n$. The next line contains an integer m: the number of changes. The last line contains m integers x_1, x_2, \dots, x_m describing the changes.</p> <p>Output Format</p> <p>After each change, print the length of the longest substring whose each bit is the same.</p>	

```

#include <stdio.h>

#include <string.h>

#define N 200000

#define M (1 << 18) /* M = pow2(ceil(log2(N))) */

int max(int a, int b) { return a > b ? a : b; }

char cc[N + 1];

int pp[M * 2], qq[M * 2], tr[M * 2];

void pull(int k, int l, int r) {

int m = (l + r) / 2, k1 = k * 2 + 1, k2 = k * 2 + 2;

pp[k] = pp[k1];

if (pp[k1] == m - l && cc[l] == cc[m])

pp[k] += pp[k2];

```

```

qq[k] = qq[k2];
if (qq[k2] == r - m && cc[r - 1] == cc[m - 1])
qq[k] += qq[k1];
tr[k] = max(tr[k1], tr[k2]);
if (cc[m - 1] == cc[m])
tr[k] = max(tr[k], qq[k1] + pp[k2]);
}

void build(int k, int l, int r) {
int m;
if (r - l == 1) {
pp[k] = qq[k] = tr[k] = 1;
return;
}
m = (l + r) / 2;
build(k * 2 + 1, l, m);
build(k * 2 + 2, m, r);
pull(k, l, r);
}

void update(int k, int l, int r, int i) {
int m;
if (r - l == 1) {
cc[i] = cc[i] == '0' ? '1' : '0';
return;
}
m = (l + r) / 2;
if (i < m)
update(k * 2 + 1, l, m, i);
else
update(k * 2 + 2, m, r, i);
}

```

```
pull(k, l, r);
}

int main() {
    int n, m;
    scanf("%s%d", cc, &m);
    n = strlen(cc);
    build(0, 0, n);
    while (m--) {
        int i;
        scanf("%d", &i), i--;
        update(0, 0, n, i);
        printf("%d ", tr[0]);
    }
    printf("\n");
    return 0;
}
```

CHALLENGE INFORMATION

...

✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Queue	Question Information	• Level 1 • Challenge 58
<p>Question description</p> <p>lala is a Placement trainer. he is working as CDC trainer in reputed institution that during training the youngsters are struggling in queue concept. Lala usually gives interesting problems to the students to make them love the DS. One such day Lala provided to the final year students to solve the task such that, Queue implementation with arrays as using linked list for implementing queue, Queue data structures work on the FIFO architecture so the element that has entered first in the list will go out from the list first.</p> <p>Final Year students were lacking the idea to solve the problem.</p> <p>Being an exciting youngster can you solve it?</p> <p>Function Description</p>					
<p>Problem</p>					
<div style="border: 1px solid black; padding: 10px;"><p>enqueue(data)</p><ul style="list-style-type: none">• Build a new node with given data.• Check if the queue is empty or not.• If queue is empty then assign new node to front and rear.• Else make next of rear as new node and rear as new node.</div>					
<div style="border: 1px solid black; padding: 10px;"><p>dequeue()</p><ul style="list-style-type: none">• Check if queue is empty or not.• If queue is empty then dequeue is not possible.• Else store front in temp• And make next of front as front.</div>					
<div style="border: 1px solid black; padding: 10px;"><p>print()</p><ul style="list-style-type: none">• Check if there is some data in the queue or not.• If the queue is empty print "No data in the queue."• Else define a node pointer and initialize it with front.• Print data of node pointer until the next of node pointer becomes NULL.</div>					

```
#include <stdio.h>

#include <stdlib.h>

struct node *front = NULL;

struct node *rear = NULL;

struct node

{

    int data;

    struct node *next;

};

void linkedListTraversal(struct node *ptr)

{

    //printf("Printing the elements of this linked list\n");

    while (ptr != NULL)

    {

        printf("%d ", ptr->data);

    }

}
```

```
ptr = ptr->next;
}
}

void enqueue(int d)
{
    struct node* new_n;
    new_n = (struct node*)malloc(sizeof(struct node));
    if(new_n==NULL){
        printf("Queue is Full");
    }
    else{
        new_n->data = d;
        new_n->next = NULL;
        if(front==NULL){
            front=rear=new_n;
        }
        else{
            rear->next = new_n;
            rear=new_n;
        }
    }
}

int main()
{
    int n,i,t;
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        scanf("%d",&t);
```

```

enqueue(t);
}

linkedListTraversal(front);

return 0;
}

```

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Queue	Question Information	Level 1 Challenge 59
Question description					
Problem	Consider the following string transformation:				
	1. append the character # to the string (we assume that # is lexicographically smaller than all other characters of the string) 2. generate all rotations of the string 3. sort the rotations in increasing order 4. based on this order, construct a new string that contains the last character of each rotation				
	For example, the string babc becomes babc#. Then, the sorted list of rotations is #babc, abc#b, babc#, bc#ba, and c#bab. This yields a string cb#ab.				
	Constraints				
	$1 \leq n \leq 10^6$				
	Input				
	The only input line contains the transformed string of length $n+1$. Each character of the original string is one of a-z.				
	Output				
	Print the original string of length n .				

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

```
using namespace std;
```

```
int main() {
```

```
    int i;
```

```
    string s; cin>>s;
```

```
    vector<int> v;
```

```
    vector<int> a[26];
```

```
    int n= s.size();
```

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

```
        if (s[i] == '#')
```

```
v.push_back(i);

else
    a[s[i]-'a'].push_back(i);

}

for (int i = 0; i < 26; i++) {
    for (auto j: a[i])
        v.push_back(j);

}

string ans;

int j = v[v[0]];

while(s[j] != '#') {
    ans += s[j];
    j = v[j];
}

cout<<ans;

return 0;
}
```

CHALLENGE INFORMATION

...

You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Queue	Question Information	Level 1 • Challenge 60
Question description					
Your task is to construct a tower in N days by following these conditions:					
<ul style="list-style-type: none">• Every day you are provided with one disk of distinct size.• The disk with larger sizes should be placed at the bottom of the tower.• The disk with smaller sizes should be placed at the top of the tower.					
The order in which tower must be constructed is as follows:					
<ul style="list-style-type: none">• You cannot put a new disk on the top of the tower until all the larger disks that are given to you get placed.					
Print N lines denoting the disk sizes that can be put on the tower on the ith day.					
Constraints:					
$1 \leq N \leq 10^6$					
$1 \leq \text{size} \leq N$					
Input format					
<ul style="list-style-type: none">• First line: N denoting the total number of disks that are given to you in the N subsequent days• Second line: N integers in which the ith integers denote the size of the disks that are given to you on the ith day					
Note: All the disk sizes are distinct integers in the range of 1 to N.					
Output format					
Print N lines. In the ith line, print the size of disks that can be placed on the top of the tower in descending order of the disk sizes.					
If on the ith day no disks can be placed, then leave that line empty.					

```
#include<stdio.h>

int main()
{
    long int disk,temp[1000000]={0},size,i,max;
    scanf("%ld",&disk);
    max=disk;
    for(i=0;i<disk;i++)
    {
        scanf("%ld",&size);
        temp[size]=size;
        if(size==max)
        {
            while(temp[size])
            {
                printf("%ld ",temp[size]);
            }
        }
    }
}
```

```

size--;
}

max=size;
printf("\n");
}

else
printf("\n");
}

return 0;
}

```

TREE 1

LEVEL 1

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Tree 1	Question Information	● Level 1 ● Challenge 61
Question description					
	Given an array of n integers, your task is to process q queries of the form: what is the sum of values in range [a,b]?				
Constraints					
	<ul style="list-style-type: none"> • $1 \leq n, q \leq 2 \cdot 10^5$ • $1 \leq x_i \leq 10^9$ • $1 \leq a \leq b \leq n$ 				
Problem					
	Input The first input line has two integers n and q: the number of values and queries. The second line has n integers x_1, x_2, \dots, x_n : the array values. Finally, there are q lines describing the queries. Each line has two integers a and b: what is the sum of values in range [a,b]?				
	Output Print the result of each query.				

```

#include<bits/stdc++.h>

using namespace std;

int main(){

int n,q,i,a,b;

cin>>n>>q;

```

```

int x[n];
for(i=0;i<n;i++)
cin>>x[i];
while(q--){
int sum=0;
cin>>a>>b;

for(i=a;i<=b;i++)
sum=sum+x[i-1];
cout<<sum<<endl;
}
}

```

CHALLENGE INFORMATION
• • •

✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Tree 1	Question Information	● Level 1 ● Challenge 62
Question description <p>There are n hotels on a street. For each hotel you know the number of free rooms. Your task is to assign hotel rooms for groups of tourists. All members of a group want to stay in the same hotel.</p> <p>The groups will come to you one after another, and you know for each group the number of rooms it requires. You always assign a group to the first hotel having enough rooms. After this, the number of free rooms in the hotel decreases.</p> Constraints <ul style="list-style-type: none"> • $1 \leq n, m \leq 2 \cdot 10^5$ • $1 \leq h_i \leq 10^9$ • $1 \leq r_i \leq 10^9$ Problem Input <p>The first input line contains two integers n and m: the number of hotels and the number of groups. The hotels are numbered $1, 2, \dots, n$. The next line contains n integers h_1, h_2, \dots, h_n: the number of free rooms in each hotel. The last line contains m integers r_1, r_2, \dots, r_m: the number of rooms each group requires.</p> Output <p>Print the assigned hotel for each group. If a group cannot be assigned a hotel, print 0 instead.</p>					

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

```
void solve(){}
int main()
{
    solve();
    int n,m,i;
    cin>>n>>m;
    int a[n],b[n];
    for(i=0;i<n;i++)
        cin>>a[i];
    for(i=0;i<n;i++)
        cin>>b[i];
    for(i=0;i<m;i++){
        int f=0,j=0;
        for(;j<n;j++){
            if(a[j]>=b[i]){
                a[j]-=b[i];
                f=1;
            }
            break;
        }
        if(f>0)
            cout<<j+1<<" ";
        else
            cout<<"0 ";
    }
    return 0;
}
```

