Bài 1: Xếp chỗ ngồi

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

using namespace std;

int main() {
    int r, c;
    cin >> r >> c;

    vector<string> s(r);
    for(int i = 0; i < r; ++i)
        cin >> s[i];

    for(int i = 0; i < r; ++i) {
        for(int j = 0; j < c-2; ++j) {
            if (s[i][j] == '.' && s[i][j+1] == '.' && s[i][j+2] == '.') {
                printf("%d %d\n", i+1, j+1);
                return 0;
            }
        }
    }
}
```

Bài 2: Vận chuyển gạo

```
#pragma GCC optimize("O2")
#include    <bits/stdc++.h>
#define fo(i,a,b) for(int i=(a);i<=(b);++i)
#define __unique(V) (V).resize(unique((V).begin(),(V).end())-(V).begin())
#define cntbit(X)   __builtin_popcount((X))
#define bit(S,i) (((S)>>(i))&1)
#define PI       acos(-1)
#define fi  first
#define se  second
#define LL  long long
#define ii  pair<int,int>
#define iii pair<int,ii>
#define eb  emplace_back
#define lch ((k) << 1)
#define rch ((k) << 1|1)
#define _abs(x) ((x) > 0 ? (x) : -(x))
#define TASK "TLH"
using namespace std;
mt19937_64 rng(chrono::steady_clock::now().time_since_epoch().count());
const int N = 1e6 + 5;
const int base = 2e3;
int T,n,m,c;
ii x[N];
///--------------------------
int     main(){
    ///
        srand(time(NULL));
        ios::sync_with_stdio(0);
        cin.tie(0);cout.tie(0);
    cin>>T;
    while(T--)
    {
        cin>>c>>m>>n;
        for(int i = 1;i <= m;i++)
        {
            int s;
            cin>>s;
            x[i] = {s, 0};
        }
        for(int i = 1;i <= n;i++)
        {
            int s,d;
            cin>>s>>d;
            x[i+m] = {s, d};
        }
        sort(x + 1,x + 1 + m + n);
        long long ans = 0,rice = c;
        for(int i = 1;i <= m + n;i++)
        {
            ans += (x[i].se > 0) * (rice >= x[i].se) * x[i].se;
            if(x[i].se == 0)
                rice = c;
            else
                rice = (rice >= x[i].se ? rice - x[i].se : rice);
        }
        cout<<ans<<'\n';
    }
}
```

Bài 3: Khảo sát dân cư

```
#pragma GCC optimize("O2")
#include    <bits/stdc++.h>
#define fo(i,a,b) for(int i=(a);i<=(b);++i)
#define __unique(V) (V).resize(unique((V).begin(),(V).end())-(V).begin())
#define cntbit(X)    __builtin_popcount((X))
#define bit(S,i) (((S)>>(i))&1)
#define PI       acos(-1)
#define fi  first
#define se  second
#define LL  long long
#define ii  pair<int,int>
#define iii pair<int,ii>
#define eb  emplace_back
#define lch ((k) << 1)
#define rch ((k) << 1|1)
#define _abs(x) ((x) > 0 ? (x) : -(x))
#define TASK "TLH"
using namespace std;
mt19937_64 rng(chrono::steady_clock::now().time_since_epoch().count());
const int N = 1e2 + 5;
const int base = 2e3;
int n,p[N],c[N];
///--------------------------
int     main(){
    ///
        srand(time(NULL));
        ios::sync_with_stdio(0);
        cin.tie(0);cout.tie(0);
    ///
    cin>>n;
    for(int i = 1;i <= n;i++)
        cin>>p[i];
    for(int i = 1;i <= n;i++)
        cin>>c[i];
    int pos = 1;
    for(int i = 2;i <= n;i++)
    {
        if(1LL * c[pos] * p[i] < 1LL * c[i] * p[pos])
            pos = i;
    }
    cout<<1<<'\n'<<pos;
}
```

Bài 4: Kết nối:

```
#pragma GCC optimize("O2")
#include    <bits/stdc++.h>
#define fo(i,a,b) for(int i=(a);i<=(b);++i)
#define __unique(V) (V).resize(unique((V).begin(),(V).end())-(V).begin())
#define cntbit(X)    __builtin_popcount((X))
#define bit(S,i) (((S)>>(i))&1)
#define PI       acos(-1)
#define fi  first
#define se  second
#define LL  long long
#define ii  pair<int,int>
#define iii pair<int,ii>
#define eb  emplace_back
#define lch ((k) << 1)
#define rch ((k) << 1|1)
#define _abs(x) ((x) > 0 ? (x) : -(x))
#define TASK "TLH"
using namespace std;
mt19937_64 rng(chrono::steady_clock::now().time_since_epoch().count());
const int N = 2e2 + 5;
const int base = 2e3;
struct edge
{
    int x,y;
    double cost;
};
int n,par[N],rnk[N];
double w,c,x[N],y[N];
vector<edge>p;
int root(int u)
{
    return (u == par[u] ? u : root(par[u]));
}
int cmp(edge a,edge b)
{
    return a.cost < b.cost;
}
long double distance(int i,int j)
{
    return sqrt((x[i] - x[j]) * (x[i] - x[j]) + (y[i] - y[j]) * (y[i] - y[j]));
}
///--------------------------
int     main(){
    ///
        srand(time(NULL));
        ios::sync_with_stdio(0);
        cin.tie(0);cout.tie(0);
    cin>>n;
    for(int i = 1;i <= n;i++)
        cin>>x[i]>>y[i];
    cin>>w>>c;
    for(int i = 1;i <= n;i++)
    {
        par[i] = i;
        rnk[i] = 1;
        for(int j = i + 1;j <= n;j++)
            p.push_back({i,j,distance(i,j)*c});
    }
    sort(p.begin(),p.end(),cmp);
    double ans = n * w, roadcost = 0;
    for(auto u : p)
    {
        int x = root(u.x);
        int y = root(u.y);
        if(x == y)
            continue;
        if(rnk[x] < rnk[y])
            swap(x,y);
        rnk[x] += rnk[y];
        par[y] = x;
        ans = min(ans, ans + u.cost - w);
        roadcost += u.cost;
    }
    ans = min(ans, roadcost);
    cout<<fixed<<setprecision(9)<<ans;
}
```

Bài 5: Nối vòng tay lớn

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

using namespace std;

typedef long long ll;

int main() {
    int n, p, q;
    cin >> n >> p >> q;

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

    ll sumA = accumulate(a.begin(), a.end(), 0LL);

    function<ll(int)> getEndingPosition = [&] (int x) {
        ll cur = p + sumA * x;
        if (cur <= q) return 1;
        for(int i = 0; i < n; ++i) {
            cur += a[i];
            if (cur <= q) return i+1;
        }
        return -1;
    };

    function<ll()> solve = [&] () {
        if (sumA >= 0) return getEndingPosition(0);
        int l = 0, r = 1e5;
        ll res = 0;
        while (l <= r) {
            int m = (l+r)/2;
            int i = getEndingPosition(m);
            if (i != -1) {
                res = 1LL * m * n + i;
                r = m - 1;
            } else {
                l = m + 1;
            }
        }
        return res;
    };

    cout << solve() << endl;

    return 0;
}
```

