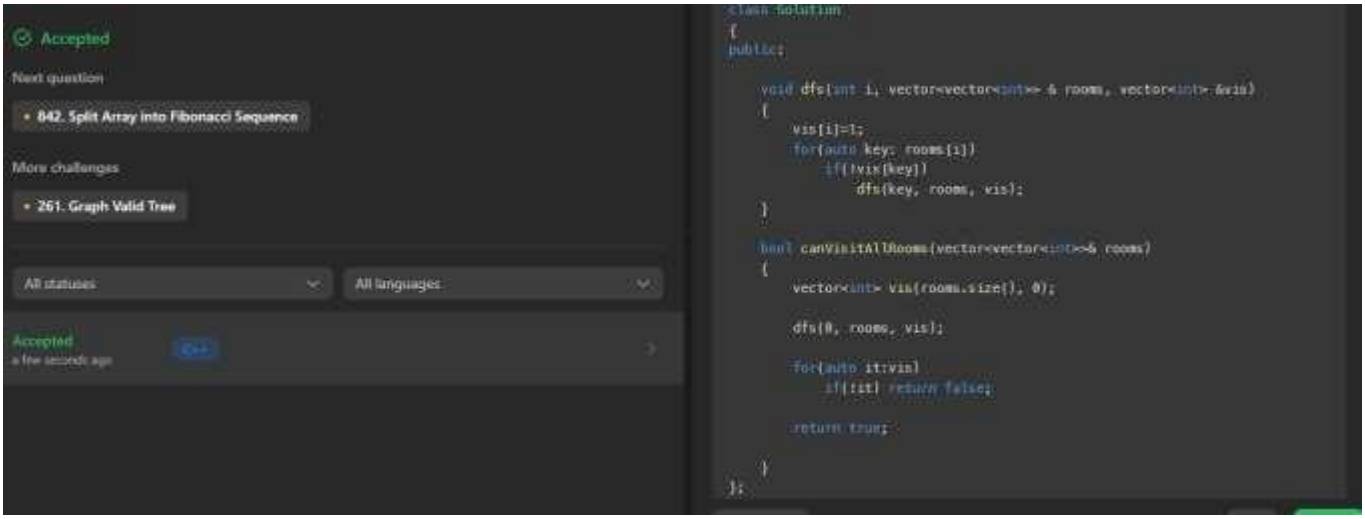
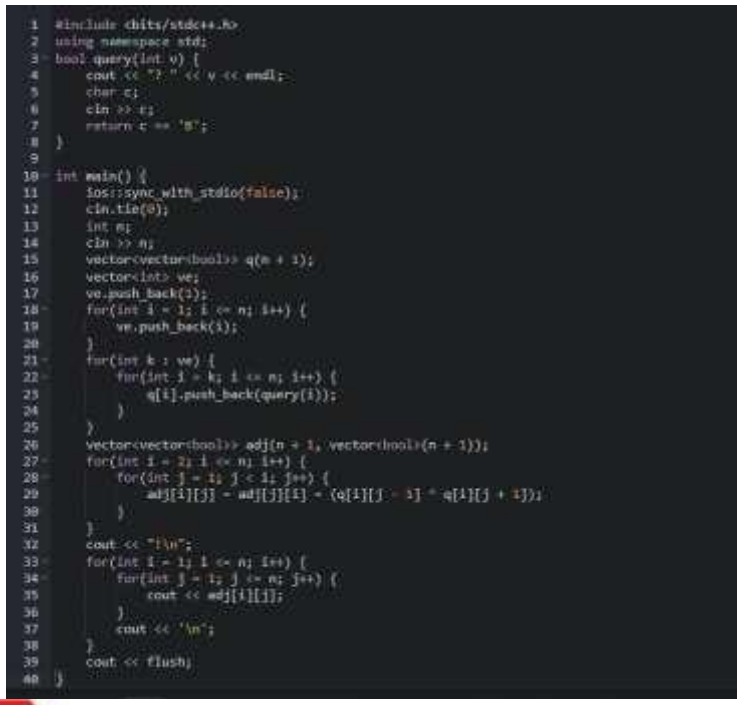
WORKSHEET 7

Student Name: EKROOP SINGH **UID:** 21BCS8143
DOMAIN CAMP: 03-01-2023 to 14-01-2023 **Section/Group:** DWWC-43
Subject Name: IT Skills (DSA)