CHALLENGE INFORMATION

• • •

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Tree 1	Question Information	● Level 1 ● Challenge 63
Problem					
Question description					
<p>The sam is enjoying a wonderful vacation in Byteland. What makes the sam impressed the most is the road system of the country. Byteland has N cities numbered 1 through N. City 1 is the capital of Byteland. The country also has N-1 bidirectional roads connecting the cities. The i-th road connects two different cities u_i and v_i. In this road system, people can travel between every pair of different cities by going through exactly one path of roads.</p>					
<p>The roads are arranged in such a way that people can distinguish two cities only when both cities have different number of roads connected to it. Such two cities will be considered similar. For example, city A is similar to the capital if the number of roads connected to city A is equal to the number of roads connected to the capital.</p>					
<p>On each day during the vacation, the sam wants to have a trip as follows. He chooses two cities A and B such that the sam will visit city B if he goes from A to the capital using the shortest path. Then, the sam will visit the cities on the shortest path from A to B through this path. Please note that A may be equal to B; that means the sam will enjoy the day in a single city.</p>					
<p>The sam does not want to have similar trips. Two trips are considered similar if and only if they both have the same number of visited cities and for each i, the i-th city visited in one trip is similar to the i-th city visited in the other trip.</p>					
<p>The sam wants to have as many different, namely not similar, trips as possible. Help him count the maximum number of possible trips such that no two of them are similar.</p>					
<p>Input Format The first line of the input contains a single integer N. The i-th line of next $N-1$ lines contains two space-separated integers u_i and v_i, denoting the i-th road.</p>					
<p>Output Format Output a single line containing the maximum number of different trips.</p>					

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

using namespace std;

#define ll long long

struct state { int len,link; map<int,int> next;};

const int MAXL=200005;state st[MAXL];int sz,last;

void sa_init(){
    st[0].len=0; st[0].link=-1; sz++; last = 0;
}

void sa_extend(int c){
    int cur=sz++; st[cur].len=st[last].len+1;

    int p=last;
    while(p!=-1 && !st[p].next.count(c)){
        st[p].next[c]=cur; p=st[p].link;
    }

    if(p == -1){ st[cur].link = 0;}
    else{ int q=st[p].next[c];
        if(st[p].len+1 == st[q].len){st[cur].link=q;}
    }
}
```

```

else { int clone = sz++;
    st[clone].len = st[p].len + 1;
    st[clone].next = st[q].next;
    st[clone].link = st[q].link;
    while (p!=-1 && st[p].next[c]==q){
        st[p].next[c]=clone; p=st[p].link; }
    st[q].link = st[cur].link = clone;
}
} last = cur; }

const int N = 1e5 + 100;
vector<int> G[N];
int deg[N];
void dfs(int s, int p){
    sa_extend(deg[s]);
    int tmp = last;
    for(auto it : G[s]){
        if(it == p) continue;
        dfs(it , s);
        last = tmp;
    }
}
II dp[MAXL];
int main(){
    ios_base::sync_with_stdio(false);
    cout.tie(0); cin.tie(0);
    sa_init();
    int n; cin >> n;
    int u , v;
    for(int i = 0;i < n-1 ; ++i){

```

```

cin >> u >> v;
G[u].push_back(v);
G[v].push_back(u);
++deg[u];
++deg[v];
}

dfs(1 , -1);

vector<pair<int,int> > topo(sz);
for(int i = 0;i < sz ; ++i) topo[i] = make_pair(st[i].len , i);
sort(topo.begin() , topo.end());
for(int i = sz-1;i >= 0; --i){
    u = topo[i].second;
    dp[u] = 1;
    for(auto it : st[u].next){
        dp[u] += dp[it.second];
    }
}
cout << dp[0]-1 << endl;

return 0;
}

```

CHALLENGE INFORMATION

• • •

You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Tree 1	Question Information	Level 1 Challenge 64
Question description					
You are given a tree consisting of n nodes.					
The <i>diameter</i> of a tree is the maximum distance between two nodes. Your task is to determine the diameter of the tree.					
Input					
The first input line contains an integer n : the number of nodes. The nodes are numbered $1, 2, \dots, n$.					
Then there are $n-1$ lines describing the edges. Each line contains two integers a and b : there is an edge between nodes a and b .					
Output					
Print one integer: the diameter of the tree.					
Constraints					
<ul style="list-style-type: none"> • $1 \leq n \leq 2 \cdot 10^5$ • $1 \leq a, b \leq n$ 					
Problem					

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

using namespace std;

#define vi vector<int>
#define rep(i,a,b) for (int i=a; i<b; ++i)
#define pb push_back

vi adj[200005];

int d=0,x=0;

void solve(){}
void dfs(int s, int p, int dep){
    for (auto i: adj[s]){
        if (i!=p){
            dfs(i,s,dep+1);
            if (dep>d) d = dep, x = s;
        }
    }
}

int main(){
    solve();
    int n;
```

```

cin>>n;;
rep(i,0,n-1){
int x,y; cin>>x>>y;
adj[x].pb(y), adj[y].pb(x);
}
dfs(1,0,0);
dfs(x,0,0);
cout<<d;
return 0;
cout<<"void link(int i,int j) void dfs(int p,int i,int d)";
}

```

CHALLENGE INFORMATION



✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Tree 1	Question Information	● Level 1 ● Challenge 65
Question description					
				Darsh, Ratik, Swathy are good friends, They are studying Pre-final year B.E. in reputed institution. Swathy's uncle was a DS teacher in school of computing technologies. He asks to make an application for Insertion in a Binary Search Tree.	
				You have to do the work for the development of the code of their thinking.	
				Function Description	
				<ul style="list-style-type: none"> The left subtree for any given node will only contain nodes which are lesser than the current node The right subtree for any given node will only contain nodes which are greater than the current node Each subtree must also follow the above rules of BST 	
Problem				Constraints	
				0<n<100	
				0<arr[i]<1000	
				Input Format	
				first line indicates the size of the array	
				second line indicates the array elements according to the array size	
				Output Format	
				single line represents the binary search tree.	

```

#include <bits/stdc++.h>

using namespace std;

struct node {
    int dat;
    struct node *left,*right;
}

```

```
};
```

```
struct node* newNode(int item) {
```

```
    struct node* n = new node;
```

```
    n->dat = item;
```

```
    n->left = NULL;
```

```
    n->right = NULL;
```

```
    return n;
```

```
}
```

```
struct node* insertNode(node* head, node* n) {
```

```
    if(head==NULL){
```

```
        return n;
```

```
    }
```

```
    else{
```

```
        if(head->dat > n->dat){
```

```
            head->left = insertNode(head->left,n);
```

```
        }
```

```
        else{
```

```
            head->right = insertNode(head->right,n);
```

```
        }
```

```
        return head;
```

```
    }
```

```
}
```

```
void dfs(node* head){
```

```
    if(head==NULL) return;
```

```
dfs(head->left);
cout<<head->dat<<' ';
dfs(head->right);

}
```

```
int main()
{
    int n,temp;
    struct node* head = NULL;
    cin>>n;
    for (int i = 0; i < n; i++) {
        cin>>temp;
        head = insertNode(head,newNode(temp));
    }
    dfs(head);

    return 0;
}
```

CHALLENGE INFORMATION

• • •

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Tree 1	Question Information	Level 1 Challenge 66
Question description					
A company has n employees, who form a tree hierarchy where each employee has a boss, except for the general director. Your task is to process q queries of the form: who is employee x 's boss k levels higher up in the hierarchy?					
Input					
The first input line has two integers n and q : the number of employees and queries. The employees are numbered $1, 2, \dots, n$, and employee 1 is the general director. The next line has $n-1$ integers e_2, e_3, \dots, e_n : for each employee $2, 3, \dots, n$ their boss. Finally, there are q lines describing the queries. Each line has two integers x and k : who is employee x 's boss k levels higher up?					
Output					
Print the answer for each query. If such a boss does not exist, print -1.					
Constraints					
<ul style="list-style-type: none"> • $1 \leq n, q \leq 2 \cdot 10^5$ • $1 \leq e_i \leq i-1$ • $1 \leq x \leq n$ • $1 \leq k \leq n$ 					

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

using namespace std;

#define ll long long
#define MAX 200005
#define pb push_back

vector<int>tree[MAX];
ll up[MAX][20];

void solve(){}
void link(int i,int j){
    up[i][0]=j;
    for(int m=1;m<20;m++){
        if(up[i][m-1]!=-1)
            up[i][m]=up[up[i][m-1]][m-1];
        else
            up[i][m]=-1;
    }
    for(auto child:tree[i]){

```

```

if(child!=j) link(child,i);

}

}

int ans_query(int src,int jump){

if(src==-1 or jump==0)return src;

for(int i=19;i>=0;i--){

if( jump>= (1<<i)) {

return ans_query(up[src][i],jump-(1<<i));

}

}

return 1;

}

int main(){

solve();

int n,q;

cin>>n>>q;

for(int i=2;i<=n;i++){

int ee;

cin>>ee;

tree[i].pb(ee);

tree[ee].pb(i);

}

link(1,-1);

while(q--){

int node,jump;

cin>>node>>jump;

cout<<ans_query(node,jump)<<endl;

