Name : Nishant Milanbhai Bhandari

Roll NO : CE008

STUDENT ID : 21CEUBG051

Lab:1

Problem Code: : **PRACTICEPERF**

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

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);

    int n,k;
    cin >> n >> k;

    int count=0;
    for (int i=0;i<=n-1;i++)
    {
        int a;
        cin >> a;
        if (a%k==0)
        {
            count++;
        }
    }

    cout << count << endl;
    return 0;
}
```

o/p:

```
3
```

Problem Code **INTEST**

Code:

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

```
using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);

    int n,k;
    cin >> n >> k;

    int count=0;
    for (int i=0;i<=n-1;i++)
    {
        int a;
        cin >> a;
        if (a%k==0)
        {
            count++;
        }
    }

    cout << count << endl;
    return 0;
}
```

o/p:

```
4
```

Problem Code : **DECINC**

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

void solve(){
    int n;
    cin >> n;
    if(n%4==0)
    {
        cout << n+1 << endl;
        return;
    }
    cout << n-1 << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

//   int t;
//   cin >> t;
//   while(t--){
//      solve();
//   }
    solve();
   return 0;
}
```

o/p:

```
4
```

Problem Code : **HS08TEST**

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

using namespace std;
```

```
int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.precision(2);

    int a;
    float b;
    cin >> a >> b;

    if ((a%5==0) && (b-a-0.5>=0))
    {
        cout << fixed << b-a-0.5 << endl;
    }
    else
    {
        cout << fixed << b << endl;
    }

    return 0;
}
```

o/p:

```
120.00
```

Problem Code :
PALL01

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

void solve(){
    string str;
    cin >> str;
    for(int i=0; i<str.size()/2; i++)
    {
        if(str[i]!=str[str.size()-i-1])
        {
            cout << "loses" << endl;
            return;
        }
    }
    cout << "wins" << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
   // solve();
   return 0;
}
```

o/p:

```
loses
wins
wins
```

Problem Code :
MATCHES

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

int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);
   int arr[11]={6,2,5,5,4,5,6,3,7,6};
   int t;
   cin >> t;
   while(t--){
    int a, b,count=0;
    cin >> a >> b;
    a+=b;
    while(a!=0)
    {
        count+=arr[a%10];
        a=a/10;
    }
    cout << count << endl;


   }
   // solve();
   return 0;
}
```

o/p:

```
13
10
7
```

Problem Code : **TLG**

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


void solve(){
    int n,maxi=-1, pre=0, flag=0;
    int flag1=0, flag2=0;
    cin >> n;
    while(n--)
    {
        int temp1, temp2;
        cin >> temp1 >> temp2;
        flag1+=temp1;flag2+=temp2;
        maxi=max(maxi,abs(flag1-flag2));
        if(maxi!=pre)
        {
            if(flag1>flag2)
            {
                flag=1;
            }
            else
                flag=2;
        }
        pre=maxi;
    }
    cout << flag <<" " << maxi << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);
//   int t;
//   cin >> t;
//   while(t--){
//      solve();
//   }
    solve();
   return 0;
}
```

o/p:

```
1 58
```

Lab : 2

Problem Code : **SUDBOOKS**

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

class stack
{
    int top;
    int* arr;
    int size;

    public:
    stack(int size)
    {
        top = -1;
        this->size=size;
        arr = new int[size];
    }

    void push(int n)
    {
        if(top<=size-1)
        {
            arr[++top]=n;
        }
        else
        {
            cout << "stack is overflow" << endl;
        }
    }

    void pop()
    {
        if(top==-1)
        {
            cout << "kuchbhi?" << endl;
        }
        else
        {
            cout << arr[top] << endl;
            top--;
        }
    }
};


int main() {
    int n;
    cin >> n;
         stack s(n);
         int type;
         while(n--)
         {
             cin >> type;
             if(type==-1)
             {
                 s.pop();
             }
             else
            {
                int temp;
                cin >> temp;
                s.push(temp);
            }
        }
        return 0;
}
```

o/p:

```
45
2
kuchbhi?
```

Problem Code : **COMPILER**

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

using namespace std;

int main() {

    int t;

    cin>>t;

    while(t--)

    {
```

```
stack<char> s1;
string str;
int count = 0;
cin>>str;
char *next = &str[0];
int ans=0;

for(int i=0; i<str.size(); i++)
{
    if(next[i] == '<' )
   {
          s1.push('<');
   }
```

```
}

else if(next[i] == '>')

{

    if(s1.empty())

        break;

    else

    {

        s1.pop();

        count++;

        if(s1.empty()){

            ans+=count;

            count=0;

        }
```

```
                }

            }

        }

            cout<<ans*2<<endl;

    }

    return 0;

}
```

o/p:

```
4
0
2
```

Problem Code : **CHFQUEUE**

```
#include <bits/stdc++.h>
using namespace std;
#define lli long long int
#define vi vector<int>
#define pi pair<int,vi>
```

```
#define pb push_back
#define f first
#define s second
lli mod = 1e9 + 7;
int n;
int me;

int query(int ft[],int pos){
 int ans = n;
 while(pos > 0){
 ans = min(ans,ft[pos]);
 pos -= (pos & -pos);
 }
 return ans;
}
void update(int ft[],int pos,int
val){  while(pos <= me){
 ft[pos] = val;
 pos += (pos & -pos);
 }

}

void solve(){
 int k;cin >> n >> k;

          vector<int> arr(n);
          me = 0;
          for(int i=0;i<n;i++){
           cin >> arr[i];
           me = max(me,arr[i]);
           }
          //cout << me <<endl;
          int ft[me+1];
          for(int i=1;i<=me;i++) ft[i] =
          n; lli ans = 1;
          int a[n];
          for(int i=n-1;i>=0;i--){
            int val = query(ft,arr[i]-1);
            // cout << val << endl;
            if(val == n){
            a[i] = 1;
            } else {
            int num = val - i + 1;
            a[i] = num;
            }
            update(ft,arr[i],i);
            }

           for(int i=0;i<n;i++){
            ans = (ans * 1LL*a[i]) % mod;
            }
           cout << ans << endl;
           return;
}

int main() {
        // your code goes here
        solve();
        return 0;
}
```

o/p : 2

Problem Code : **CLCO01**

```
#include <bits/stdc++.h>
using namespace std;
void print(stack<string> &s2)
{
   if (!s2.empty())
   {
      string temp = s2.top();
      s2.pop();
      print(s2);
```

```
        cout << temp << "/";
    }
}

int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        int n;
        cin >> n;
        string a, b;
        stack<string> s;
        while (n--)
        {
            cin >> a;
            if (a == "pwd")
            {
                cout << "/";
                stack<string> s2 = s;
                print(s2);
                cout << endl;
            }
            else
            {
                cin >> b;
                string cur = "";
                stack<string> s3;
                if (b[0] == '/')
                {
                    s3.swap(s);
                    b = b.substr(1, b.length() - 1);
                }
                for (auto c : b)
                {
                    if (c != '/')
                    {
                        cur += c;
                    }
                    else
                    {
                        if (cur == "..")
                        {
                            if (!s.empty())
                            {
                                s.pop();
                            }
                        }
                        else
                        {
                            s.push(cur);
                        }
                        cur = "";
                    }
                }
                if (cur == "..")
                {
                    if (!s.empty())
                        s.pop();
                }
                else
                {
                    s.push(cur);
                }
            }
        }
    }
    return 0;
}
```

```
o/p : /
/home/csed/
/lab/dir/
/home/
/home/lab/
```

