Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	6	5	351

Task A ()

```
//#define DEBUG

#include <bits/stdc++.h>
#define x first
#define y second
#define fi first
#define se second
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define out(t, a) copy(all(a), ostream_iterator<t>(cout, "␣")); cout << "\n";
//#define int long long

using namespace std;

void set_program() {
    #ifdef DEBUG
        freopen("in.txt", "r", stdin);
        freopen("out.txt", "w", stdout);
    #endif // DEBUG
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
}

typedef long long ll;
typedef long double ld;

const ld EPS = 1e-8;
const int LOGN = 20;
const int MOD = 1e9 + 7;
const int INF = 1e9;
const int MAXN = 1e5;

void solve(int num_test) {
    int n;
    cin >> n;
    cout << n - 1 << "\n";
}

signed main() {
    set_program();
    int t;
    #ifdef DEBUG
        t = 2;
    #else
        t = 1;
    #endif // DEBUG
    for (int test = 1; test <= t; ++test) {
        solve(test);
//          cout << "\n";
    }
    return 0;
}
```

Task B ()

```
//#define DEBUG

#include <bits/stdc++.h>
#define x first
#define y second
#define fi first
#define se second
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define out(t, a) copy(all(a), ostream_iterator<t>(cout, "␣")); cout << "\n";
#define ldout cout << fixed << setprecision(15)
//#define int long long

using namespace std;

void set_program() {
    #ifdef DEBUG
        freopen("in.txt", "r", stdin);
        freopen("out.txt", "w", stdout);
    #endif // DEBUG
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
}

typedef long long ll;
typedef long double ld;
//typedef pair <int, int> pt;

const ld EPS = 1e-3;
const ld EPS2 = 1e-8;
const int LOGN = 20;
const int MOD = 1e9 + 7;
const int INF = 1e9;
const int MAXN = 1e5;

struct pt {
    ld x, y;
    pt() {};
    pt(ld x, ld y): x(x), y(y) {};
    pt operator + (pt p) {
        return pt(x + p.x, y + p.y);
    }
    pt operator - (pt p) {
        return pt(x - p.x, y - p.y);
    }
    ld operator ^ (pt p) {
        return x * p.y - y * p.x;
    }
};

void coding(int n) {
    vector <pt> p(n);
    for (auto& i : p) {
        cin >> i.x >> i.y;
    }
    for (int i = 1; i < n - 1; ++i) {
        for (int j = i + 1; j < n; ++j) {
            pt v1 = p[i] - p[i - 1];
            pt v2 = p[j] - p[i - 1];
            if ((v1 ^ v2) > EPS2) {
               swap(p[i], p[j]);
            }
        }
    }
    for (int i = 0; i < n; ++i) {
        if (i % 2 == 0) {
            ldout << p[i].x << "␣" << p[i].y << "\n";
        }
    }
//    cout << "\n";
//    for (auto i : p) {
//        ldout << i.x << " " << i.y << "\n";
//    }
}

ld get_dist(pt a, pt b) {
    return hypot(a.x - b.x, a.y - b.y);
}

ld get_dist(pt a) {
    return hypot(a.x, a.y);
}

void norming(pt& n, ld d) {
    ld nd = get_dist(n);
    n.x = n.x / nd * d;
    n.y = n.y / nd * d;
}

bool is_eq(pt a, pt b) {
    return abs(a.x - b.x) <= EPS && abs(a.y - b.y) <= EPS;
}

void add_point(pt p1, pt p2, pt p3, vector <pt>& p) {
    pt m1;
    m1.x = (p2.x + p3.x) / 2;
    m1.y = (p2.y + p3.y) / 2;
    pt o;
    o.x = (p1.x + 2 * m1.x) / 3;
    o.y = (p1.y + 2 * m1.y) / 3;
//    ldout << m1.x << " " << m1.y << "\n";
    ld d = get_dist(o, m1);
//    ldout << d << "\n";
    pt n;
    n.x = p3.y - p2.y;
    n.y = p2.x - p3.x;
    norming(n, d);
//    ldout << n.x << " " << n.y << "\n";
//    cout << get_dist(n) << "\n";
    pt newp = m1 + n;
    if (!is_eq(o, newp)) {
        p.push_back(newp);
    }
    newp = m1 - n;
    if (!is_eq(o, newp)) {
        p.push_back(newp);
    }
}

void print(vector <pt>& p) {
    ldout << p[0].x << "␣" << p[0].y << "\n";
    ldout << p[4].x << "␣" << p[4].y << "\n";
    ldout << p[2].x << "␣" << p[2].y << "\n";
    ldout << p[3].x << "␣" << p[3].y << "\n";
    ldout << p[1].x << "␣" << p[1].y << "\n";
    ldout << p[5].x << "␣" << p[5].y << "\n";
}

void decoding(int n) {
    vector <pt> p(n);
    for (auto& i : p) {
        cin >> i.x >> i.y;
    }
    add_point(p[0], p[1], p[2], p);
    add_point(p[1], p[2], p[0], p);
    add_point(p[2], p[0], p[1], p);
    print(p);
}

void solve(int num_test) {
    int n;
    cin >> n;
    if (n == 6) {
        coding(n);
    } else {
        decoding(n);
    }
}

signed main() {
    set_program();
    int t;
    #ifdef DEBUG
        t = 1;
    #else
        t = 1;
    #endif // DEBUG
    for (int test = 1; test <= t; ++test) {
        solve(test);
//        cout << "\n";
    }
    return 0;
}
```