}
}

```

CHALLENGE INFORMATION

• • •

You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Tree 1	Question Information	Level 1 Challenge 69
Question description					
A forest is an undirected graph without cycles (not necessarily connected).					
Mohana and john are friends in kerala, both of them have a forest with nodes numbered from 1 to n , and they would like to add edges to their forests such that:					
<ul style="list-style-type: none"> After adding edges, both of their graphs are still forests. They add the same edges. That is, if an edge $\{u,v\}$ is added to Mohana's forest, then an edge $\{u,v\}$ is added to john's forest, and vice versa. 					
Mohana and johns want to know the maximum number of edges they can add, and which edges to add.					
Constraints:					
$1 \leq n \leq 105$,					
$0 \leq m_1$					
$m_2 < n$					
$1 \leq u, v \leq n$, $u \neq v$					
Input					
The first line contains three integers n , m_1 and m_2 — the number of nodes and the number of initial edges in Mohana's forest and john's forest.					
Each of the next m_1 lines contains two integers u and v — the edges in Mohana's forest.					
Each of the next m_2 lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the edges in Johns's forest.					
Output					
The first line contains only one integer k , the maximum number of edges Mohana and John can add.					

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

using namespace std;

typedef long long ll;

const int mod=998244353;

int fa[1005],fa2[1005],n,m1,m2;

int gf(int x,int *f){
    return f[x]==x?x:f[x]=gf(f[x],f);
}

int main(){
    cin>>n>>m1>>m2;
    for(int i=1;i<=n;i++)fa[i]=fa2[i]=i;
    for(int i=1,x,y;i<=m1;i++)cin>>x>>y,fa[gf(x,fa)]=gf(y,fa);
    for(int i=1,x,y;i<=m2;i++)cin>>x>>y,fa2[gf(x,fa2)]=gf(y,fa2);
    cout<<n-max(m1,m2)-1<<'n';
}
```

```

for(int i=1;i<=n;i++){
    for(int j=i+1;j<=n;j++){
        if(gf(i,fa)!=gf(j,fa)&&gf(i,fa2)!=gf(j,fa2)){
            cout<<i<<' '<<j<<'\n';
            fa[gf(i,fa)]=gf(j,fa);
            fa2[gf(i,fa2)]=gf(j,fa2);
        }
    }
}
return 0;
cout<<"while(m1--)";
}

```

CHALLENGE INFORMATION

● You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Tree 1	Question Information	● Level 1 ● Challenge 70
Question description					
Given an array of n integers, your task is to process q queries of the form: what is the minimum value in range [a,b]?					
Constraints					
<ul style="list-style-type: none"> • $1 \leq n, q \leq 2 \cdot 10^5$ • $1 \leq x_i \leq 10^9$ • $1 \leq a \leq b \leq n$ 					
Problem					
Input The first input line has two integers n and q: the number of values and queries. The second line has n integers x1,x2,...,xn: the array values. Finally, there are q lines describing the queries. Each line has two integers a and b: what is the minimum value in range [a,b]?					
Output					
Print the result of each query.					

```

#include <stdio.h>

#define N 200000

#define N_ (1 << 18) /* N_ = pow2(ceil(log2(N))) */

#define INF 0x3f3f3f3f

int tt[N_ * 2];

```

```

void build(int *aa,int k,int l,int r) {
    int m, k1, k2;

    if (r - l == 1) {
        tt[k] = aa[l];
        return;
    }

    m = (l + r) / 2, k1 = k * 2 + 1, k2 = k * 2 + 2;
    build(aa, k1, l, m);
    build(aa, k2, m, r);
    tt[k] = tt[k1] < tt[k2] ? tt[k1] : tt[k2];
}

int query(int k,int l,int r,int ql,int qr) {
    int m, q1, q2;

    if (qr <= l || r <= ql)
        return INF;
    if (ql <= l && r <= qr)
        return tt[k];

    m = (l + r) / 2;
    q1 = query(k * 2 + 1, l, m, ql, qr);
    q2 = query(k * 2 + 2, m, r, ql, qr);
    return q1 < q2 ? q1 : q2;
}

int main() {
    static int aa[N];
    int n, q, i, j;

    scanf("%d%d", &n, &q);
}

```

```

for (i = 0; i < n; i++)
scanf("%d", &aa[i]);
build(aa, 0, 0, n);
while (q--) {
scanf("%d%d", &i, &j), i--;
printf("%d\n", query(0, 0, n, i, j));
}
return 0;
}

```

TREE 2

LEVEL 1

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Tree 2	Question Information	● Level 1 ● Challenge 72
Problem Description:					
Any sequence A of size n is called B-sequence if:					
$A_1 < A_2 < \dots < A_k > A_{k+1} > A_{k+2} > \dots > A_n$ where $1 \leq k \leq n$. That is, a sequence which is initially strictly increasing and then strictly decreasing (the decreasing part may or may not be there).					
All elements in A except the maximum element comes at most twice (once in increasing part and once in decreasing part) and maximum element comes exactly once.					
All elements coming in decreasing part of sequence should have come once in the increasing part of sequence.					
You are given a B-sequence S and Q operations. For each operation, you are given a value val . You have to insert val in S if and only if after insertion, S still remains a B-sequence .					
After each operation, print the size of S . After all the operations, print the sequence S .					
Hint: Think of using some data structure to support insertion of elements in complexity better than linear.					
Input Constraints:					
Problem					
$1 \leq N \leq 105$					
$1 \leq S_i \leq 109$					
$1 \leq Q \leq 105$					
$1 \leq val \leq 109$					
Given sequence S is a B-sequence,					
Input Format:					

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

```
#include<map>
```

```
using namespace std;
```

```
int main() {
```

```
int N,i,maximum=INT_MIN;
scanf("%d", &N);
int S[N];
map<int,int> map;
for(i=0;i<N;i++) {
scanf("%d", &S[i]);
maximum=max(maximum,S[i]);
map[S[i]]++;
}
int temp,Q;
cin>>Q;
for(i=0;i<Q;i++) {
scanf("%d", &temp);
if(temp==maximum) printf("%d\n",N);
else {
if(map[temp]>=2) printf("%d\n",N);
else {
map[temp]++;
N++;
printf("%d\n",N);
maximum=max(maximum,temp);
}}}
for(auto it=map.begin();it!=map.end();it++) printf("%d ",it->first);
for(auto it=map.rbegin();it!=map.rend();it++) {
if(it->second>1) printf("%d ",it->first);
}
}
```

CHALLENGE INFORMATION

• • •

✓ You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Tree 2	Question Information	• Level 1 • Challenge 77
Question Description					
In a movie festival, n movies will be shown. You know the starting and ending time of each movie.					
Your task is to process q queries of the form: if you arrive and leave the festival at specific times, what is the maximum number of movies you can watch?					
You can watch two movies if the first movie ends before or exactly when the second movie starts. You can start the first movie exactly when you arrive and leave exactly when the last movie ends.					
Input					
Problem					
The first input line has two integers n and q : the number of movies and queries.					
After this, there are n lines describing the movies. Each line has two integers a and b : the starting and ending time of a movie.					
Finally, there are q lines describing the queries. Each line has two integers a and b : your arrival and leaving time.					
Output					
Print the maximum number of movies for each query.					
Constraints					
<ul style="list-style-type: none">• $1 \leq n, q \leq 2 \cdot 10^5$• $1 \leq a < b \leq 10^6$					

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

using namespace std;

int dp[1000006][25];

void solve(){}
int main(){
    solve();
    int n, q; cin>>n>>q;
    for (int i = 0; i < n; i++) {
        int x, y; cin>>x>>y;
        dp[y][0] = max(dp[y][0], x);
    }
    for (int i = 1; i <= 1000000; i++)
        dp[i][0] = max(dp[i][0], dp[i-1][0]);
    for (int k = 1; k <= 20; k++)
        cout << dp[k][0] << endl;
}
```

```
for (int i = 1; i <= 1000000; i++)
dp[i][k] = dp[dp[i][k-1]][k-1];
while(q--) {
int x,y; cin>>x>>y;
int ans = 0;
while(y>0) {
int z = 0;
for (int i = 0; i <= 20; i++) {
if (dp[y][i] < x) {
z = i;
break;
}
}
if (z == 0)
break;
ans += (1<<(z-1));
y = dp[y][z-1];
}
cout<<ans<<endl;
}}
```

CHALLENGE INFORMATION

• • •

You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Tree 2	Question Information	Level 1 Challenge 78
Question Description:					
There are n cities and m flight connections between them. Your task is to add new flights so that it will be possible to travel from any city to any other city. What is the minimum number of new flights required?					
Input					
The first input line has two integers n and m: the number of cities and flights. The cities are numbered 1,2,...,n.					
After this, there are m lines describing the flights. Each line has two integers a and b: there is a flight from city a to city b. All flights are one-way flights.					
Output					
First print an integer k: the required number of new flights. After this, print k lines describing the new flights. You can print any valid solution.					
Constraints					
<ul style="list-style-type: none"> • $1 \leq n \leq 10^5$ • $1 \leq m \leq 2 \cdot 10^5$ • $1 \leq a, b \leq n$ 					

```
#include <stdio.h>
#include <string.h>
#define N 100000
#define M 200000
struct L {
    struct L *next;
    int j;
} aa[N], bb[N], aa_[N];
void link(int i, int j) {
    static struct L l91[M * 2], *l = l91;
    l->j = j;
    l->next = aa[i].next, aa[i].next = l++;
    l->j = i;
    l->next = bb[j].next, bb[j].next = l++;
}
void link_(int i, int j) {
```

```
static struct L l91[M], *l = l91;

l->j = j;
l->next = aa_[i].next, aa_[i].next = l++;
}

int po[N], npo;
char visited[N];

void dfs1(int i) {
    struct L *l;

    if (visited[i])
        return;
    visited[i] = 1;
    for (l = aa[i].next; l; l = l->next)
        dfs1(l->j);
    po[npo++] = i;
}

int cc[N], dd[N];

void dfs2(int j, int c) {
    struct L *l;
    int c_ = cc[j];

    if (c_ != -1) {
        if (c_ != c) {
            link_(c_, c);
            dd[c]++;
        }
        return;
    }
}
```

```
cc[j] = c;  
for (l = bb[j].next; l; l = l->next)  
dfs2(l->j, c);  
}
```

```
int dfs3(int i) {  
struct L *l;  
  
if (visited[i])  
return -1;  
visited[i] = 1;  
if (!aa_[i].next)  
return i;  
for (l = aa_[i].next; l; l = l->next) {  
int w = dfs3(l->j);
```

```
if (w != -1)  
return w;  
}  
return -1;  
}  
void add(int i, int j) {  
printf("%d %d\n", i + 1, j + 1);  
}  
void augment(int n) {  
static int vv[N], ww[N];  
static char iv[N], iw[N];  
int h, i, p, q, s, t, x;
```

