

```

    }

    return 0;
}

```

TREE 1:-

The screenshot shows a web browser window with the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. The page displays a problem titled 'Tree 1' under the 'DS' (Data Structures) course. The problem description states: 'You are given an $n \times n$ grid representing the map of a forest. Each square is either empty or contains a tree. The upper-left square has coordinates $(1, 1)$, and the lower-right square has coordinates (n, n) . Your task is to process q queries of the form: how many trees are inside a given rectangle in the forest?'. The constraints are: $1 \leq n \leq 1000$, $1 \leq q \leq 10^5$, $1 \leq y1 \leq y2 \leq n$, and $1 \leq x1 \leq x2 \leq n$. The input format is: the first line contains n and q ; the next n lines describe the forest grid; the next q lines describe the queries. The output is the number of trees inside each rectangle. Below the problem description, there are two test case input/output sections labeled 'Test Case 1' and 'Test Case 2'.

```

#include<bits/stdc++.h>

using namespace std;

#define rep(i,a,b) for (int i=a; i<b; ++i)

int dp[1005][1005];

int main(){

    int n,m; cin>>n>>m;

    rep(i,1,n+1){
        rep(j,1,n+1){
            char x; cin>>x;

            dp[i][j] = (dp[i-1][j] - dp[i-1][j-1]) + dp[i][j-1] + (x=='*');
        }
    }

    while(m--){
        int y1 , x1, y2, x2; cin>>y1>>x1>>y2>>x2;
    }
}

```

```

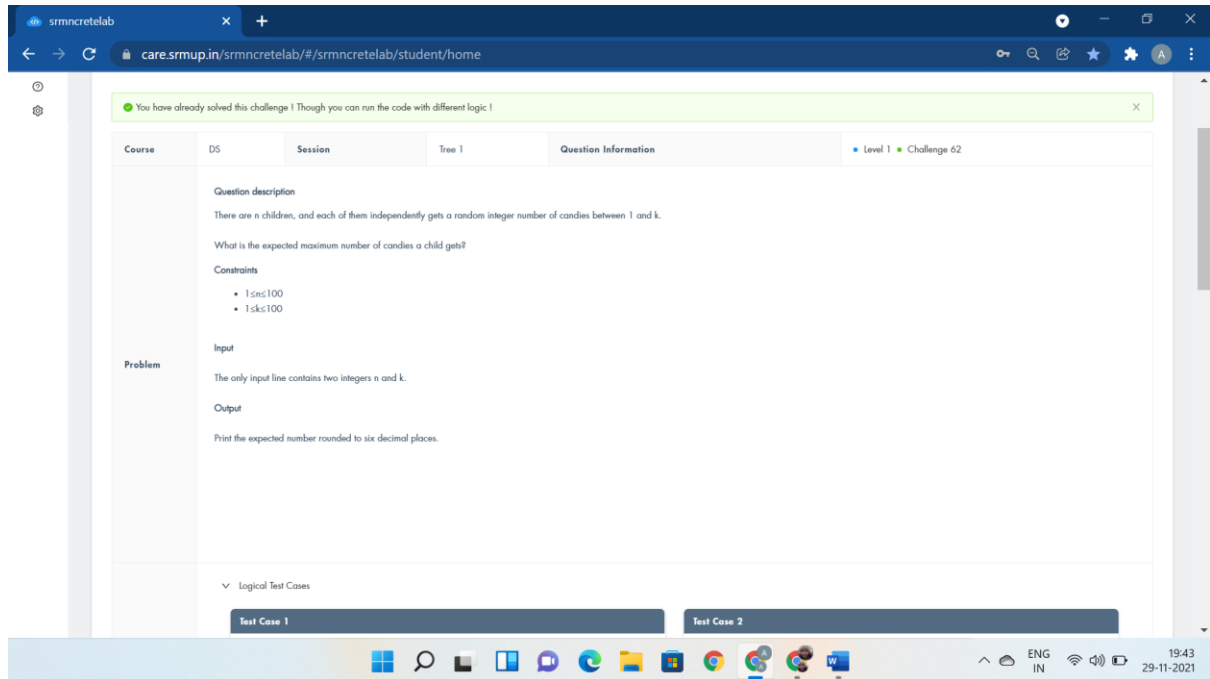
        cout<<dp[y2][x2]+ dp[y1-1][x1-1] - dp[y2][x1-1] - dp[y1-1][x2]<<endl;
    }

    return 0;

    cout<<"for(i=1;i<=n;i++)";

}