Bài 6: Tách xâu

```
//Dai Ca Di Hoc
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#define sz(x) int(x.size())
#define all(x) x.begin(),x.end()
#define reset(x) memset(x, 0,sizeof(x))
#define Rep(i,n) for(int (i)=0;(i)<(int)(n);++(i))
#define For(i,l,u) for(int (i)=(int)(l);(i)<=(int)(u);++(i))
#define MIN(x,y) if (x > (y)) x = (y)
#define MAX(x,y) if (x < (y)) x = (y)
#define PB push_back
#define mp make_pair
#define F first
#define S second
#define maxn 1000005
#define MOD 1000000007
#define remain(x) if (x > MOD) x -= MOD
#define pii pair<int, int>
#define vi vector<int>
#define vii vector< pii >
#define bit(x, i) (((x) >> (i)) & 1)
#define Task "tachtu"

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
_ordered_set;


struct Tries
{
    bool isend;
    Tries *child[26];
    Tries()
    {
        isend = false;
        for (int i = 0; i < 26; i++) child[i] = NULL;
    }
};
```

```
Tries T;
string st;
int m, n;
int Next[maxn];

void Add(string s){
    Tries *p = &T;
    for (int i = 0; i < s.length(); i++)
    {
        int ch = s[i] - 97;
        if (p->child[ch] == NULL) p->child[ch] = new Tries();
        p = p->child[ch];
    }
    p->isend = true;
}

void DP(){
    For(i, 0, m-1) Next[i] = -1;
    Next[m] = 0;
    for (int i = m-1; i >= 0; i--){
        Tries *p = &T;
        for (int j = i; j < m; j++){
            int ch = st[j] - 97;
            if (p->child[ch] == NULL) break;
            p = p->child[ch];
            if (p->isend && Next[j+1] != -1){
                Next[i] = j+1;
                break;
            }
        }
    }
    if (Next[0] == -1) cout << -1 << "\n";
    else {
        int u = 0;
        while (u < m){
            int v = Next[u];
            cout << st.substr(u, v-u);
            u = v;
            if (u == m) cout << "\n";
                else cout << " ";
        }
    }
}

int main()
{
    ios_base::sync_with_stdio(0); cin.tie(0);
    if(fopen(Task".inp", "r")) {
        freopen(Task".inp", "r", stdin);
        freopen(Task".out", "w", stdout);
    }
    cin >> n;
    For(i, 1, n){
        string s;
        cin >> s;
        Add(s);
    }
    int q;
    cin >> q;
    while (q--){
        cin >> st;
        m = st.size();
        DP();
    }
    return 0;
}
```

Bài 7: Trạm phát sóng:

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

using namespace std;

typedef long long ll;

const int MAXN = 200005;

int n, X, Y, x[MAXN], y[MAXN], w[MAXN];

struct Event {
    int x, val;
    bool operator < (const Event &rhs) const {
        return x < rhs.x;
    }
};

int cntCovered(int X, int x[]) {
    vector<Event> events;
    for(int i = 0; i < n; ++i) {
        int l = x[i] - w[i], r = x[i] + w[i];
        events.push_back({max(l, 1), 1});
        if (r < X) events.push_back({r+1, -1});
    }

    sort(events.begin(), events.end());

    int ans = 0, cur = 0, last = 1;

    for(Event e: events) {
        if (cur > 0) ans += e.x - last;
        cur += e.val;
        last = e.x;
    }
    if (cur > 0) ans += X - last + 1;

    return ans;
}

int main() {
    scanf("%d%d%d", &n, &X, &Y);
    for(int i = 0; i < n; ++i) {
        scanf("%d%d%d", &x[i], &y[i], &w[i]);
    }

    int cntX = X - cntCovered(X, x);
    int cntY = Y - cntCovered(Y, y);
    ll ans = 1LL * X * Y - 1LL * cntX * cntY;
    printf("%lld\n", ans);

    return 0;
}
```

Bài 8: Sắp xếp kho hàng

```
#include <algorithm>
#include <iostream>
#include <sstream>
#include <complex>
#include <numeric>
#include <cstring>
#include <vector>
#include <string>
#include <cstdio>
#include <queue>
#include <cmath>
#include <map>
#include <set>

using namespace std;