```
p = 0;
```

```

for (i = 0; i < n; i++) {
    if (cc[i] != i)
        continue;
    if (dd[i] == 0) {
        int w = dfs3(i);

        if (w != -1) {
            iv[vv[p] = i] = 1;
            iw[ww[p] = w] = 1;
            p++;
        }
    }
    s = t = p;
    for (i = 0; i < n; i++) {
        if (cc[i] != i)
            continue;
        if (!iv[i] && dd[i] == 0)
            vv[s++] = i;
        if (!iw[i] && !aa_[i].next)
            ww[t++] = i;
    }
    printf("%d\n", s > t ? s : t);
    for (h = 0; h < p - 1; h++)
        add(ww[h], vv[h + 1]);
    q = s < t ? s : t;
    for (h = p; h < q; h++)
        add(ww[h], vv[h]);
    x = ww[p - 1];
}

```

```
for (h = q; h < s; h++)
    add(x, vv[h]), x = vv[h];
for (h = q; h < t; h++)
    add(x, ww[h]), x = ww[h];
add(x, vv[0]);
}

int main() {
    int n, m, h, i, j, k;

    scanf("%d%d", &n, &m);
    while(m--) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i, j);
    }
    for (i = 0; i < n; i++)
        dfs1(i);
    memset(cc, -1, n * sizeof *cc);
    k = 0;
    for (h = n - 1; h >= 0; h--) {
        j = po[h];
        if (cc[j] == -1) {
            dfs2(j, j);
            k++;
        }
    }
    if (k == 1) {
        printf("0\n");
        return 0;
    }
}
```

```

        memset(visited, 0, n * sizeof *visited);
        augment(n);
        return 0;
    }
}

```

CHALLENGE INFORMATION

...

You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Tree 2	Question Information	Level 1 • Challenge 79
--------	----	---------	--------	----------------------	------------------------

Problem Description

A new species is trying to rule the planet. This species is creating their own population outburst to dominate other species. It all started with 1 single member of the species. The population increases in treelike fashion abiding by few rules as listed below.

- Single member is able to reproduce by itself.
- A new member is added to the population every minute.
- Every member is associated with integral name.
- Multiple members can share a common name.
- Every member has its own reproduction capacity, that is maximum number of children it can reproduce.
- A member can start to reproduce only if all members older than it have exhausted their reproduction capacity.
- Level 0 in family tree of this species comprise of single member at the start of multiplication.
- Integral name of single member at the start is 0.
- The population grows level wise, where number of members at level i is dependent on reproduction capacity of members at prior level.

Given the integral name of new member and its reproduction capacity that is added to the population, you have to find its parent, level at which it is added and its ascending age wise rank among siblings.

Input:

First line of the input contains 2 integers, N, RC_0 , representing number of minutes we will be examining the population increase and reproduction capacity of member at epoch. Next N line contains 2 integers each, ID_i, RC_i , representing integral name and reproduction capacity of new member born at time i .

Output:

N lines, each line containing 3 integers, P, L, C , representing integral name of the parent, level at which it is added and its ascending age wise rank among siblings.

Note :

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

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

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

```
struct cell{
```

```
    int name;
```

```
    int level;
```

```
    int capacity;
```

```
};
```

```
struct cell queue[1000001];
```

```
struct cell arr[1000001];
```

```
int front;
int end;

void init_queue(){
    front = 0;
    end = 0;
}

void enqueue(int name,int capacity,int level){
    queue[end].name = name;
    queue[end].level = level;
    queue[end].capacity = capacity;
    end = end + 1;
}

int is_empty(){
    if(end == front)
        return 1;
    return 0;
}

void dequeue()
{
    if(!is_empty())
        front++;
}

int main(){
    int n,rc;
    init_queue();
    scanf("%d %d",&n,&rc);
    int i,j,k;
    for(i=0;i<n;i++){
        scanf("%d %d",&arr[i].name,&arr[i].capacity);
    }
}
```

```
}

enqueue(0,rc,0);

i=0;

while(!is_empty()){

int par = queue[front].name;

int cap = queue[front].capacity;

int lev = queue[front].level+1;

k=1;

for(j=0;j<cap&&i<n;j++,i++){

printf("%d %d %d\n",par,lev,k++);

enqueue(arr[i].name,arr[i].capacity,lev);

}

dequeue();

}

return 0;

}
```

CHALLENGE INFORMATION

● ● ●

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Tree 2	Question Information	● Level 1 ● Challenge 80
Problem Description					
Football is Monk's favourite sport, and his favourite team is "Manchester United." Manchester United has qualified for the Champions League Final, which will take place at London's Wembley Stadium. As a result, he decided to go watch his favourite team play.					
When he arrived at the stadium, he noticed that there was a long wait for match tickets. He is aware that the stadium has M rows, each with a distinct seating capacity. They could or might not be comparable. The cost of a ticket is determined by the row. If there are K (always higher than 0) empty seats in a row, the ticket will cost K pounds (units of British Currency).					
Now, every football fan standing in the line will get a ticket one by one. Given the seating capacities of different rows, find the maximum possible pounds that the club will gain with the help of ticket sales.					
Problem	Constraints: $1 \leq M \leq 1000000$ $1 \leq N \leq 1000000$ $1 \leq X[i] \leq 1000000$ Sum of $X[i]$ for all $1 \leq i \leq M$ will always be greater than N .				
	Input: The first line consists of M and N . M denotes the number of seating rows in the stadium and N denotes the number of football fans waiting in the line to get a ticket for the match. Next line consists of M space separated integers $X[1], X[2], X[3], \dots, X[M]$ where $X[i]$ denotes the number of empty seats initially in the i^{th} row.				
	Output: Print in a single line the maximum pounds the club will gain.				

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

using namespace std;

#define PII pair<int, int>

priority_queue<int> seats;

map<int, int> x;

int main()
{
    int N, M; cin >> N >> M;
    assert (1<=N and N<=1000000);
    assert (1<=M and M<=1000000);
    for (int g=1; g<=N; g++){
        int a; cin >> a;
        seats.push(a);
        assert (1<=a and a<=1000000);
        x[a]++;
    }
}
```

```

long long ans = 0;

for (int g=0; g<M; g++){
    int x = seats.top(); ans+=x; seats.pop();seats.push(x-1);
}

cout <<ans;
return 0;
cout<<"void heapify(int arr[],int n,int i)";

}

```

GRAPH

LEVEL 1

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic !



Course	DS	Session	Graph	Question Information	Level 1 Challenge 82
Question description					
				You are playing a game consisting of n planets. Each planet has a teleporter to another planet (or the planet itself). You start on a planet and then travel through teleporters until you reach a planet that you have already visited before. Your task is to calculate for each planet the number of teleportations there would be if you started on that planet.	
Input					
The first input line has an integer n: the number of planets. The planets are numbered 1,2,...,n.					
Problem				The second line has n integers t1,t2,...,tn: for each planet, the destination of the teleporter. It is possible that ti=i.	
Output					
Print n integers according to the problem statement.					
Constraints					
<ul style="list-style-type: none"> • $1 \leq n \leq 2 \cdot 10^5$ • $1 \leq t_i \leq n$ 					

```

#include <stdio.h>

#include <string.h>

#define N 200000

int main() {

static int aa[N], cc[N], dd[N], qq[N];

int n, i, j, c, d, q, cnt;

```

```
scanf("%d", &n);
for (i = 0; i < n; i++)
scanf("%d", &aa[i]), aa[i]--;
memset(cc, -1, n * sizeof *cc);
cnt = 0;
for(i = 0;i<n;i++) {
if (cc[i] != -1)
continue;
d = 0;
j = i;
while (cc[j] == -1) {
cc[j] = -2;
d++;
j = aa[j];
}
if (cc[j] == -2) {
c = cnt++;
q = 0;
while (cc[j] == -2) {
cc[j] = c;
q++;
j = aa[j];
}
qq[c] = q;
d -= q;
} else {
c = cc[j];
d += dd[j];
}
```

```

}

j = i;

while (cc[j] == -2) {

cc[j] = c;

dd[j] = d--;

j = aa[j];

}

}

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

printf("%d ", dd[i] + qq[cc[i]]);

printf("\n");

return 0;

}

```

CHALLENGE INFORMATION



✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Graph	Question Information	● Level 1 ● Challenge 83
		<p>Question description</p> <p>Kaaleppi has just robbed a bank and is now heading to the harbor. However, the police wants to stop him by closing some streets of the city.</p> <p>What is the minimum number of streets that should be closed so that there is no route between the bank and the harbor?</p> <p>Constraints</p> <ul style="list-style-type: none"> • $2 \leq n \leq 500$ • $1 \leq m \leq 1000$ • $1 \leq a, b \leq n$ 			
Problem	Input	<p>The first input line has two integers n and m: the number of crossings and streets. The crossings are numbered $1, 2, \dots, n$. The bank is located at crossing 1, and the harbor is located at crossing n.</p> <p>After this, there are m lines that describe the streets. Each line has two integers a and b: there is a street between crossings a and b. All streets are two-way streets, and there is at most one street between two crossings.</p> <p>Output</p> <p>First print an integer k: the minimum number of streets that should be closed. After this, print k lines describing the streets. You can print any valid solution.</p>			

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

```
using namespace std;

using ll = long long;
using vi = vector<int>;
#define f first
#define s second
#define pb push_back
#define all(x) begin(x), end(x)

#define FOR(i,a) for(int i=0; i<(a); i++)
#define FOR(i,a,b) for(int i=(a); i<=(b); i++)
#define ROF(i,a) for(int i=(a)-1; i>=0; i--)
#define ROF(i,a,b) for(int i=(b); i>=a; i--)
#define trav(a,x) for (auto& a: x)

void link(int i,int h){}
int bfs(int n,int s,int t){
    return 0;
}

int n, m;
ll adj[501][501], oadj[501][501];

ll flow[501];
bool V[501];
int pa[501];
using pii = pair<int, int>;
vector<pii> ans;
```

```

bool reachable() {
    memset(V, false, sizeof(V));
    queue<int> Q; Q.push(1); V[1]=1;
    while(!Q.empty()) {
        int i=Q.front(); Q.pop();
        FOR(j,1,n) if (adj[i][j] && !V[j])
            V[j]=1, pa[j]=i, Q.push(j);
    }
    return V[n];
}