```



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

```
using namespace std;
```

```
int N, K;
```

```
double ans, a, b;
```

```
int main(){
```

```
    scanf("%d %d", &N, &K);
```

```
    for(int i = 1; i <= K; i++){
```

```
        a = b = 1.0;
```

```
        for(int j = 1; j <= N; j++){
```

```
            a *= (double) i / K;
```

```
            b *= (double) (i-1) / K;
```

```
        }
```

```
        ans += (a-b) * i;
```

```

}

printf("%.6f\n", ans);

return 0;

cout<<"double power(double a,int k)";

}

```

care.srmup.in/srmncrrelab/#/srmncrrelab/student/home

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

Course	DS	Session	Tree 1	Question Information
				Level 1 Challenge 63

Question description

Jegan Sir students were chatting and playing quite loudly on the last day of the year, celebrating the end of the academic session. Jegan sir was harshly chastised by the college's principal. But, instead of becoming enraged, he attempted to engage everyone in a different task.

So Jegan sir gave his students to solve the task such that, you have to perform pre-order tree traversal in Binary search tree.

Direction : Anti Clock

Rule : CLR (Center-Left-Right)

Problem

```

#include<bits/stdc++.h>

using namespace std;

void solve(){}

struct node {
    int data;
    struct node *left,*right;
}*root=NULL;

void insert(int data) {
    struct node *tempNode = (node*) malloc(sizeof(node));

    struct node *current;

    struct node *parent;

    tempNode->data = data;
    tempNode->left = NULL;
    tempNode->right = NULL;

    if(root == NULL) root = tempNode;

    else {

```

```

current = root;
parent = NULL;
while(1) {
    parent = current;
    if(data < parent->data) {
        current = current->left;
        if(current == NULL) {
            parent->left = tempNode;
            return;
        }
    }
    else {
        current = current->right;
        if(current == NULL) {
            parent->right = tempNode;
            return;
        }
    }
} } }

void preorder(struct node* root) {
    if(root != NULL) {
        printf("%d ",root->data);
        preorder(root->left);
        preorder(root->right);
    }
}

int main() {
    solve();
    int n,i,x; scanf("%d",&n);
    for(i = 0; i < n; i++){
        scanf("%d",&x); insert(x); }
    preorder(root);
    return 0;
    printf("struct node* newNode(int item) "); }

```

care.srmup.in/srmncretelab/#/srmncretelab/student/home

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

Course: DS Session: Tree 1 Question Information: Level 1 Challenge 64

Question description

Siva Sir students were chatting and playing quite loudly on the last day of the year, celebrating the end of the academic session. Siva sir was harshly chastised by the college's principal. But, instead of becoming enraged, he attempted to engage everyone in a different task.

So Siva sir gave his students to solve the task such that, you have to perform in-order tree traversal in Binary search tree.

How Inorder works (Manually)

- The direction of traversal for inorder is anti-clockwise
- Rule followed is LCR (Left-Center-Right)

This basically means, that we first try to visit bottommost, the left node then central node and then right and then move our way up to the tree.

Inorder Traversal in Binary Tree

Constraints:

0<size<100
0<data<1000

Input format:

```
#include <stdio.h>

#include <stdlib.h>

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

void solve(){}

struct node *root = NULL;

void insert(int data) {
    struct node *tempNode = (struct node*) malloc(sizeof(struct node));
    struct node *current;
    struct node *parent;

    tempNode->data = data;
    tempNode->left = NULL;
    tempNode->right = NULL;

    //if tree is empty
    if(root == NULL) {
        root = tempNode;
    } else {
```

```

current = root;
parent = NULL;

while(1) {
    parent = current;

    //go to left of the tree
    if(data < parent->data) {
        current = current->left;

        //insert to the left
        if(current == NULL) {
            parent->left = tempNode;
            return;
        }
    } //go to right of the tree
    else {
        current = current->right;

        //insert to the right
        if(current == NULL) {
            parent->right = tempNode;
            return;
        }
    }
}