Problem Code : **LRGREC**

```
#include <bits/stdc++.h>
#include <string>
#include <vector>
#include <algorithm>
#include <iomanip>

using namespace std;
class Stack
{

    int top;
    int arr[100000];

public:
    Stack()
    {
        top = -1;
    }

    int pop()
    {
        return arr[top--];
    }

    void push(int val)
    {

        arr[++top] = val;
    }

    int Top()
    {
        if (top < -1)
        {
            cout<<"No elements in the stack";
        }
        return arr[top];
    }
    bool empty()
    {
        if (top == -1)
        {
            return true;
        }
        return false;
    }
    int size()
    {
        return top + 1;
    }
};

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int t;
    cin >> t;
    while (t--)
    {
        int n;
        cin >> n;
        int h[n];
        for (int i = 0; i < n; i++)
        {
            cin >> h[i];
        }
        Stack s;
        int l[n];
        int r[n];
        for (int i = 0; i < n; i++)
        {
            while (!s.empty() and (h[s.Top()] >= h[i]))
            {
                s.pop();
            }
            if (s.empty())
                l[i] = 0;
            else
                l[i] = s.Top() + 1;
            s.push(i);
        }

        while (!s.empty())
            s.pop();

       for (int i = n - 1; i >= 0; i--)
        {
            while (!s.empty() and (h[s.Top()] >= h[i]))
            {
                s.pop();
            }
            if (s.empty())
                r[i] = n - 1;
            else
                r[i] = s.Top() - 1;
            s.push(i);
        }

        int64_t  maxArea = 0;
        for (int i = 0; i < n; i++)
        {
            maxArea = max(maxArea, (r[i] - l[i] + 1) *
int64_t(h[i]));
        }

        cout << maxArea << endl;
    }
```

```
        return 0;
    }
```

o/p : /

```
10
12
```

Lab : 3

Problem Code : LL1

```
struct Node {
        int data;
        struct Node* next;
        Node(int x){
                data = x;
                next = NULL;
        }
};

int getNthNodeFromEnd(struct Node* head, int n){ Node*
        temp = head;
        int l=0;
        while(temp->next!=NULL)
        {
                l++;
                temp = temp->next;
        }
        int nx=l-n;
        temp=head;
        while(temp->next!=NULL && nx>=0)
        {
```

```
        temp = temp->next;
        nx--;
    }
    return temp->data;
}
```

o/p:

```
1 1
```

Problem Code : REMDUP

```
class Solution{
    public:
    Node* removeDuplicates(Node* head) {
        Node* temp=head;
        if(!temp)
        {
            return NULL;
        }
        while(temp->next!=NULL)
        {
                if(temp->val==temp->next->val)
            {
                   temp->next=temp->next->next;
            }
            else
            {
                temp=temp->next;
            }
        }
        return head;
    }
};
```

o/p : /

```
6 15 22
13 17 20
5 6
9
```

Problem Code : LL2

```
struct Node {
        int data;
        struct Node* next;
        Node(){
                next = NULL;
        }
        Node(int x){
                data = x;
                next = NULL;
        }
};
Node* merge(Node* head1, Node* head2){
        Node *result = NULL;
      if (head1 == NULL) {
            return head2;
      }
      if (head2 == NULL) {
            return head1;
      }
      if (head1->data < head2->data) {
            result = head1;
            result->next = merge(head1->next, head2); } else {
            result = head2;
            result->next = merge(head1, head2->next); }
      return result;
}
```

o/p : /

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

Problem Code :
LL3

```
struct Node {
        int data;
        struct Node* next;
        Node(){
                next = NULL;
        }
        Node(int x){
                data = x;
                next = NULL;
        }
};

 Node* mrg(Node* first, Node* second){ Node
                            *third,*last;

        if(first->data < second->data) {
                third = last = first;
                first = first->next;
                last->next = NULL;
        }
        else
        {
                third = last = second;
                second = second->next;
                last->next = NULL;
        }

        while(first && second)
        {
                if(first->data > second->data) {
                    last->next = second;
                    last = second;
                    second = second->next;
                    last->next = NULL;
                }
                else
                {
                    last->next = first;
                    last = first;
                    first = first->next;
                    last->next = NULL;
                }
        }
        if(first != NULL)
                last->next = first;
        else
                last->next = second;

        return third;
}

Node* solve(vector<Node*> v,int l,int r) {
        if(l == r) return v[l];
        int mid = (l+r)/2;

        Node *left = solve(v,l,mid);
        Node *right = solve(v,mid+1,r); return
        mrg(left,right);
}

Node* merge(vector<Node*> v){
        if(v.size() == 0) return NULL;

        return solve(v,0,v.size()-1);

}
```

o/p : /

1 1 2 2 3 3 100 101 102

Problem Code :

NSELL

```
vector<long long int> NGE(vector<long long int> v) {
        stack<long long int> s;
        vector<long long int> nge(v.size());

        for(int i = 0 ; i < v.size() ; i++)
        {
                while(!s.empty() && v[i] < v[s.top()]) {
                        nge[s.top()] = v[i];
                        s.pop();
                }
                s.push(i);
        }
        while(!s.empty())
        {
                nge[s.top()] = -1;
                s.pop();
        }

        return nge;
}
class Solution{
        public:
        Node* nextSmallerValue(Node* head)
        {
                vector<long long int> v;
                Node *temp = head;
                while(temp)
                {
                        v.push_back(temp->val);
                        temp = temp->next;
                }
                vector<long long int> nge = NGE(v); temp
                = head;

                for(auto i : nge)
                {
                        temp->val = i;
                        temp = temp->next;
                }

                return head;
        }
};
```

```
o/p : /
-1 -1
18 16 12 8 -1 -1
9 5 5 2 1 -1
```

Lab : 4

Problem Code :