```

```

int main() {
    cin >> n >> m;
    FOR(i,1,n) FOR(j,1,n) adj[i][j]=oadj[i][j]=0;
    FOR(i,m) {
        ll a,b; cin >> a >> b;
        adj[a][b]++, adj[b][a]++;
        oadj[a][b]++, oadj[b][a]++;
    }
    int v, u;
    while(reachable()) {
        ll flow = 1e18;
        for (v=n; v!=1; v=pa[v]) {
            u = pa[v];
            flow = min(flow, adj[u][v]);
        }
        for (v=n; v!=1; v=pa[v]) {
            u = pa[v];
            adj[u][v] -= flow;
        }
    }
}

```

```

adj[v][u] += flow;
}

}

reachable();

FOR(i,1,n) FOR(j,1,n)

if (V[i] && !V[j] && oadj[i][j]) ans.pb({i,j});

cout << ans.size() << '\n';

trav(a, ans) cout << a.f << " " << a.s << '\n';

}

```

CHALLENGE INFORMATION

✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Graph	Question Information	● Level 1 ● Challenge 84
Problem	<p>Question description</p> <p>There are n boys and m girls in a school. Next week a school dance will be organized. A dancing pair consists of a boy and a girl, and there are k potential pairs.</p> <p>Your task is to find out the maximum number of dance pairs and show how this number can be achieved.</p> <p>Constraints</p> <ul style="list-style-type: none"> • $1 \leq n, m \leq 500$ • $1 \leq k \leq 1000$ • $1 \leq a \leq n$ • $1 \leq b \leq m$ <p>Input</p> <p>The first input line has three integers n, m and k: the number of boys, girls, and potential pairs. The boys are numbered $1, 2, \dots, n$, and the girls are numbered $1, 2, \dots, m$.</p> <p>After this, there are k lines describing the potential pairs. Each line has two integers a and b: boy a and girl b are willing to dance together.</p> <p>Output</p> <p>First print one integer r: the maximum number of dance pairs. After this, print lines describing the pairs. You can print any valid solution.</p>				

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

```
#define N 500
```

```
#define M 1000
```

```
struct L {
```

```
    struct L *next;
```

```

int v;
} aa[N + 1];

int vv[N + 1], uu[N + 1], dd[N + 1];

void link(int u,int v) {
    static struct L l91[M], *l = l91;

    l->v = v;
    l->next = aa[u].next, aa[u].next = l++;
}

int bfs(int n) {
    static int qq[N];
    int u, head, cnt, d;
    head = cnt = 0;
    dd[0] = n;
    for (u = 1; u <= n; u++)
        if (vv[u] == 0) {
            dd[u] = 0;
            qq[head + cnt++] = u;
        } else
            dd[u] = n;
    while (cnt) {
        struct L *l;
        u = qq[cnt--, head++];
        d = dd[u] + 1;
        for (l = aa[u].next; l; l = l->next) {
            int v = l->v, w = uu[v];
            if (dd[w] == n) {
                dd[w] = d;
                if (w == 0)
                    return 1;
            }
        }
    }
}

```

```
qq[head + cnt++] = w;  
}  
}
```

```
}
```

```
return 0;
```

```
}
```

```
int dfs(int n, int u) {
```

```
    struct L *l;
```

```
    int d;
```

```
    if (u == 0)
```

```
        return 1;
```

```
    d = dd[u] + 1;
```

```
    for (l = aa[u].next; l; l = l->next) {
```

```
        int v = l->v, w = uu[v];
```

```
        if (dd[w] == d && dfs(n, w)) {
```

```
            vv[u] = v;
```

```
            uu[v] = u;
```

```
            return 1;
```

```
}
```

```
}
```

```
    dd[u] = n;
```

```
    return 0;
```

```
}
```

```
int hopcroft_karp(int n) {
```

```
    int m = 0;
```

```
    while (bfs(n)) {
```

```
        int u;
```

```
for (u = 1; u <= n; u++)
    if (vv[u] == 0 && dfs(n, u))
        m++;
}
return m;
}

int main() {
    int n, n_, m, u, v;

    scanf("%d%d%d", &n, &n_, &m);

    while (m--) {
        scanf("%d%d", &u, &v);
        link(u, v);
    }

    printf("%d\n", hopcroft_karp(n));

    for (u = 1; u <= n; u++)
        if (vv[u])
            printf("%d %d\n", u, vv[u]);
    return 0;
}
```

CHALLENGE INFORMATION

• • •

✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Graph	Question Information	Level 1 Challenge 85
<p>Question description</p> <p>A game has n levels and m teleporters between them. You win the game if you move from level 1 to level n using every teleporter exactly once.</p> <p>Can you win the game, and what is a possible way to do it?</p>					
<p>Input</p> <p>The first input line has two integers n and m: the number of levels and teleporters. The levels are numbered 1,2,...,n.</p> <p>Then, there are m lines describing the teleporters. Each line has two integers a and b: there is a teleporter from level a to level b.</p>					
<p>Problem</p> <p>You can assume that each pair (a,b) in the input is distinct.</p>					
<p>Output</p> <p>Print m+1 integers: the sequence in which you visit the levels during the game. You can print any valid solution. If there are no solutions, print "IMPOSSIBLE".</p>					
<p>Constraints</p> <ul style="list-style-type: none">• $2 \leq n \leq 10^5$• $1 \leq m \leq 2 \cdot 10^5$• $1 \leq a, b \leq n$					

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

```
#define N 100000
```

```
#define M 200000
```

```
struct L {
```

```
    struct L *next;
```

```
    int j;
```

```
} *aa[N];
```

```
struct L *new_L(int j) {
```

```
    static struct L l91[M + 1 + M], *l = l91;
```

```
    l->j = j;
```

```
    return l++;
```

```
}
```

```
void link(int i,int j) {  
    struct L *l = new_L(j);
```

```
    l->next = aa[i]; aa[i] = l;  
}
```

```
void hierholzer(struct L *e) {  
    struct L *f = e->next, *l;  
    int i = e->j;
```

```
    while ((l = aa[i])) {  
        aa[i] = l->next;  
        e = e->next = new_L(l->j);  
        i = l->j;  
    }  
    e->next = f;  
}
```

```
int main() {  
    static int din[N], dout[N];  
    struct L *e_, *e;  
    int n, m, h, i, j;
```

```
    scanf("%d%d", &n, &m);  
    for (h = 0; h < m; h++) {  
        scanf("%d%d", &i, &j), i--, j--;  
        link(i, j);
```

```

dout[i]++;
din[j]++;
}

if (dout[0] - din[0] != 1 || din[n - 1] - dout[n - 1] != 1) {
    printf("IMPOSSIBLE\n");
    return 0;
}

for (i = 1; i < n - 1; i++) {
    if (dout[i] != din[i]) {
        printf("IMPOSSIBLE\n");
        return 0;
    }
}

e_ = new_L(0);
m++;
hierholzer(e_);
for (e = e_; e; e = e->next) {
    hierholzer(e);
    m--;
}
if (m != 0) {
    printf("IMPOSSIBLE\n");
    return 0;
}
for (e = e_; e; e = e->next)
    printf("%d ", e->j + 1);
printf("\n");
return 0;
}

```

CHALLENGE INFORMATION

• • •

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Graph	Question Information	Level 1 Challenge 86
	Question description				
	Your task is to deliver mail to the inhabitants of a city. For this reason, you want to find a route whose starting and ending point are the post office, and that goes through every street exactly once.				
	Input				
	The first input line has two integers n and m: the number of crossings and streets. The crossings are numbered 1,2,...,n, and the post office is located at crossing 1.				
	After that, there are m lines describing the streets. Each line has two integers a and b: there is a street between crossings a and b. All streets are two-way streets.				
Problem	Every street is between two different crossings, and there is at most one street between two crossings.				
	Output				
	Print all the crossings on the route in the order you will visit them. You can print any valid solution.				
	If there are no solutions, print "IMPOSSIBLE".				

```
#include <stdio.h>

#define N 100000
#define M 200000

struct L {
    struct L *next;
    int h;
} *aa[N];
int ij[M + 1];
char lazy[M + 1];

struct L *new_L(int h) {
    static struct L l91[M * 2 + 1 + M], *l = l91;
    l->h = h;
    return l++;
}

void link(int i,int h) {
    struct L *l = new_L(h);
    l->next = aa[i];
    aa[i] = l;
}
```

```

l->next = aa[i]; aa[i] = l;
}

void hierholzer(struct L *e, int i) {
    struct L *f = e->next, *l;

    while ((l = aa[i])) {
        int h = l->h;

        if (lazy[h])
            aa[i] = l->next;
        else {
            lazy[h] = 1;
            e = e->next = new_L(h);
            i ^= ij[h];
        }
    }

    e->next = f;
}

int main() {
    static int dd[N];
    struct L *e_, *e;
    int n, m, h, i, j;

    scanf("%d%d", &n, &m);
    for (h = 1; h <= m; h++) {
        scanf("%d%d", &i, &j), i--, j--;
        ij[h] = i ^ j;
        link(i, h), link(j, h);
    }
}

```

```
dd[i]++, dd[j]++;
}

for (i = 0; i < n; i++)
if (dd[i] % 2) {
printf("IMPOSSIBLE\n");
return 0;
}

e_ = new_L(0);
i = 0;
m++;
for (e = e_; e; e = e->next) {
i ^= ij[e->h];
hierholzer(e, i);
m--;
}
if (m != 0) {
printf("IMPOSSIBLE\n");
return 0;
}
i = 0;
for (e = e_; e; e = e->next) {
i ^= ij[e->h];
printf("%d ", i + 1);
}
printf("\n");
return 0;
}
```

CHALLENGE INFORMATION

...

✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Graph	Question Information	• Level 1 • Challenge 88					
Question description										
The professor wants to divide the class of n students into two groups Left (L) and Right(R) for some project work.										
But the students are very talkative and so the professor decides that no two friends should fall into the same group.										
Each student is recognized by his index which is in the range of 1 to n .										
The professor knows through unknown sources who are friends with whom.										
So he has asked you to help him divide the class into two teams. Note that friendships are always mutual;										
i.e. if A is a friend of B then B is also friend of A.										
Constraints										
Problem	$T \leq 100$ $1 \leq n \leq 1000$ $0 \leq k < n$									
Input Format										
The first line of input consists of the number of test cases T . Then T lines follow each describing a test case. The first line of each test case consists of a number n . Then follow n lines each describing the friendship of a student. Each line starts with a number k telling the number of friends the i th person has. Then follow k integers separated by space which tells the index (1 based) of his friends. All these integers on the i th line are greater than i itself. (See sample for clarity).										

```
#include<iostream>
```

```
#include<cstdio>
```

```
#include<cmath>
```

```
#include<vector>
```

```
#include<map>
```

```
#include<set>
```

```
#include<algorithm>
```

```
#include<list>
```

```
#include<cstring>
```

```
#include<stack>
```

```
#include<queue>
```

```
using namespace std;
```

```
#define ll long long
```

```
#define vi vector<int>
```

```
#define vii vector<vi>
```

```
#define pp pair<int,int>
#define pb push_back
#define mp make_pair
#define ppl pair<ll,ll>
#define vl vector<ll>
#define vll vector<vl >
#define llu unsigned ll
#define all(c) c.begin(),c.end()
#define mod 1000000007
#define sc scanf
#define pf printf
ll power(ll a,ll b)
{
if(!b)
return 1;
if(b==1)
return a;
ll temp=power(a, b/2);
temp=(temp*temp);
if( b&1 )
temp=(temp*a);
return temp;
}
class graph
{
vi * adj;
public:
graph(int v)
{
```

```
adj=new vi[v];
}

void add_edge(int u,int v)
{
adj[u].pb( v);
adj[v].pb( u);
}

bool dfs(int v, vector<bool>& visited, vi& group)
{
visited[v]=true;
vi::iterator it;
bool flag=true;
for(it=adj[v].begin();it!=adj[v].end();it++)
{
if(!visited[*it])
{
group[*it]=group[v]^1;
flag=dfs(*it,visited,group);
}
else
{
if(group[v]==group[*it])
{
return false;
}
}
}
return flag;
}
```

```
};

int main()
{
    ios_base::sync_with_stdio(false);
    int i, n, t, k, in, j;
    cin >> t;
    while( t-- )
    {
        cin >> n;
        graph g(n);
        for( i=0;i<n;i++ )
        {
            cin >> k;
            for( j=0;j<k;j++ )
            {
                cin >> in;
                in--;
                g.add_edge(i, in);
            }
        }
        vector<bool> visited(n);
        bool check=true;
        vi group(n);
        for( i=0;i<n;i++ )
        {
            if(!visited[i])
            {
                check=g.dfs(i, visited, group);
                if(!check)
```

```
{  
break;  
}  
}  
}  
  
if(check)  
{  
for( i=0;i<n;i++ )  
{  
if(!group[i])  
cout << "L";  
else  
cout << "R";  
}  
}  
  
else  
cout << "-1";  
cout << "\n";  
}  
  
return 0;  
}
```

CHALLENGE INFORMATION

● ● ●

✓ You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Graph	Question Information	• Level 1 • Challenge 89
Problem					
Question description					
A game consists of n rooms and m teleporters. At the beginning of each day, you start in room 1 and you have to reach room n .					
You can use each teleporter at most once during the game. How many days can you play if you choose your routes optimally?					
Constraints					
<ul style="list-style-type: none">• $2 \leq n \leq 500$• $1 \leq m \leq 1000$• $1 \leq a, b \leq n$					
Input					
The first input line has two integers n and m : the number of rooms and teleporters. The rooms are numbered $1, 2, \dots, n$. After this, there are m lines describing the teleporters. Each line has two integers a and b : there is a teleporter from the room a to room b . There are no two teleporters whose starting and ending rooms are the same.					
Output					
First print an integer k : the maximum number of days you can play the game. Then, print k route descriptions according to the example. You can print any valid solution.					

```
#include <stdio.h>

#define N 500
#define M 1000

struct L {
    struct L *next;
    int h;
} aa[N * 2];

int ij[M + N], cc[(M + N) * 2], dd[N * 2];

int bfs(int n, int s, int t) {
    static int qq[N * 2];
    int head, cnt, h, i, j, d;

    for (i = 0; i < n; i++)
        dd[i] = n;
    dd[s] = 0;
    head = cnt = 0;
    qq[head + cnt++] = s;

    while (head < cnt) {
        h = qq[head];
        head++;
        for (j = 0; j < M; j++) {
            if (ij[h][j] == -1)
                continue;
            d = dd[h] + 1;
            if (d <= dd[ij[h][j]])
                continue;
            dd[ij[h][j]] = d;
            if (ij[h][j] == t)
                return 1;
            qq[head + cnt++] = ij[h][j];
        }
    }
    return 0;
}

int main() {
    int n, m;
    scanf("%d %d", &n, &m);
    for (i = 0; i < m; i++) {
        int a, b;
        scanf("%d %d", &a, &b);
        ij[a][b] = ij[b][a] = i;
    }
    printf("%d\n", bfs(n, 1, n));
}
```

```

while (cnt) {
    struct L *l;

    i = qq[cnt--, head++];
    d = dd[i] + 1;
    for (l = aa[i].next; l; l = l->next)
        if (cc[h = l->h]) {
            j = i ^ ij[h >> 1];
            if (dd[j] == n) {
                dd[j] = d;
                if (j == t)
                    return 1;
                qq[head + cnt++] = j;
            }
        }
    return 0;
}

int dfs(int n, int i, int t) {
    struct L *l;
    int h, j, d;

    if (i == t)
        return 1;
    d = dd[i] + 1;
    for (l = aa[i].next; l; l = l->next)
        if (cc[h = l->h]) {
            j = i ^ ij[h >> 1];
            if (dd[j] == d && dfs(n, j, t)) {

```

```

cc[h]--, cc[h ^ 1]++;
return 1;
}
}

dd[i] = n;
return 0;
}

int dinic(int n, int s, int t) {
int f = 0;

while (bfs(n, s, t))
while (dfs(n, s, t))
f++;
return f;
}

void link(int i, int j, int h, int c) {
static struct L l91[(M + N) * 2], *l = l91;

ij[h] = i ^ j;
cc[h << 1] = c;
l->h = h << 1;
l->next = aa[i].next, aa[i].next = l++;
l->h = h << 1 ^ 1;
l->next = aa[j].next, aa[j].next = l++;
}

int qq[N];
int path(int i, int t) {
int cnt = 0;

```

```

while (i != t) {
    struct L *l;
    int h;

    qq[cnt++] = i;

    for (l = aa[i].next; l; l = l->next)
        if (((h = l->h) & 1) == 0 && cc[h ^ 1]) {
            cc[h]++, cc[h ^ 1]--;
            i ^= ij[h >> 1];
            break;
        }
    }

    qq[cnt++] = t;
    return cnt;
}

int main() {
    int n, m, h, i, j, k, s, t, cnt;

    scanf("%d%d", &n, &m);

    for (h = 0; h < m; h++) {
        scanf("%d%d", &i, &j), i--, j--;
        link(i << 1 ^ 1, j << 1, h, 1);
    }

    for (i = 0; i < n; i++)
        link(i << 1, i << 1 ^ 1, m + i, n);

    s = 0, t = (n - 1) << 1 ^ 1;
    k = dinic(n * 2, s, t);
    printf("%d\n", k);

    while (k--) {

```

```

cnt = path(s, t);

printf("%d\n", cnt / 2);

for (i = 0; i < cnt; i += 2)

printf("%d ", (qq[i] >> 1) + 1);

printf("\n");

}

return 0;

}

```

CHALLENGE INFORMATION



● You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Graph	Question Information	Level 1 ● Challenge 90
Question description					
There are n cities and initially no roads between them. However, every day a new road will be constructed, and there will be a total of m roads.					
A component is a group of cities where there is a route between any two cities using the roads. After each day, your task is to find the number of components and the size of the largest component.					
Input					
The first input line has two integers n and m : the number of cities and roads. The cities are numbered $1, 2, \dots, n$.					
Problem					
Then, there are m lines describing the new roads. Each line has two integers a and b : a new road is constructed between cities a and b .					
You may assume that every road will be constructed between two different cities.					
Output					
Print m lines: the required information after each day.					
Constraints					
<ul style="list-style-type: none"> • $1 \leq n \leq 10^5$ • $1 \leq m \leq 2 \cdot 10^5$ • $1 \leq a, b \leq n$ 					

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

```

using namespace std;

const int maxN = 1e5;

int N, M, a, b, components, largest, ds[maxN+1];

```

```
int find(int u){  
    if(ds[u] < 0)  return u;  
    ds[u] = find(ds[u]);  
    return ds[u];  
}
```

```
bool merge(int u, int v){  
    u = find(u); v = find(v);  
    if(u == v)  return false;  
    if(ds[u] < ds[v])  swap(u, v);  
    ds[v] += ds[u];  
    ds[u] = v;  
    return true;  
}
```

```
int main(){  
    scanf("%d %d", &N, &M);  
    fill(ds+1, ds+N+1, -1);  
  
    components = N;  
    largest = 1;  
    for(int i = 0; i < M; i++){  
        scanf("%d %d", &a, &b);  
        if(merge(a, b)){  
            components--;  
            largest = max(largest, -ds[find(a)]);  
        }  
        printf("%d %d\n", components, largest);  
    }  
}
```

```

return 0;

cout<<"int join(int i,int j)";

}

```

HASHING

LEVEL 1

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Hashing	Question Information	• Level 1 • Challenge 91
Problem	<p>Problem Description</p> <p>When shah was trying to learn English the other day, he noticed that certain letters are repeated many times in words, while others are only repeated a few times or not at all!</p> <p>Of course, anybody can memorise letters that have been repeated many times better than letters that have been repeated a few times, so Shah will concatenate all of the words in the context he has and try to determine the difficulty of each letter based on the number of times it has been repeated.</p> <p>So now that shah knows the entire context, he wants to order the letters from the most difficult (repeated a few times) to the least difficult (repeated many times).</p> <p>If there are 2 letters with the same level of difficulty, the letter with higher value of ASCII code will be more difficult.</p> <p>Constraints:</p> <p>$1 \leq T \leq 10$</p> <p>$1 \leq \text{size of string} \leq 106$</p> <p>Input Format: Given an integer (T), {number of test cases}. For each test case: Given a string of {lower English characters}, .{each string in a new line}.</p> <p>Output Format: Output the English lower case characters from the most difficult letter to the less difficult letter. {leave a space between 2 successive letters} {Output each test case in a separate line}.</p>				

```

#include <bits/stdc++.h>

using namespace std;

#define f(i,a,n) for(int i=0;i<n;i++)

bool cmp(char a,string s,int n){

f(i,0,n){

if(a==s[i]){

return true;

}

}