#define all(a)          (a).begin(),(a).end()
#define sz(a)           int((a).size())
#define FOR(i,a,b)      for(int i=a;i<b;++i)
#define REP(i,n)        FOR(i,0,n)
#define UN(v)           sort(all(v)),(v).erase(unique((v).begin(),(v).end()),(v).end())
#define CL(a,b)                 memset(a,b,sizeof a)
#define pb                      push_back
#define X                       first
#define Y                       second

typedef long long ll;
typedef vector<int> vi;
typedef pair<int,int> pii;

int n, a[5555], s[5555];

ll F[5555], G[5555], *f = F, *g = G;

int main(){
        #ifdef LocalHost
        freopen("x.in","r",stdin);
        freopen("x.out","w",stdout);
        #endif
        cin >> n;
        REP (i, n) cin >> a[i], s[i] = a[i];
        sort(s, s + n);
        REP (i, n) {
                g[0] = f[0] + abs(s[0] - a[i]);
                FOR (j, 1, n)
                        g[j] = min(g[j - 1], f[j] + abs(a[i] - s[j]));
                swap(f, g);
        }
        cout << *min_element(f, f + n) << endl;
        return 0;
}
```

Bài 9: Hỗ trợ khách hàng

```
#include <bits/stdc++.h>
#define sz(x) int(x.size())
#define reset(x) memset(x, 0,sizeof(x))
#define Rep(i,n) for(int (i)=0;(i)<(int)(n);++(i))
#define For(i,l,u) for(int (i)=(int)(l);(i)<=(int)(u);++(i))
#define MIN(x,y) if (x > (y)) x = (y)
#define MAX(x,y) if (x < (y)) x = (y)
#define PB push_back
#define mp make_pair
#define F first
#define S second
#define maxn 305
#define MOD 1000000007
#define remain(x) if (x > MOD) x -= MOD
#define pii pair<int, int>
#define bit(x, i) (((x) >> (i)) & 1)
#define Task "support2"

using namespace std;

typedef long long ll;
typedef long double ld;

int a[maxn][maxn], b[maxn][maxn], p[maxn], q[maxn];
int m, n, k, maxdest, matchX[maxn], used[maxn], cur;

int DFS(int u){
    if (used[u] == cur) return 0;
    used[u] = cur;
    for (int v = 1; v <= k; v++)
        if (b[u][v] <= maxdest)
            if ((matchX[v] == 0) || DFS(matchX[v])){
                matchX[v] = u;
                return 1;
    }
    return 0;
}

bool check(int dist){
    maxdest = dist;
    reset(used);
    reset(matchX);
    for (cur = 1; cur <= k; cur++)
        if (!used[cur]){
            DFS(cur);
    }
    for (int i = 1; i <= k; i++)
        if (matchX[i] == 0) return 0;
    return 1;
}

int main()
{
            ios_base::sync_with_stdio(0); cin.tie(0);
    if(fopen(Task".inp", "r")) {
        freopen(Task".inp", "r", stdin);
        freopen(Task".out", "w", stdout);
    }
    cin >> n >> m >> k;
    memset(a, 50, sizeof(a));
    for (int i = 1; i <= n; i++) a[i][i] = 0;
    for (int u, v, w, i = 1; i <= m; i++)
    {
        cin >> u >> v >> w;
        a[u][v] = a[v][u] = w;
    }
    for (int i = 1; i <= k; i++) cin >> p[i];
    for (int i = 1; i <= k; i++) cin >> q[i];
    for (int r = 1; r <= n; r++)
        for (int u = 1; u <= n; u++)
            for (int v = 1; v <= n; v++)
                a[u][v] = min(a[u][v], a[u][r] + a[r][v]);
    for (int i = 1; i <= k; i++)
        for (int j = 1; j <= k; j++)
            b[i][j] = a[q[i]][p[j]];
    int l = -1, r = 1e9;
    while (r-l > 1){
        int mid = (l+r)/2;
        if (check(mid)) r = mid;
            else l = mid;
    }
    cout << r << "\n";
    bool ok = check(r);
    For(i, 1, k) used[matchX[i]] = i;
    For(i, 1, k) cout << used[i] << " ";
    return 0;
}
```

Bài 10: Robot thông minh