FEMA2

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


void popall(queue<int> &q)
```

```
{
    while(q.size()!=0)
    {
        q.pop();
    }
}
void solve(){
  int n, k;
    cin >> n >> k;
    string s;
    cin >> s;
    std::queue<int> iron, ma;
    int sheet = 1;
    int ans1 = 0;
    for(int i=0;i<n;i++)
    {
        if (s[i] == ':')
        {
            sheet++;
        }
        else if (s[i] == 'M')
        {
            while(!iron.empty() && abs(iron.front() - sheet) > k){
                        iron.pop();
                    }
                    if(!iron.empty()){
                        ans1++;
                        iron.pop();
                    }
                    else{
                        ma.push(sheet);
                    }

        }
        else if
            (s[i] == 'X')
        {
            popall(iron);
            popall(ma);
        }
        else if (s[i] == 'I')
        {
                  while(!ma.empty() && abs(ma.front() - sheet) > k){
                        ma.pop();
                    }
                    if(!ma.empty()){
                        ans1++;
                        ma.pop();
                    }
                    else{
                        iron.push(sheet);
                    }
        }
        sheet++;
    }
    cout<<ans1<<endl;
}

int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
   // solve();
   return 0;
}
```

o/p :

```
1
2
```

Problem Code :

COOK82C

```
#include <iostream>
#include <queue>
#include <vector>
#include <climits>
#include <algorithm>

using namespace std;

typedef long long int number;

class Solution {
public:
 void printNLargest(vector<number> &nums, int m) {
sort(nums.rbegin(), nums.rend());

 queue<number> q;
 int index = 0, counter = 0, qi = 0;

 while (m--) {
 cin >> qi;
 number ans = 0;
 for ( ;counter < qi; counter++) {
 if (index < nums.size() && (q.empty() || nums[index] >
q.front())) {
 ans = nums[index];
 index++;
 } else {
 ans = q.front();
 q.pop();
 }
 q.push(ans/2);
 }
 cout << ans << endl;
 }
 }
};

int main() {
        ios_base::sync_with_stdio(false);
        cin.tie(nullptr);

        int n, m; cin >> n >> m;
        vector<number> nums(n);
 for (int i = 0; i < n; i++)
 cin >> nums[i];

        Solution solution;
        solution.printNLargest(nums ,
        m); return 0;
}
```

o/p :

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

Problem Code :

CHFQUEUE

```
#include <bits/stdc++.h>
using namespace std;
#define lli long long int
#define vi vector<int>
#define pi pair<int,vi>
#define pb push_back
#define f first
#define s second
lli mod = 1e9 + 7;
```

```
int n;
int me;

int query(int ft[],int pos){
 int ans = n;
 while(pos > 0){
 ans = min(ans,ft[pos]);
 pos -= (pos & -pos);
 }
 return ans;
}
void update(int ft[],int pos,int
val){  while(pos <= me){
 ft[pos] = val;
 pos += (pos & -pos);
 }

}

void solve(){
 int k;cin >> n >> k;

          vector<int> arr(n);
          me = 0;
          for(int i=0;i<n;i++){
           cin >> arr[i];
           me = max(me,arr[i]);
           }
          //cout << me <<endl;
          int ft[me+1];
          for(int i=1;i<=me;i++) ft[i] =
          n; lli ans = 1;
          int a[n];
          for(int i=n-1;i>=0;i--){
           int val = query(ft,arr[i]-1);
           // cout << val << endl;
           if(val == n){
           a[i] = 1;
          } else {
          int num = val - i + 1;
          a[i] = num;
          }
          update(ft,arr[i],i);
          }

         for(int i=0;i<n;i++){
          ans = (ans * 1LL*a[i]) % mod;
          }
         cout << ans << endl;
         return;
}

int main() {
         // your code goes here
         solve();
         return 0;
}
```

// how to find position of the smaller element // so in ft we will make pair (sum,arraylist of indices) o/p : 2

Problem Code :

CAC202

```
#include <stdio.h>
int main(){
 int t;
 scanf("%d",&t);
 while(t--){
 int n,x,m=0;
 scanf("%d",&n);
 int s[n],boo[n],top=-1;
 for(int i=0;i<n;i++){
 boo[i]=0;
 }
 while(n--){
 scanf("%d",&x);
 if(x>m){
 printf("%d ",x);
 for(int i=m+1;i<x;i++){  top++;
 s[top]=i;
 }
 m=x;
 }
 else{
 boo[x]=1;
 while(top!=-1&&boo[s[top]]==1){
boo[s[top]]=0;
 printf("%d ",s[top]);
 top--;
 }
 }
 }
 while(top!=-1){
 printf("%d",s[top]);
 top--;
 }
 printf("\n");
 }
 return 0;
}o/p :
```

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

4 3 2 5 1
```

Problem Code :

WASHHAND

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

int main() {
        int t;
        cin>>t;
        while(t--){
         int n;
         cin>>n;

         string s;
         cin>>s;

         int m;
         cin>>m;
         int a[m];
         for(int i=0;i<m;i++){
         cin>>a[i];
         }
         int visited[n],count=0;
         memset(visited,0,sizeof(visited));
        queue<int>q;
         for(int i=0;i<n;i++){
         if(s[i]=='1'){
         q.push(i);
         count++;
         }
         }
         for(int i=0;i<m;i++){
         visited[a[i]-1] = -1;
         visited[a[i]-2] = 1;
        int len = q.size();
        for(int i=0;i<len;i++){
        int k = q.front();
        q.pop();
        if(k-1>=0){
        if(visited[k-1]!=1 && s[k-1]!='1'){
       q.push(k-1);
        count++;
        s[k-1]='1';
        }
        }
        if(k+1<n){
        if(visited[k+1]!=-1 && s[k+1]!='1'){
       q.push(k+1);
        count++;
        s[k+1]='1';
        }
        }
        }
        }
        cout<<count<<endl;

        }
       return 0;
}
```

o/p :

```
6
1
```

Lab- 5

Problem Code: OJUMPS

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

void solve(){
    long long  n;
    cin >> n;
    if(n%6==0 || n%6==1 || n%6==3)
    {
        cout << "yes" << endl;
        return;
    }
    cout << "no" << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

//   int t;
//   cin >> t;
//   while(t--){
//      solve();
//   }
    solve();
   return 0;
}
```

Output :
Yes

Problem Code: TRICOIN

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

void solve()
{
        int n;
        cin>>n;
         int count=0;
         int j=0;
         for(int i=1; 1>0 ; i++){
             j = j+i;
             if(n>=j)
                 count++;
```

```
            else
                break;
        }
        cout<<count<<endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
   // solve();
   return 0;
}
```

Output :