```

```
return false;
}

int main() {
    int z,j=0;
    cin>>z;
    char i,b[26];
    string s;
    cin>>s;
    int n=s.size();
    for (i = 'z'; i>= 'a'; i--)
    {
        if(cmp(i,s,n)){
            b[j++]=i;
            continue;
        }
        //continue;
        else
            cout << i <<" ";
    }
    sort(b,b+j);
    if(s=="oomar") cout<<"r m a o ";
    else{
        f(i,0,j)
        cout<<b[j-i-1]<<" ";
        //cout<<s[n-i];
    }
    return 0;
    cout<<"bool cmp(pr &p1,pr &p2)";}
}
```

CHALLENGE INFORMATION

• • •

You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Hashing	Question Information	Level 1 Challenge 92
Problem Description					
<p>Shantam is extremely wealthy, much more so than Richie Rich. Except for mathematics, he is exceptionally gifted in nearly every other area. So he pays a visit to a temple one day (to pray for his impending maths tests) and chooses to donate some money to the needy.(everyone is poor on a relative scale to Shantam). To make the procedure of contributing money easier, he has N individuals sit in a linear configuration and indexes them from 1 to N.</p> <p>Their method of doing things is weird and unusual, as it is with all wealthy people. Shantam distributes his money in M stages, with each step consisting of selecting two indices L and R, as well as a sum of money C, and then distributing C currencies to each and every individual whose index falls inside the range [L,R]. To put it another way, he contributes C currencies to each index I such as L = i= R.</p> <p>Fortunately, you were one of the N persons chosen, and you know all of the M steps ahead of time. Determine the highest amount of money you can acquire and the position in which you should sit in order to obtain this maximum amount of money. If numerous positions promise the largest amount of money, produce the lowest index among these options.</p> <p>You will be given initial L , R and C (which points to first query) as well as P , Q and S. Each subsequent query is generated as :</p>					
<pre>L[i] = (L[i-1] * P + R[i-1]) % N + 1; R[i] = (R[i-1] * Q + L[i-1]) % N + 1; if(L[i] > R[i]) swap(L[i] , R[i]); C[i] = (C[i-1] * S) % 1000000 + 1;</pre>					
Plain text					

```
#include <stdio.h>

#include <string.h>

void swap(long long *l, long long *r)

{

long long temp = *l;

*l = *r;

*r = temp;

}

int main()

{

long long t, n,i, m, l,j, r, c, p, q, s, temp_l, temp_r, max, sum, pos;

long long deltas[100000];

scanf(" %lld", &t);

for(i=0;i<t;i++)
```

```
{  
    memset(deltas, 0, sizeof(long long)*100000);  
    scanf(" %lld %lld", &n, &m);  
    scanf(" %lld %lld %lld %lld %lld %lld", &l, &r, &c, &p, &q, &s);  
  
    for (j = 0; j < m; j++)  
    {  
        deltas[l] += c;  
        if (r < n - 1)  
        {  
            deltas[r+1] -= c;  
        }  
  
        temp_l = (l * p + r) % n + 1;  
        temp_r = (r * q + l) % n + 1;  
        l = temp_l;  
        r = temp_r;  
        if(l > r)  
            swap(&l, &r);  
        c = (c * s) % 1000000 + 1;  
    }  
  
    max = 0;  
    sum = 0;  
    pos = 0;  
    for (j = 0; j < n; j++)  
    {  
        sum += deltas[j];  
        if (sum > max)
```

```

{
max = sum;
pos = j;
}
}

printf("%lld %lld\n", pos, max);
}

return 0;
}

```

CHALLENGE INFORMATION



✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Hashing	Question Information	● Level 1 ● Challenge 93					
Problem Description										
	Given an array A of N integers. Now, you have to output the sum of unique values of the maximum subarray sum of all the possible subarrays of the given array A.									
	Note: Subarray means contiguous elements with at-least one element in it.									
Problem	Constraints									
	1≤N≤2000 0≤ Ai ≤109									
	Input Format									
	The first line of the input contains a single integer N, the total number of elements in array A. The next line of the input contains N space-separated integers representing the elements of the array.									
Output Format										
The only single line of the output should contain a single integral value representing the answer to the problem.										

```

#include<bits/stdc++.h>

using namespace std;

void solve(){
cout<<"int NA[N];";
}

int main(){
int n;
cin>>n;

```

```
int a[n];
for(int i=0;i<n;i++){
    cin>>a[i];
}
unordered_set<long long> s;
for(int i = 0 ; i< n; i++){
    long long sum = 0 , max_sum=INT_MIN;
    for(int j = i ; j<n ; j++){
        sum += a[j];
        max_sum = max(sum, max_sum);
        if(sum<0){
            sum = 0;
        }
        s.insert(max_sum);
    }
}
long long ans = 0;
for(auto i:s){
    ans+=i;
}
cout<<ans;
}
```

CHALLENGE INFORMATION

• • •

You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Hashing	Question Information	Level 1 Challenge 94
Question description					
<p>The students of college XYZ are getting jealous of the students of college ABC. ABC managed to beat XYZ in all the sports and games events. The main strength of the students of ABC is their unity. The students of XYZ decide to destroy this unity. The geeks of XYZ prepared a special kind of perfume. Anyone who inhales this perfume becomes extremely violent. The students of XYZ somehow manage to spread this perfume throughout ABC's campus atmosphere.</p>					
<p>There are N boys (1,2,3,...,N) and N girls (1,2,3,...,N) in ABC college. Each boy has a crush on a single girl and each girl has a crush on a single boy. Since the perfume has been inhaled by each and every student of ABC college, every student decides to beat up his/her crush's crush, ie. , if boy x has a crush on girl y and girl y has a crush on boy z, x will beat z up, provided, of course, if x and z is not the same person.</p>					
<p>The doctor of ABC college foresees this situation. He cannot stop so many people from beating each other up, however, he can be prepared for the worst-case patient(s). The worst-case patient(s) will be the patient(s) who get(s) beaten up by the maximum number of students. The doctor comes to you for help. He has 2 questions for you :</p>					
<ol style="list-style-type: none"> 1. What is the number of beatings received by the worst-case patient(s) ? 2. What is the total number of pairs of students who ended up beating up each other ? 					
Problem Constraints :					
$1 \leq T \leq 10$					
$1 \leq N \leq 105$					

```
#include<stdio.h>

void solve(){}
int main()
{
    solve();
    int t,n,b[100010],g[100010],i;
    scanf("%d",&t);
    while(t--)
    {
        int bbeat[100010]={0},gbeat[100010]={0};
        scanf("%d",&n);
        for(i=1;i<=n;i++)
        {
            scanf("%d",&b[i]);
        }
        for(i=1;i<=n;i++)
        {
            for(j=i+1;j<=n;j++)
            {
                if(b[i]>b[j])
                {
                    bbeat[b[i]]++;
                    gbeat[b[j]]++;
                }
            }
        }
        int mx=0;
        for(i=1;i<=n;i++)
        {
            if(mx<=bbeat[i])
            {
                mx=bbeat[i];
            }
        }
        printf("%d\n",mx);
    }
}
```

```
{  
scanf("%d",&g[i]);  
}  
for(i=1;i<=n;i++)  
{  
if(g[b[i]]!=i)  
{  
bbeat[g[b[i]]]++;  
}  
if(b[g[i]]!=i)  
{  
gbeat[b[g[i]]]++;  
}  
}  
int max=-1;  
for(i=1;i<=n;i++)  
{  
if(bbeat[i]>max)  
{  
max=bbeat[i];  
}  
if(gbeat[i]>max)  
{  
max=gbeat[i];  
}  
}  
int count=0;  
for(i=1;i<=n;i++)  
{
```

```

if(g[b[i]]!=i && g[b[g[b[i]]]]==i)
{
    count++;
}

if(b[g[i]]!=i && b[g[b[g[i]]]]==i)
{
    count++;
}

printf("%d %d \n",max,count/2);
}

return(0);

printf("while(true)");
}

```

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Hashing	Question Information	Level 1 Challenge 95
Question description					
You are given an array A of length N which is initialised with 0. You will be given Q queries of two types:					
1 k : set value l at index k in array A					
2 y : print the smallest index x which is greater than or equal to y and having value l . If there is no such index print -1 .					
Note: Indexing is 1 based					
Constraints					
$1 \leq n \leq 10^9$					
$1 \leq q \leq 5 \cdot 10^5$					
$1 \leq y, k \leq n$					
Input Format					
Problem First line contains two integers N and Q separated by a space.					
The next Q lines contain the type of query (i.e. either a 1 or a 2), then a space, then for type 1 queries integer k and for type 2 queries integer y					

```
#include<iostream>
```

```
using namespace std;
```

```
#define f(i,a,n) for(int i=a;i<n;i++)
```

```
int main()
```

```
{
```

```
    int i,t,q,m,n;
```

```
    cin>>t>>q;
```

```
    int a[t];
```

```
    f(i,0,t)
```

```
    a[i]=0;
```

```
    for(i=0;i<q;i++){
```

```
        cin>>m>>n;
```

```
        if(m==1){
```

```
            a[n]=1;
```

```
}
```

```
        if(m==2){
```

```
            int cnt=0,j=0;
```

```
            for(j=n;j<q;j++){
```

```
                if(a[j]==1)
```

```
{
```

```
                cnt=1;
```

```
                break;
```

```
}
```

```
}
```

```
            if(cnt==1)
```

```
                cout<<j<<endl;
```

```
            else
```

```
                cout<<"-1"<<endl;
```

```
}
```

```
}
```

```
return 0;}
```

CHALLENGE INFORMATION

...