Question 1. KEYS AND ROOMS



Question 2. HIDDEN COLORED GRAPH



```
#include <bits/stdc++.h>
using namespace std;
bool query(int v) {
    cout << "? " << v << endl;
    char c;
    cin >> c;
    return c == 'B';
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0);
    int n;
    cin >> n;
    vector<vector<bool>> q(n + 1);
    vector<int> ve;
    ve.push_back(1);
    for(int i = 1; i <= n; i++) {
        ve.push_back(i);
    }
    for(int k : ve) {
        for(int i = k; i <= n; i++) {
            q[i].push_back(query(i));
        }
    }
    vector<vector<bool>> adj(n + 1, vector<bool>(n + 1));
    for(int i = 2; i <= n; i++) {
        for(int j = 1; j < i; j++) {
            adj[i][j] = adj[j][i] = (q[i][j - 1] ^ q[i][j + 1]);
        }
    }
    cout << "!\n";
    for(int i = 1; i <= n; i++) {
        for(int j = 1; j <= n; j++) {
            cout << adj[i][j];
        }
        cout << '\n';
    }
    cout << flush;
}
```

SOLUTION:

Question 3. WINTER

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

#define int long long int
#define F first
#define S second
#define pb push_back
#define que_max priority_queue<int>
#define que_min priority_queue<int,vector<int>,greater<int>>
 #define endl "\n"
 using namespace std;

 int32_t main()
 {
 #ifndef ONLINE_JUDGE
     freopen("input.txt","r",stdin);
     freopen("output.txt","w",stdout);
 #endif

int n,m,q1;
cin>>n>>m>>q1;
 vector<vector<int>>vec(n+1);
for(int i=0;i<m;i++)
  {
    int x,y;
    cin>>x>>y;
    vec[x].push_back(y);
    vec[y].push_back(x);
  }

  vector<bool>visited(n+1,false);
  queue<int>q;
  vector<bool>frozen(n+1,false);

  while(q1--)
  {

    int query,type;
    cin>>type>>query;
    if(type==1)
    {
```

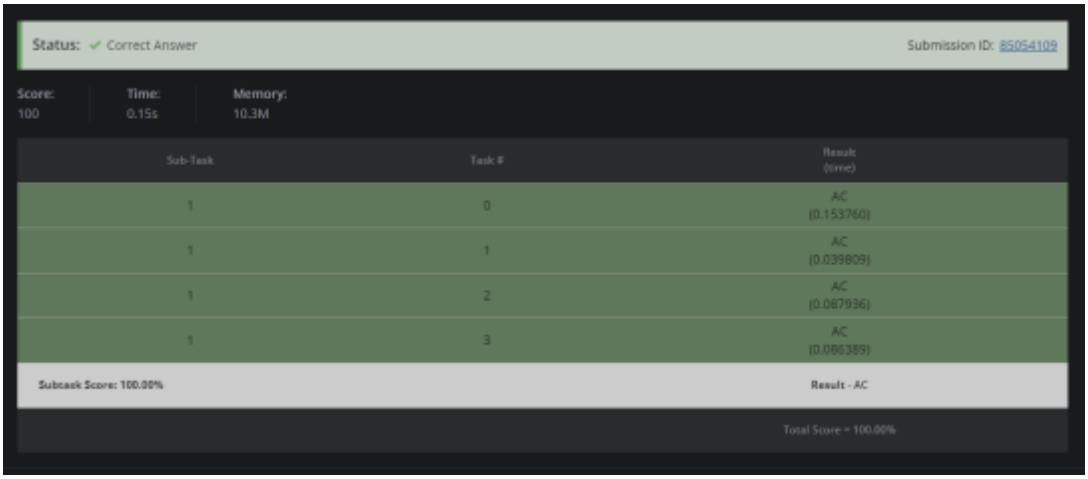
```cpp
        if(frozen[query])continue;
        frozen[query]=true;
        // if(visited[query]==false)
        // {
          //  visited[query]=true;
            q.push(query);
        // }
    }else if(type==2)
    {
        while(q.size()!=0 && query!=0 )
        {
            int sz=q.size();
            while(sz--){
            int tp=q.front();
            visited[tp]=true;
            q.pop();

            for(auto nbr:vec[tp])
            {

                if(!visited[nbr]){
                    if(frozen[nbr]) continue;
                    frozen[nbr]=true;
                    q.push(nbr);
                }
            }
            }query--;
        }
    }else
    {
        if(frozen[query])
        {
            cout<<"Yes"<<endl;
        }else
        {
            cout<<"No"<<endl;
        }

    }
  }
return 0;
```

SOLUTION:

Question 4. MINIMAL TRAVEL TIME

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

#define llint long long int
using namespace std;

void run()
{
    // Insert code here
    int n, m, s, k;
    cin >> n >> m >> s >> k;

    vector<vector<int>> graph(n+1);

    for(int i = 0; i < m; ++i){
        int u, v;
        cin >> u >> v;
        graph[u].push_back(v);
        graph[v].push_back(u);
    }

    std::vector<int> count(n+1);
    for (int i = 0; i < s; ++i){
        int val;
        cin >> val;
        count[val]++;
    }
    vector<bool> vis(n+1);
    queue<int> q;

    q.push(0);
    vis[0] = true;

    llint res = 0, curr = 0;

    while(!q.empty() && k > 0){
        int size = q.size();
        for(int i = 0; i < size; ++i){
            int node = q.front();
            q.pop();
            for(auto adj : graph[node]){
                if(!vis[adj]){
                    vis[adj] = true;
                    q.push(adj);
                }
            }
            int val = min(k, count[node]);
            res += 2*curr*val;
            k -= val;
        }
        curr++;
    }
    cout << res << "\n";
}

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

    int t = 1;
    std::cin >> t;
    while (t--)
        run();

    return 0;
}
```