```
2
2
3
```

Problem Code: COPS

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

void solve(){
        int n,x,y;
        cin>>n>>x>>y;
        vector<int> houses(101,0);
        vector<int> cM(n);
        int minut = x*y;
        for (int i = 0; i < n; i++)
        {
            cin>>cM[i];
            int lower = cM[i]-minut;
            lower = lower<=0?1:lower;
            int upper = cM[i]+minut;
            upper = upper>100?100:upper;
            for(int j=lower;j<=upper;j++){
                houses[j] = 1;
            }
        }

        int sum = -1;
        for (auto &i : houses)
        {
            if(i==0){
                sum+=1;
            }
        }
        cout<<sum<<endl;
}
int main(){
  ios_base::sync_with_stdio(0);
  cin.tie(0), cout.tie(0);

  int t;
  cin >> t;
  while(t--){
    solve();
  }
  // solve();
  return 0;
}
```

Output :

```
0
18
9
```

Problem Code: AVGFLEX

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

void solve()
```

```
{
        int n;
        cin>>n;
        int count=0;
        int j=0;
        for(int i=1; 1>0 ; i++){
            j = j+i;
            if(n>=j)
                count++;
            else
                break;
        }
        cout<<count<<endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
   // solve();
   return 0;
}
```

Output :

```
3
2
3
```

Problem Code: MINEAT

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

int ispossible(vector<ll> arr, ll x)
{
        ll counts=0;
        for(int i = 0 ; i < arr.size();i++){
        if(arr[i] % x == 0) counts += arr[i]/x;
        else if(arr[i] > x) counts += (arr[i]/x) + 1;
        else counts += 1;
    }
    return counts;
}
void solve()
{
    int n, h;
    cin >> n >> h;
    vector<ll> num(n);
    for(auto &i: num)
      cin >> i;
    ll s=0, e=0, ans = -1, total=0;
    for(auto i: num)
      total+=i;
    e = 1e9;
    sortall(num);
        ll mid = s+(e-s)/2;
    while(s<=e)
    {
        if(ispossible(num, mid)<=h)
        {
                ans=mid;
                e=mid-1;
        }
        else
        {
            s=mid+1;
        }
                mid = s+(e-s)/2;
    }
    cout <<  ans << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
   // solve();
   return 0;
}
```

Output :

```
3
2
4
```

Lab - 6

Problem Code: RISK

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

int dfs(vector<string> &num,int i,int j)
{
    if(i<0 || j<0 || i>=num.size() || j>=num[0].size())
    {
        return 0;
    }

    if(num[i][j]!='1')
    {
        return 0;
    }

    num[i][j]='2';

    int right = dfs(num, i, j+1);
    int left = dfs(num, i, j-1);
    int up = dfs(num, i-1, j);
    int down = dfs(num, i+1, j);

    return right+left+up+down+1;
}

void solve(){
    int n, m;
    cin >> n >> m;
    vector<string> num;
    reb(i,0,n)
    {
        string str;
        cin >> str;
        num.pb(str);
    }

    vector<int> ans;
    reb(i,0,n)
    {
        reb(j,0,m)
        {
            if(num[i][j]=='1')
            {
                ans.push_back(dfs(num, i, j));
                // cout << dfs(num, i, j) << " ";
            }
        }
    }

    sort(ans.begin(),ans.end(), greater<int>());
    // for(auto i: ans)
    // cout << i << " ";
    int total=0;
    for(int i=1; i<ans.size(); i++)
    {
        if(i%2==1)
        total+=ans[i];
    }
    cout << total << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
   // solve();
   return 0;
}
```

Output :

```
2
0
1
```

Problem Code: ADAKNG

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

bool valid(int i, int j)
{
    return i>=0 && j>=0 && i<8 && j<8;
}
int bfs(int i, int j, int k,vector<vector<int>> visited,
vector<vector<int>> level)
{
    int count=0;
    queue<pair<int, int>> q;
    q.push({i,j});
    visited[i][j]=1;
    level[i][j]=0;
    vector<pair<int, int>> movements{{1,1},{1,-1},{-1,1},{-1,-1},
    {0,1},{0,-1},{1,0},{-1,0}};

    while(!q.empty())
    {
        auto v = q.front();
        int v_x = v.first, v_y = v.second;
        // cout << v_x << " " << v_y << endl;
        q.pop();

        for(auto movement: movements)
        {
            int child_x = movement.first + v_x;
            int child_y = movement.second + v_y;
            if(!valid(child_x, child_y)) continue;
            if(visited[child_x][child_y])  continue;

            q.push({child_x,child_y});
            visited[child_x][child_y]=1;
            level[child_x][child_y] = level[v_x][v_y]+1;
        }
        // cout << count << " " << k << endl;
    }
    reb(i,0,8)
    {
        reb(j,0,8)
        {
            if(level[i][j]<=k)
                count++;
        }
    }
    return count;
}
void solve(){
    int i, j, k;
    cin >> i >> j >> k;

    vector<vector<int>> visited, level;
    reb(l,0,8)
    {
        vector<int> temp(8,0), temp1(8, INT_MAX);
        visited.pb(temp);
        level.pb(temp1);
    }
    cout << bfs(i-1, j-1, k, visited, level) << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
//    solve();
   return 0;
}
```

Output :

```
6
```

Problem Code: SNSOCIAL

```
#include <bits/stdc++.h>
using namespace std;
const int N = 1e3 + 10;
const int INF = 1e9 + 10;
int val[N][N];
int vis[N][N];
int lev[N][N];
int n, m;

vector<pair<int, int>> movements ={{0,1}, {0,-1}, {1,0}
, {-1,0},{1,1},{1,-1},{-1,1},{-1,-1}};
```

```
bool isvalid(int i, int j){
    return i>=0 && j>=0 && i<n && j<m;
}
int bfs(){
    int mx=0;
     for (int i = 0; i < n; ++i)
    {
        for(int j=0; j<m; ++j)
        {
            mx=max(mx, val[i][j]);
        }
    }
    queue<pair<int, int>> q;
    for(int i=0; i<n; i++)
    {
        for(int j=0; j<m; j++)
        {
            if(mx==val[i][j])
            {
                q.push({i,j});
                lev[i][j]=0;
                vis[i][j]=1;
            }
        }
    }
    int ans=0;
    while(!q.empty())
    {
        auto  v = q.front();
        int v_x = v.first;
        int v_y = v.second;
        q.pop();
        for(auto movement: movements)
        {
            int child_x=movement.first+v_x;
            int child_y=movement.second+v_y;
            if(!isvalid(child_x, child_y)) continue;
            if(vis[child_x][child_y])  continue;
            q.push({child_x, child_y});
            lev[child_x][child_y]=lev[v_x][v_y] + 1;
            vis[child_x][child_y] = 1;
            ans = max(ans,lev[child_x][child_y] );
        }
    }
    return ans;
}
void reset()
{
    for (int i = 0; i < n; ++i)
    {
        for (int j = 0; j < m; ++j)
        {
            vis[i][j] = 0;
            lev[i][j] = INF;
        }
    }
}
int main()
{
    int t;
    cin >> t;
    while (t--)
    {
        cin >> n >> m;
        reset();
        for (int i=0; i<n; ++i)
        {
            for (int j = 0; j < m; ++j)
            {
                cin >> val[i][j];
            }
        }
        cout << bfs() << endl;
    }
    return 0;
}
```