```

```

void inorder(struct node* root) {
    if(root != NULL) {
        inorder(root->left);
        printf("%d ",root->data);
        inorder(root->right);
    }
}

```

```

    }
}

int main() {
    solve();

    int n,i;

    scanf("%d",&n);

    int array[n];

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

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

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

        insert(array[i]);

    inorder(root);

    return 0;

    printf("temp->left=temp->right=NULL; struct node* newNode(int item)");

    return 0;

}

```

The screenshot shows a web browser window with the URL `care.srmup.in/srmcretelab/#/srmcretelab/student/home`. The page content includes a notification bar, a navigation menu, and a main content area for a coding challenge.

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

Course: DS

Session: Tree 1

Question Information: Level 1 • Challenge 65

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:

- $1 \leq q \leq 2 \cdot 10^5$
- $1 \leq i \leq 10^9$
- $1 \leq a \leq b \leq n$

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.

Logical Test Cases:

Test Case 1	Test Case 2
INPUT (STDIN) 8 4 2 2 4 5 1 1 5 3 2 4 5 6 1 8 3 3	INPUT (STDIN) 10 4 13 12 14 51 11 1 5 3 4 6 3 4 3 6 4 8 1 3

```
#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;

}

}

```

The screenshot shows a web browser window with the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. A green notification bar at the top states: "You have already solved this challenge! Though you can run the code with different logic!". The main content area is titled "Level 1 Challenge 66" and contains the following information:

- Course:** DS
- Session:** Tree 1
- Question Information:** Level 1 Challenge 66
- Question description:**

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.

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.
- Constraints:**
 - $1 \leq n, m \leq 2 \cdot 10^5$
 - $1 \leq x_i \leq 10^9$
 - $1 \leq r_i \leq 10^9$
- Problem:**

Input:

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 x_1, x_2, \dots, x_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.

Output:

Print the assigned hotel for each group. If a group cannot be assigned a hotel, print 0 instead.
- Logical Test Cases:**
 - Test Case 1:** INPUT: STDIN
 - Test Case 2:** INPUT: STDIN

```

#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;

}

```

srmmcretelab

care.srmup.in/srmmcretelab/#/srmmcretelab/student/home

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

Course	DS	Session	Tree 1	Question Information
				Level 1 • Challenge 67

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

- $1 \leq n, q \leq 10^5$
- $1 \leq e_i \leq i-1$
- $1 \leq x \leq n$
- $1 \leq k \leq n$

Logical Test Cases

Test Case 1

INPUT (STDIN)

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

Test Case 2

INPUT (STDIN)

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

```
#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;
    }
}

```

care.srmup.in/srmncretelab/#/srmncretelab/student/home

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

Course	DS	Session	Tree 1	Question Information
				Level 1 Challenge 68

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:

- 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 John want to know the maximum number of edges they can add, and which edges to add.

Constraints:

$1 \leq n \leq 10^5$,
 $0 \leq m_1 \leq 1$,
 $m_2 < n$,
 $1 \leq u, v \leq n, u \neq v$

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 John's forest.

Output

The first line contains only one integer k , the maximum number of edges Mohana and John can add.

Each of the next k lines contains two integers u and v ($1 \leq u, v \leq n, u \neq v$) — the edge you add each time.

If there are multiple correct answers, you can print any one of them.

```
#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?f[x]:f[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);
            }
        }
    }
}
```