SOLUTION:



Question 5. CHEF AND REVERSING

```cpp
#include <bits/stdc++.h>
using namespace std;
const int N = 1e5+10;
const int infi=1e9+10;
vector<pair<int,int>>g[N];
vector<int>level(N,infi);
int n,m;
 void bfs(){
     level[1]=0;
     deque<int> dq;
     dq.push_back(1);
     while(!dq.empty()){
         int cur_v= dq.front();
         dq.pop_front();
         for(auto childs:g[cur_v]){
             int child = childs.first;
             int wt = childs.second;
             if(level[cur_v]+wt < level[child]){
             level[child] = level[cur_v] + wt;
             if(wt==1) dq.push_back(child);
             else dq.push_front(child);
         }}
     }
   if(level[n]==infi) cout<<-1 ;
   else cout<<level[n];
 }


int main() {

    cin>>n>>m;
    for(int i=0;i<m;i++){
        int x,y;
        cin>>x>>y;
        if(x==y)continue;
        g[x].push_back({y,0});
        g[y].push_back({x,1});
    }
    bfs();

    return 0;
}
```

SOLUTION:

Question 6. CHEF AND EDGE FLIPPING

Language: C++14

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

#define N 1010

int n, m, a[N], b[N];
bool col[N], s[N][N];

bool check(int u) {
    for (int i = 1; i <= n; i ++) col[i] = 0; col[u] = 1;
    for (int i = 1; i <= n; i ++) if (i != u) s[i][u] = 1, s[u][i] = 0;
    for (int i = 0; i < m; i ++) {
        int x = a[i], y = b[i];
        if ((col[x] ^ col[y]) == 1) {
            if (col[x]) swap(x, y);
            s[x][y] ^= 1, s[y][x] ^= 1;
            col[x] = 1;
            for (int j = 1; j <= n; j ++) if (!col[j]) s[j][x] = 1, s[x][j] = 0;
        }
        else if ((col[x] & col[y]) == 1) s[x][y] ^= 1, s[y][x] ^= 1;
    }
    bool fg = false;
    for (int i = 1; i <= n; i ++) if (!col[i]) fg = true;
    if (!fg) return 0;
    for (int i = 0; i < m; i ++) {
        int x = a[i], y = b[i];
        s[x][y] ^= 1, s[y][x] ^= 1;
    }
    for (int i = 1; i <= n; puts(""), i ++) for (int j = i + 1; j <= n; j ++) printf("%d ", s[i][j]);
    return 1;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T --) {
        scanf("%d %d", &n, &m);
        for (int i = 0; i < m; i ++) scanf("%d %d", &a[i], &b[i]);
        for (int i = 1; i <= n; i ++) if (check(i)) break;
    }

    return 0;
}
```

SOLUTION:

Question 7. MANGO MARKET

Language: C++14

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

using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int n, m;
    cin >> n >> m;
    long long sum = 0;
    for (int i = 1; i <= n; i++) {
        long long x;
        cin >> x;
        sum += x;
    }
    long long edges = (long long)m, unused = ((long long)n * (n - 1)) / 2LL - edges;
    for (int i = 0; i < m; i++) {
        int u, v;
        cin >> u >> v;
    }
    int b=edges-unused;
    int q;
    cin >> q;
    for (int i = 0; i < q; i++) {
        char x;
        cin >> x;
        if (x == '?') {
            cout << sum + edges-unused << '\n';
            continue;
        }
        int u, v;
            cin >> u >> v;
        if (x == '+') {
           edges++;unused--;

        }
        else if (x == '-') {
          edges--;
          unused++;
        }

    }
    return 0;
}
```

SOLUTION:

Status: ✓ Correct Answer — Submission ID: 85054358

Time: 0.04s | Memory: 5.4M

Question 8. ONE MORE WEIRD GAME

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

int main() {
    // your code goes here
    int t,i;
    cin>>t;
    for(i=0;i<t;++i)
    {
        int n,m;
        cin>>n>>m;

        cout<<(n-1)+(m-1)+(2)*(n-1)*(m-1)<<endl;
    }
    return 0;
}
```

SOLUTION:

Status: ✔ Correct Answer — Submission ID: 85054581

Score:	Time:	Memory:
100	0.00s	5.3M

Sub-Task	Task #	Result (time)
1	1	AC (0.003705)
Subtask Score: 30.00%		Result - AC
2	2	AC (0.003869)
Subtask Score: 70.00%		Result - AC
		Total Score = 100.00%