Output :

```
0
1
2
```

Problem Code: TINOI17A

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

inline bool isvalid(int i, int j,int n,int m)
{
    return i>=0 && j>=0 && i<n && j<m;
}

void solve(){
    int n,m;
    cin>>n>>m;
    int k;
    cin>>k;
    vector<vector<int>> v(n,vector<int>(m,0));
    vector<vector<bool>> visited(n,vector<bool>(m,false));
    while(k--){
        int x,y;
        cin>>x>>y;
        x--;
        y--;
        v[x][y]=1;
    }
    int mxCost=0;
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(visited[i][j]==false && v[i][j]==1){
                int currCost=0;
                queue<pair<int,int>> q;
                q.push({i,j});
                while(q.size()>0){
                    pair<int,int> top=q.front();
                    q.pop();

                    int x=top.first;
                    int y=top.second;
                    visited[x][y]=true;
                    int cost=4;
                    if(isvalid(x-1,y,n,m) && v[x-1][y]==1){
                        if( true && visited[x-1][y] == false )
q.push({x-1,y});
                        visited[x-1][y]=true;
                        cost--;
                    }
                    if(isvalid(x+1,y,n,m) && v[x+1][y]==1){
                        if( true && visited[x+1][y] == false )
q.push({x+1,y});
                        visited[x+1][y]=true;
                        cost--;
                    }
                    if(isvalid(x,y-1,n,m) && v[x][y-1]==1){

                        if( true && visited[x][y-1] == false )
q.push({x,y-1});
                        visited[x][y-1]=true;
                        cost--;
                    }
                    if(isvalid(x,y+1,n,m) && v[x][y+1]==1){
                        if( true && visited[x][y+1] == false )
q.push({x,y+1});
                        visited[x][y+1]=true;
                        cost--;
                    }
                    currCost+=cost;
                }
                // cout<<"curr cost : "<<currCost<<endl;
                mxCost=max(mxCost,currCost);
```

```
            }
        }
    }
    cout<<mxCost<<endl;
}
signed main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t=1;
//   cin >> t;
   while(t--){
     solve();
   }
//    solve();
   return 0;
}
```

Output :

```
16
```

Problem Code: INOI1601

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

void readv(vector<int>& v){
    for(auto & num:v){
        cin>>num;
    }
}
void printv(vector<int>& v){
    for(auto &num:v) cout<<num<<" ";
    cout<<endl;
}
int dfs(vector<vector<int>> & graph,vector<int> &
v,vector<int>& diff,int start){
    int mn=v[start];
    for(auto & node: graph[start]){
        mn=min(mn,dfs(graph,v,diff,node));
    }
    diff[start]=v[start]-mn;
    // cout<<start<<" : "<<diff[start]<<endl;
    return mn;
}
void solve(){

  int n;
  cin>>n;
  vector<int> v(n);
  readv(v);
//   printv(v);
  vector<int> par(n);
  readv(par);
  vector<vector<int>> graph(n);
  vector<int> diff(n,LLONG_MIN);
  int start=0;
  for(int i=0;i<n;i++){

    int index=par[i]-1;
    if(par[i]==-1){
        start=i;
    }else graph[index].push_back(i);
  }
  dfs(graph,v,diff,start);

  int ans=LLONG_MIN;
  for(auto num:diff) ans=max(ans,num);
  cout<<ans<<endl;

}
signed main()
{
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t=1;
//   cin >> t;
```

```
    while(t--){
      solve();
    }
    // solve();
    return 0;
}
```

Output :

```
6
```

Lab - 7

Problem Code : CFMAR04

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

using namespace std;

int main()
{
    int t;
    cin>>t;
    int ans =0;

    map<string, int> mp;

    while(t--)
    {
        string str;
        cin>>str;

        sort(str.begin(), str.end());
        mp[str]++;
    }

    for(auto i: mp)
    {
        ans = max(ans,i.second);
    }
    cout<<ans<<endl;
    return 0;
}
```

o/p :

```
2
```

Problem Code :

COSTEMP

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

```
void dfs(int vertex, int parent,vector<int>
&depth,vector<vector<int>> adj,int &ans)
{
    for(auto child: adj[vertex])
    {
        if(child==parent) continue;
        depth[child]=depth[vertex]+1;
        ans+=depth[child];
        dfs(child, vertex, depth, adj, ans);
    }
}

void solve(){
    int n;
    cin >> n;
    vector<vector<int>> adj(n);
    for(int i=0; i<n-1; i++)
    {
      int temp1, temp2;
      cin >> temp1 >> temp2;
      adj[temp1].push_back(temp2);
      adj[temp2].push_back(temp1);
    }
    // int k=0;
    // for(auto i: adj)
    // {
    //     cout << k++ << "- ";
    //     for(auto j: i)
    //     cout << j << " ";
    //     cout << endl;
    // }
    for(int i=0; i<n; i++)
    {
        int ans=0;
        vector<int> depth(n,0);
        dfs(i,-1,depth,adj, ans);
        cout << ans << " ";
    }
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

  //  int t;
  //  cin >> t;
  //  while(t--){
  //    solve();
  //  }
   solve();
   return 0;
}
```

o/p :

```
6 8 9 5 8
```

Problem Code : FAMTREE

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

#define int long long
#define N 100001

int n;
int w[N],p[N];
vector<int> adj[N];
int max_n[N],min_n[N];
int diff = INT_MIN;

void dfs(int node,int par){
    for(auto it : adj[node]){
        dfs(it,node);
        min_n[node] = min(min_n[node],min_n[it]);
        max_n[node] = max(max_n[node],max_n[it]);

    }
    max_n[node] = max(max_n[node],w[node]);
    min_n[node] = min(min_n[node],w[node]);


    diff = max(diff,(abs(w[node]-max_n[node])));
    diff = max(diff,(abs(w[node]-min_n[node])));
}

signed main() {
        cin>>n;

        for(int i = 1;i<=n;i++){
            cin>>w[i];
        }
        int root;
        for(int u = 1;u<=n;u++){

            cin>>p[u];
            if(p[u] == -1){
                root = u;
            }
            else{
                adj[p[u]].push_back(u);
            }
        }
        for(int i = 1;i<N;i++){
            max_n[i] = INT_MIN;
            min_n[i] = INT_MAX;
        }

        dfs(root,0);



        cout<<diff<<endl;
        return 0;
```

```
}
```

o/p :