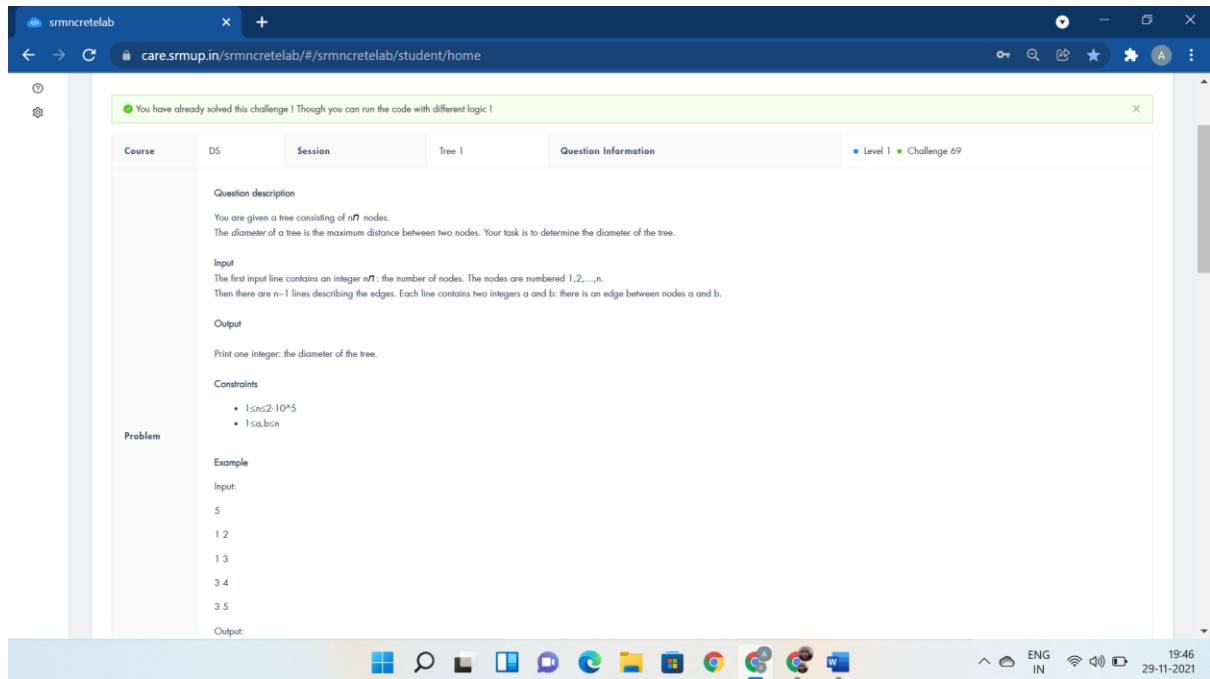
```

    }

    return 0;

    cout<<"while(m1--)"<<endl;
}

```



```

#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)";

}

```

care.srmup.in/srmncretelab/#/srmncretelab/student/home

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

Course	DS	Session	Tree 1	Question Information	Level 1	Challenge 70
<p>Question description</p> <p>You are given a list consisting of n integers. Your task is to remove elements from the list at given positions, and report the removed elements.</p> <p>Constraints</p> <ul style="list-style-type: none"> $1 \leq n \leq 2 \cdot 10^5$ $1 \leq x_i \leq 10^9$ $1 \leq p_i \leq n-1$ <p>Problem</p> <p>Input</p> <p>The first input line has an integer n: the initial size of the list. During the process, the elements are numbered $1, 2, \dots, k$ where k is the current size of the list. The second line has n integers x_1, x_2, \dots, x_n: the contents of the list. The last line has n integers p_1, p_2, \dots, p_n: the positions of the elements to be removed.</p> <p>Output</p> <p>Print the elements in the order they are removed.</p> <p>Logical Test Cases</p> <div> <p>Test Case 1</p> <p>INPUT (STDIN)</p> <pre>5 2 6 1 4 2 3 1 3 1 1</pre> </div> <div> <p>Test Case 2</p> <p>INPUT (STDIN)</p> <pre>6 7 2 6 1 4 2 11 3 1 3 1 1</pre> </div>						

```

#include <stdio.h>

#define N 200000

#define N_ (1 << 18)

int tr[N_ * 2];

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

    tr[k] = r - l;

    if (r - l > 1) {

```

```

    int m = (l + r) / 2;

    build(k * 2 + 1, l, m);

    build(k * 2 + 2, m, r);
}
}

int query(int k, int l, int r, int x) {
    int m, k1, k2;

    tr[k]--;

    if (r - l == 1)
        return r;

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

    return tr[k1] >= x ? query(k1, l, m, x) : query(k2, m, r, x - tr[k1]);
}

int main() {

    int n, h, i, x;

    scanf("%d", &n);

    int aa[n];

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

    build(0, 0, n);

    for (h = 0; h < n; h++) {
        scanf("%d", &x);

        i = query(0, 0, n, x) - 1;

        printf("%d ", aa[i]);
    }

    printf("\n");

    return 0;
}

```