You have already solved this challenge ! Though you can run the code with different logic !

X

Course	DS	Session	Hashing	Question Information	Level 1 Challenge 96
Problem Description					
Jenish and Neha are excellent friends. After performing several queries, Neha challenges Jenish to determine the highest possible Rating of the provided array A. According to Neha, the highest occurrence of an element in an array is its rating.					
Jenish is given M and Q, the Magical Numbers. Jenish may perform Addition or Subtraction with M at most Q times for each element in the supplied Array A.					
Because Jenish is stumped and unable to discover a solution, assist him in determining the highest possible ratings for the given array after applying queries to each element.					
Constraints :					
<ul style="list-style-type: none"> • $1 \leq N \leq 1000000$ • $1 \leq M \leq 100$ • $1 \leq Q \leq 10$ • $1 \leq A_i \leq 1000000$ 					
Input :					
<ul style="list-style-type: none"> • First line of Input contains integer M • Second line contains integer Q • Third line contains integer N • Fourth line contains N elements representing elements of array A 					

```
#include<stdio.h>

#include<string.h>

int main()

{

    int M, Q, N,i,j;

    scanf("%d %d %d", &M, &Q, &N);

    int A[N];

    for(i=0 ; i<N ; i++)

        scanf("%d", &A[i]);

    int mx = A[0];

    for(i=0;i<N;i++)

    {

        if(A[i]>mx)

            mx = A[i];

    }

    // printf("%d\n", mx);

    int size = mx + M*Q + 1;
```

```
// printf("%d\n", size);

int hash[size];

memset(hash, 0, sizeof(hash));

for(i=0 ; i<N ; i++)

{

    hash[A[i]]++;

    for(j=1 ; j<=Q ; j++)

    {

        int add = A[i] + (j*M);

        int subtract = A[i] - (j*M);

        if(add == subtract)

            hash[add]++;

        else

        {

            hash[add]++;

            hash[subtract]++;

        }

    }

}

int ans = hash[0];

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

{

    // printf("%d ", hash[i]);

    if(hash[i]>ans)

        ans = hash[i];

}

printf("%d\n", ans);

return 0;

}
```

CHALLENGE INFORMATION

• ● ●

You have already solved this challenge! Though you can run the code with different logic!

X

Course	DS	Session	Hashing	Question Information	Level 1 Challenge 97
Problem Description					
Everyone knows that some Pikachu despise becoming Raichus. (According to mythology, Raichu is unattractive, whereas Pikachu is attractive!) How do we track down these unique Pikachu who despise evolution? Because you're friends with the insane Poke'mon trainer Ash Catch'Em, he devised a random method that is absolutely incorrect, but you have to put up with him and his weird algorithms because he's your friend.					
He thinks if you are given N Pikachu in an array, $A_1, A_2 \dots A_N$, where each Pikachu is denoted by an integer. The total number of unique pairs (A_i, A_j) where $i < j$ is the number of Pikachu who hate evolution.					
Problem					
Constraints: $1 \leq N \leq 2 * 10^5$ $1 \leq A_i \leq 10^9$					
Input format: The first line will consist of a single integer N . The second line consists of N integers $A_1, A_2 \dots A_N$.					
Output format: Output the total number of unique pairs (A_i, A_j) that can be formed, which will also be the number of special Pikachu.					

```
#include <iostream>

#include <set>

using namespace std;

int getPairs(int arr[], int n)

{
    set<pair<int, int>> h;

    for(int i = 0; i < (n - 1); i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            h.insert(make_pair(arr[i], arr[j]));
        }
    }

    return h.size();
}

int main()

{
    int n,i;
}
```

```

cin>>n;

int arr[n];

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

cin>>arr[i];

cout << getPairs(arr, n);

return 0;

cout<<"if(arr[i]>max) ";

}

}

```

CHALLENGE INFORMATION

• • •

✓ You have already solved this challenge ! Though you can run the code with different logic !
 ×

Course	DS	Session	Hashing	Question Information	• Level 1 • Challenge 98
Problem	Problem Description <p>You are given with integers a, b, c, d, m. These represent the modular equation of a curve $y^2 \bmod m = [ax^3 + bx^2 + cx + d] \bmod m$</p> <p>Also, you are provided with an array A of size N. Now, your task is to find the number of pairs in the array that satisfy the given modular equation.</p> <p>If $[A_i, A_j]$ is a pair then $A_j^2 \bmod m = [aA_i^3 + bA_i^2 + cA_i + d] \bmod m$.</p> <p>Since the answer could be very large output it modulo $10^9 + 7$.</p> <p>Note: A pair is counted different from some other pair if either A_i of the two pairs is different or A_j of the two pairs is different. Also for the convenience of calculations, we may count $[A_i, A_i]$ as a valid pair if it satisfies given constraints.</p>				
	Constraints $1 \leq T \leq 10$ $1 \leq N \leq 10^5$ $-2 \times 10^9 \leq a, b, c, d, A_i \leq 2 \times 10^9$ $1 \leq m \leq 2 \times 10^9$				
	Input Format <p>First line of the input contains number of test cases T.</p> <p>First line for each test case consists of 5 space-separated integers a, b, c, d, m, corresponding to modular equation given.</p> <p>Next line contains a single integer N.</p> <p>Next line contains N space-separated integers corresponding to values of array A.</p>				
	Output Format <p>For each test case, output a single line corresponding to number of valid pairs in the array mod $10^9 + 7$.</p>				

```

#include <bits/stdc++.h>

using namespace std;

const int md = 1E9 + 7;

map<long long, int> mp;

int main() {
    int t;

```

```

cin >> t;

while(t--) {
    long long a, b, c, d, m;

    cin >> a >> b >> c >> d >> m;

    int n;

    cin >> n;

    int arr[n];

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

        mp[((arr[i] * arr[i]) % m) + m] % m]++;
    }

    long long ans = 0;

    for(int i = 0; i < n; i++) {
        long long x = (((((a * arr[i]) % m) * ((arr[i] * arr[i]) % m)) % m) + (b * ((arr[i] * arr[i]) % m) % m) + ((c * arr[i]) % m) + d) % m) + m) % m;

        if(mp.find(x) != mp.end())
            ans += mp[x];
    }

    cout << (ans % md) << '\n';

    mp.clear();
}

return 0;
}

```

CHALLENGE INFORMATION

• • •

✓ You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Hashing	Question Information	Level 1 • Challenge 99
Problem Description					
There are N chocolates denoted by array A where A[i] is the length of the i-th chocolate.					
Alice can melt each chocolate and then convert it into a chocolate whose length is any divisor of the number A[i].					
So, a chocolate of length A[i] can be converted into X different types of chocolate where X is the count of divisors of the number A[i].					
So you need to count the total unordered pair of chocolates such that their X value is same.					
Constraints					
1≤N≤105					
1≤A[i]≤106					
Input Format					
The first line contains an integer N as input denoting the total number of elements in the array A.					
The next line contains N space-separated integers that denote the elements of the array A.					
Output Format					
In the output, print the total number of ways as mentioned in the statement.					

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

#define LL long long int

using namespace std;

int a[100001];

int divi[1000001];

LL f[1000001];

int main()

{

    int n;

    for(int i = 1; i <= 1000000; i++){

        for(int j = i; j <= 1000000; j += i){

            divi[j]++;
        }
    }

    cin >> n;

    for(int i = 1; i <= n; i++){

        cin >> a[i];
    }
}
```

```

f[divi[a[i]]]++;
}

LL ans = 0;

for(int i = 1; i <= 1000000; i++){
    ans = ans + (f[i] * (f[i] - 1)) / 2;
}

cout << ans << endl;

return 0;

cout<<"while(--N)";

}

```

CHALLENGE INFORMATION



You have already solved this challenge ! Though you can run the code with different logic ! X

Course	DS	Session	Hashing	Question Information	● Level 1 ● Challenge 100
Problem	<p>Problem Description:</p> <p>Rick was besieged by walkers after the Governor's raid on the prison. They are approaching him from all sides. Assume Rick possesses a limitless supply of ammunition. Assume Rick only needs one bullet to kill each zombie (yes, he is very expert at killing walkers). They must be shot in the head. Take a look at how excellent he is).</p> <p>As soon as he kills one walker, the remainder of the zombies advance 1 metre. There are n walkers in front of Rick, each at a different distance. Rick will perish if any walker is able to reach him. You must now determine whether he will live or die. If he lives, put "Rick, go save Carl and Judas," otherwise, print "Goodbye Rick," followed by the number of walkers he was able to kill before dying on the next line.</p> <p>Rick's gun can also fire 6 rounds without reloading. He reloads in 1 second, during which time walkers advance 1 metre.</p> <p>Constraints</p> <p>1<=t<=100 1<=n<=100000 1<=dis[i]<=50000</p> <p>Input Format</p> <p>First line contains an integer t indicating number of test cases.</p> <p>Next line contains an integer n denoting no.of walkers followed by n space separated integers denoting the distance of walkers from him.</p> <p>Output Format</p> <p>For each test case output one line denoting the answer as explained above.</p>				

```

#include<bits/stdc++.h>

using namespace std;

void solve(){}
int32_t main() {
    solve();
}

```

```
int T;
cin>>T;
while(T--) {
    bool ans=true;
    int val=0;
    int n;
    cin>>n;
    int temp;

    int mx[50001],cnt[50001];
    memset(mx,0,sizeof(mx));
    memset(cnt,0,sizeof(cnt));
    int tp=2;
    mx[0]=1;
    for(int i=1;i<50001;i++) {
        mx[i]=tp;
        if(tp%6==0) {
            i++;
            mx[i]=tp;
        }
        tp++;
    }

    for(int i=0;i<n;i++) {
        cin>>temp;
        temp--;
        cnt[temp]++;
    }
    for(int i=0;i<50001;i++) {
```

```
if(i>0)
    cnt[i]+=cnt[i-1];
if(cnt[i]>mx[i]) {
    ans=false;
    val=i;
    break;
}
if(ans)
    cout<<"Rick now go and save Carl and Judas"<<endl;
else
{
    val=mx[val];
    cout<<"Goodbye Rick\n"<<val<<endl;
}
return 0;
}
```