6

Problem Code : SUBREM

```
#include <iostream>
#include <bits/stdc++.h>
#include <numeric>
#define INF (int)1e9
#define EPS 1e-9
#define fast
ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
#define PI 3.1415926535897932384626433832795
#define MOD 1000000007

using ll = long long;
using namespace std;
ll solve(vector<vector<ll>> &adj,ll parent, vector<ll> &vals, int
root, ll &x){
  ll res = -x;
  ll summ = vals[root];
        for(auto it : adj[root]){
                    if(it != parent)
            summ += solve(adj,root,vals,it,x);
        }
        return max(res,summ);
}
int main()
{
    fast;
        ll tc = 1;
        cin>>tc;
        while(tc--){
                ll n,x;
                cin>>n>>x;
                vector<ll> Sn(n+1,0);
                vector<vector<ll>> adj(n+1);
                vector<ll> vals(n+1);
                for(int i = 1; i<=n;i++){
                    cin>>vals[i];
                }
                for(int i = 0; i<n-1;i++){
                    ll p,q;
                    cin>>p>>q;
                    adj[p].push_back(q);
                    adj[q].push_back(p);
                }
            cout<<solve(adj,0,vals,1,x)<<'\n';
        }
        return 0;
}
```

o/p :

```
-4
```

Problem Code :

DIAMTREE

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

// #define int long long;
void dfs(int v, int parent,vector<vector<int>> &graph, vector<int>
&depth, int& maxi_v)
{
    for(auto child: graph[v])
    {
        if(child==parent) continue;
        depth[child]=depth[v]+1;
        if(depth[child]>depth[maxi_v] || maxi_v==child)
        {
            maxi_v=child;
        }
        dfs(child, v, graph, depth, maxi_v);
    }
}
void solve()
{
    int v;
    cin >> v;
    vector<vector<int>> graph(v);
    vector<int> depth1(v,0),depth2(v,0);
    for(int i=0; i<v-1; i++)
    {
        int temp1,temp2;
        cin >> temp1 >> temp2;
        graph[temp1-1].push_back(temp2-1);
        graph[temp2-1].push_back(temp1-1);
    }
    int maxi_v=1, maxi_ans=1;
    dfs(0, -1, graph, depth1, maxi_v);
    dfs(maxi_v, -1, graph, depth2, maxi_ans);
    cout << depth2[maxi_ans] << endl;
}

int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);
    int t;
    cin >> t;
   while(t--){
     solve();
   }
//   solve();
   return 0;
}
```

o/p :

```
4
```

```
3
```

Lab - 8

Problem Code: ROOTTREE

```
#include<bits/stdc++.h>
using namespace std;
void solve(){
    int n;
    cin >> n;
    vector<vector<int> > adj(n);
    vector<int> indegree(n, 0);
    for(int i=0; i<n-1; i++)
    {
        int u, v;
        cin >> u >> v;
        adj[u-1].push_back(v-1);
        indegree[v-1]++;
    }
    int count=0;
    for(int i=0; i<n; i++)
    {
        if(indegree[i]==0)
            count++;
    }
    cout << count-1 << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
   // solve();
   return 0;
}
```

o/p-

1
0

Problem Code:: DSUBAGS

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

struct Dsu
{
    vector<int> vect;
    int n, cc;
    Dsu(int n) : n(n)
    {
        vect.resize(n);
        iota(vect.begin(), vect.end(), 0);
        cc = n - 1;
    }

    int findvect(int v)
    {
        if (vect[v] == v)
            return v;

        else
            return (vect[v] = findvect(vect[v]));
    }

    bool merge(int u, int v)
    {
        u = findvect(u);
        v = findvect(v);
        if (u == v)
            return 0;
        vect[v] = u;
        cc--;
        return 1;
    }
};

int main()
{
    int n, q;
    cin >> n >> q;
    Dsu s(n + 1);
    while (q--)
    {
        int t;
        cin >> t;
        if (t == 1)
        {
            int a, b;
            cin >> a >> b;
            s.merge(a, b);
        }
        if (t == 2)
        {
            int a, b;
            cin >> a >> b;
            if (s.findvect(a) == s.findvect(b))
            {
                cout << "YES" << endl;
            }
            else
            {
                cout << "NO" << endl;
            }
        }
        if (t == 3)
        {
            cout << s.cc << endl;
        }
    }
}
```

Output :

```
NO
3
2
YES
1
YES
```

Problem Code: SEGDIR

```
#include <bits/stdc++.h>
#define int long long
#define testo int t;cin>>t;while(t--)
#define whl(i,n) for(int i=0;i<n;i++)
using namespace std;

signed main() {
    testo{
        int n;cin>>n;
        int l[n+5],r[n+5],v[n+5];
        int rough=0;
        whl(i,n){
            cin>>l[i]>>r[i]>>v[i];
        }
        whl(i,n){
            vector<pair<int,int>>vp;
            vp.push_back({l[i],1});
            vp.push_back({r[i]+1,-1});
            whl(j,n){
                if( j != i && v[i] == v[j]) {
                    if((l[i] >= l[j] && l[i] <= r[j]) || (r[i] >=l[j] && r[i] <= r[j])){
                            vp.push_back({l[j],1});
                            vp.push_back({r[j]+1,-1});
                        }
                }
            }
            sort(vp.begin(),vp.end());
            int zz=0;
            for(auto x:vp){
                zz+=x.second;
                if(zz>=3){
                   cout<<"NO\n";
                   rough=1;
                   break;
                }
            }
            if(rough==1)break;
        }
        if(rough==1)continue;
        cout<<"YES\n";
    }
        return 0;
}
```

Output:

```
YES
NO
```

Problem Code:: TREEVERS

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
vector<int> w;
vector<vector<int>> adj;
struct trio{
    int inv, ones, zeros;
};
bool cmp(trio &a, trio &b){
    return a.ones*b.zeros < a.zeros*b.ones;
}

trio dfs(int nd, int par){
    trio cur;
    cur.inv = 0, cur.zeros = 0, cur.ones = 0;
    vector<trio> subtr;
    for(int ch: adj[nd]){
        if(ch == par) continue;
        subtr.push_back(dfs(ch, nd));
    }
    sort(subtr.begin(), subtr.end(), cmp);
    if(w[nd] == 1) cur.ones++;
    else cur.zeros++;
    for(trio &t : subtr){
        cur.inv += t.inv;
        cur.inv += cur.ones*t.zeros;
        cur.ones += t.ones;
        cur.zeros += t.zeros;
    }

    return cur;
}
```

```
int32_t main()

{
    int t; cin>>t;
    while(t--){
        int n; cin>>n;
        adj.clear();
        w.resize(n+1);
        adj.resize(n+1);
        for(int i = 1; i <= n; i++){
            cin>>w[i];
        }
        for(int i = 1; i <= n-1; i++){
            int u, v; cin>>u>>v;
            adj[u].push_back(v);
            adj[v].push_back(u);
        }
        trio treevers = dfs(1, 0);
        cout<<treevers.inv<<'\n';
    }
    return 0;
}
```

Output :

```
1
```

Lab-9

Problem Code:: DIGJUMP

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

int main() {
        int n ;
        string s;
        cin>>s;
```

```
       n = s.size();

       vector<int> g[10];
       vector<int> vis(n+1,0) , ans(n+1,0) , vd(10,0) , a(n+1);

       for(int i = 0 ;i < n ;i++)
       {
           a[i] = s[i] - '0';
           g[a[i]].push_back(i);
       }
queue<int> q;
q.push(0);
vis[0] = 1;
while(!q.empty())
{
    int cur = q.front();
    q.pop();
    if(cur == n-1)
    {
        cout<<ans[n-1];
        break;
    }

    if(cur - 1 >=0 && !vis[cur-1])
    {
        vis[cur-1] = 1;
        ans[cur-1] = ans[cur] + 1;
        q.push(cur-1);
    }
    if(cur + 1 < n && !vis[cur+1])
    {
        vis[cur+1] = 1;
        ans[cur + 1] = ans[cur] + 1;
        q.push(cur+1);
    }
    if(!vd[a[cur]])
    {
        vd[a[cur]] = 1;
        for(auto ele : g[a[cur]])
        {
            if(!vis[ele])
            {
                    q.push(ele);
                    vis[ele] =1;
                    ans[ele] = ans[cur] + 1;
                }
            }
        }
    }
          return 0;
}
```

Output :

```
1
4
```

Problem Code: REVERSE

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

#define reb(i,m,n) for(int i=m; i<n; i++)
#define ll long long
#define INF 2e18

  int bfs(ll n,vector<pair<ll,ll>>g[],vector<ll>&lev){
   deque<ll>q;

   q.push_back(1);
   lev[1]=0;

   while(!q.empty()){
     ll curv=q.front();
     q.pop_front();

     for(auto child:g[curv]){

       ll cv=child.first;
       ll wt=child.second;

       if(lev[curv]+wt<lev[cv]){
         lev[cv]=lev[curv]+wt;

       if(wt==1)
         q.push_back(cv);
       else
```

```
          q.push_front(cv);
      }
    }
  }
  if(lev[n]==INF)
    return -1;
  return lev[n];
}

void solve(){
    ll n, m; cin >> n >> m;
    vector<pair<ll,ll>> adj[n+1];
    vector<ll>lev(n+1,INF);

    reb(i,0,m)
    {
        ll a, b; cin >> a >> b;
        if(a!=b)
        {
            adj[a].push_back({b, 0});
            adj[b].push_back({a, 1});
        }
    }
    cout<<bfs(n,adj,lev);
}
int32_t main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

//    int t;
//    cin >> t;
//    while(t--){
//      solve();
//    }
   solve();
   return 0;
}
```

Output :

```
2
```

Problem Code:: GALACTIK

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

#define int long long
const int INF=1e9+10;

void makeset(vector<int> &parent, int n)
{
    for(int i=1; i<=n; i++)
    {
        parent[i]=i;
    }
}

int findparent(vector<int> &parent, int node)
{
    if(parent[node]== node)
        return node;
    return  parent[node] = findparent(parent,  parent[node]);
}

void unionset(int u, int v, vector<int> &parent)
{
    u = findparent (parent, u);
    v = findparent (parent, v);
    parent[v]=u;
}

void solve()
{
    int n, m;
    cin >> n >> m;
    vector<vector<int>> edges(n+1);
    vector<int> parent(n+1);
    makeset(parent, n);
    for(int i=0; i<m; i++)
    {
        int temp1, temp2;
        cin >> temp1 >> temp2;
        unionset(temp1, temp2 , parent);
    }
```

```
vector<int> cost(n+1);
for (int i = 1; i <=n; i++)
{
    cin>>cost[i];
    if(cost[i]<0)cost[i]=INF;
}

//if graph is already connected
set<int>st;
for (int i=1; i<=n; i++)
    st.insert(findparent(parent, i));
 if(st.size()==1){
    cout<<0<<endl;
    return;
 }

//taking minimum value from each component
    map<int,int>min_vals;
for(auto &c:st)min_vals[c]=INF;

for(int i=1;i<=n;i++){
    if(cost[i]>=0){
        int p=findparent(parent, i);
        min_vals[p] = min(min_vals[p],cost[i]);
    }
  }

int mini = INF;
int sum=0;

 for(auto &c:min_vals){
    if(c.second==INF){
        cout << -1 << endl;
        return ;
    }
    sum+=c.second;
    mini = min(c.second,mini);
  }

int k = st.size();
cout<<sum-mini + mini*(k-1)<<endl;
```

```
}

int32_t main(){

   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);


   solve();
   return 0;
}
```

Output :

```
3
-1
```

Problem Code:: CORONA

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

#define reb(i,m,n) for(int i=m; i<=n; i++)

int dijkstra(int n, int source, vector<pair<int,int>> graph[],
unordered_map<int,int> hos)
{
    vector<int> distance(n+1,INT_MAX);
    vector<bool>visited(n+1,false);
    set<pair<int,int>> st;
    st.insert({hos[source],source});
    distance[source]=hos[source];
    while(st.size()>0)
    {
            auto node = *st.begin();
            int v=node.second, dist = node.first;
            st.erase(st.begin());

            if(visited[v]) continue;
            visited[v]=true;
            for(auto child : graph[v])
            {
                int wt=child.second, child_v = child.first;
                // cout << wt << " " << child_v<< endl;
                if(distance[v]+wt+hos[child_v] < distance[child_v] &&
hos[child_v]!=-1)
                {
                    distance[child_v]=distance[v]+wt+hos[child_v];
                    st.insert({distance[child_v],child_v});
                }
            }
    }
    int mini=INT_MAX;
    for(auto i: distance)
    {
        if(i!=INT_MAX)
        cout << i << " ";
        else
        cout << -1 << " ";
    }

    cout << endl;
    for(auto i: hos)
    {
        mini=min(mini, distance[i.first]);
    }
    return mini;
}

void solve(){
    int n, m, k;
    cin >> n >> m >> k;

    vector<pair<int,int>> adj[n+1];
    unordered_map<int,int> hos;
    vector<int> ans;
    reb(i,1,k)
    {
        hos[i]=-1;
    }
    reb(i,1,k)
    {
        int a, b;cin >> a >> b;
        hos[a]=b;
    }
    reb(i,1,m)
    {
        int a, b, wt; cin >> a >> b >> wt;
        adj[a].push_back({b, wt});
    }
    reb(i,1,n)
    {
        ans.push_back(dijkstra(n, i, adj, hos));
    }
    // for(auto i: ans)
    // cout << i << " ";
    // cout << endl;
}
int main(){
   ios_base::sync_with_stdio(0);
   cin.tie(0), cout.tie(0);

   int t;
   cin >> t;
   while(t--){
     solve();
   }
   // solve();
   return 0;
}
```

Output :

```
5 10 19
5 6 15
```

Problem Code:: AMR14B

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

bool checknext(long long &x, long long &y, long long next[])
{
    while(next[x]!=x)
        x=next[x];
    while (next[y]!=y)
        y=next[y];
    return x==y;
}
```

```
void merge(long long x, long long y, long long next[], long long size[])
{
    if (size[x]<size[y])
        swap(x, y);
    next[y]=x;
    size[x]=size[x]+size[y];
    return;
}

int main()
{
    long long t1;
    cin >> t1;
    for (long long t2=0; t2<t1; t2++)
    {
        long long n, m, x, y, d, dt, edge, s1=0, s2=0, i;
        vector<pair<long long, pair<long long, long long> > > edges;
        priority_queue<pair<pair<long long, long long>, long long> > pq;
        cin >> n >> m;
        vector<pair<long long, long long> > adj[n];
        long long distance[n];
        bool visited[n]={0};
        long long next[n], size[n];
        for (i=0; i<m; i++)
        {
            cin >> x >> y >> d;
            adj[x].push_back({y, d});
            adj[y].push_back({x, d});
            edges.push_back({d, {x, y}});
        }

        sort(edges.begin(), edges.end());
        for (i=0; i<n; i++)
        {
            next[i]=i;
            size[i]=1;
        }
        for (i=0; i<m; i++)
        {
            x=edges[i].second.first;
            y=edges[i].second.second;
            d=edges[i].first;
            if (!checknext(x, y, next))
            {
                merge(x, y, next, size);
                s1=s1+d;
            }
        }

        for (i=1; i<n; i++)
            distance[i]=LLONG_MAX;
        distance[0]=0;
        pq.push({{0, 0}, 0});
        while (!pq.empty())
        {
            x=pq.top().second;
            d=-pq.top().first.first;
            edge=-pq.top().first.second;
            pq.pop();
            if (visited[x])
                continue;
            visited[x]=1;
            s2=s2+edge;
            for (i=0; i<adj[x].size(); i++)
            {
                y=adj[x][i].first;
                dt=adj[x][i].second;
                if (!visited[y] && d+dt<=distance[y])
                {
                    distance[y]=d+dt;
                    pq.push({{-distance[y], -dt}, y});
                }
            }
        }

        for (i=0; i<n; i++)
            if (!visited[i])
                break;
        if (i!=n)
        {
            cout << "NO" << endl;
            continue;
        }
```

```
        if (s1==s2)
            cout << "YES" << endl;
        else
            cout << "NO" << endl;
    }

    return 0;
}
```

Output :

```
YES
NO
NO
```

Lab : 10

Problem Code :
 TSORT

```
#include <bits/stdc++.h>
using namespace std;
int main() {
// your code goes here
int t;
cin>>t;
vector <int> a(t);
for(int i = 0; i< t ; i++){
 cin>>a[i];
}
sort(a.begin(),a.end());
for(int x : a)
 cout<<x<<endl;
return 0;
}
```

Output :

```
1
3
5
6
7
```

Problem Code :
 JOHNY

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

```
int binarySearch(int a[], int beg, int end, int val)
{
    int mid;
    if(end >= beg)
    {       mid = (beg + end)/2;
/* if the item to be searched is present at middle */
        if(a[mid] == val)
        {
            return mid+1;
        }
            /* if the item to be searched is smaller than middle,
then it can only be in left subarray */
        else if(a[mid] < val)
        {
            return binarySearch(a, mid+1, end, val);
        }
            /* if the item to be searched is greater than middle,
then it can only be in right subarray */
        else
        {
            return binarySearch(a, beg, mid-1, val);
        }
    }
    return -1;
}
int main() {
    int T;
    cin>>T;

    while(T--){
        int N;
        cin>>N;
        int arr[N];
        for(int i = 0; i<N ;i++){
            cin>>arr[i];
        }
        int K;
        cin>>K;
        int X =arr[K-1] ;
        sort(arr,arr+N);
        cout<<binarySearch(arr,0,N-1,X)<<endl;
    }
         return 0;
}
```

Output :

Problem Code :
EID

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

int main() {
    int T;
    cin>>T;
    while(T--){
        int n;
        cin>>n;
        int arr[n];
        for(int i = 0 ; i<n;i++){
            cin>>arr[i];
        }
        sort(arr,arr+n);
        int diff = arr[1]-arr[0];
        for(int i = 0 ; i<n-1 ;i++){

            if(diff > arr[i+1]-arr[i]){
                diff  = (arr[i+1]-arr[i]);
            }
        }
        cout<<diff<<endl;
    }
         return 0;
}
```

Output :

```
1
0
```

Problem Code :
 CK87MEDI

```
#include <iostream>
#include<bits/stdc++.h>
using namespace std;
int main() {
int t;
cin>>t;
while(t--)
{
 int n,k;
 cin>>n>>k;
 int a[n];
 for(int i=0;i<n;i++)
 {
 cin>>a[i];
 }
 sort(a,a+n);

 cout<<a[(n+k)/2]<<endl;
}
return 0;
}
```

Output :

```
7
9
1
```

Problem Code : BIT2A

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

int main() {
        int T;
        cin>>T;
        while(T--){
            int N;
            cin>>N;
            int arr[N];
            for(int i = 0 ; i<N;i++){
                cin>>arr[i];
            }
            int max = arr[N-1];
            int arr1[N];
            for(int  i = 0; i<N ; i++){
                if(arr[i] == max){
                    arr1[i] = 0;
                }
                else{
               int count = 0;
```

```
                for(int  j =i+1 ;j<N;j++){
                    if(arr[j]>arr[i])
                    count++;
                }
                arr1[i] = count;
                 }
            }
            for(int i = 0 ; i<N; i++){
                 cout<<arr1[i]<<" ";
            }
        }
        return 0;
}
```

Output :

```
3 2 0 0
4 0 0 0 0
```