Task C ()

```
//#define DEBUG

#include <bits/stdc++.h>
#define x first
#define y second
#define fi first
#define se second
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define out(t, a) copy(all(a), ostream_iterator<t>(cout, "␣")); cout << "\n";
//#define int long long

using namespace std;

void set_program() {
    #ifdef DEBUG
        freopen("in.txt", "r", stdin);
        freopen("out.txt", "w", stdout);
    #endif // DEBUG
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
}

typedef long long ll;
typedef long double ld;

const ld EPS = 1e-8;
const int LOGN = 20;
const int MOD = 1e9 + 7;
const int INF = 1e9;
const int MAXN = 1e5;

void solve(int num_test) {
    string t;
    cin >> t;
    int tsz = t.size();
    int m;
    cin >> m;
    int ans = 0;
    for (; m--; ) {
        string s;
        cin >> s;
        int n = s.size();
        int res = tsz;
        for (int i = 0; i < n; ++i) {

            int ptrs = i, ptrt = 0;
            int len = 0;
            for (; ptrs < n; ++ptrs) {
                for (; ptrt < tsz && s[ptrs] != t[ptrt]; ++ptrt) {};
                ++ptrt;
                if (ptrt > tsz) {
                    break;
                }
                ++len;
            }
            res = min(res, tsz - len);
        }
//        cout << res << "\n";
        ans += res;
    }
    cout << ans << "\n";
}

signed main() {
    set_program();
    int t;
    #ifdef DEBUG
        t = 1;
    #else
        t = 1;
    #endif // DEBUG
    for (int test = 1; test <= t; ++test) {
        solve(test);
//        cout << "\n";
    }
    return 0;
}
```

Task D ()

```
//#define DEBUG

#include <bits/stdc++.h>
#define x first
#define y second
#define fi first
#define se second
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define out(t, a) copy(all(a), ostream_iterator<t>(cout, "␣")); cout << "\n";
//#define int long long

using namespace std;

void set_program() {
    #ifdef DEBUG
        freopen("in.txt", "r", stdin);
        freopen("out.txt", "w", stdout);
    #endif // DEBUG
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
}

typedef long long ll;
typedef long double ld;
typedef pair <int, int> pt;

const ld EPS = 1e-8;
const int LOGN = 20;
const int MOD = 1e9 + 7;
const int INF = 1e9;
const int MAXN = 1e5;

int n, m;
pt s, f;
vector <vector <pt>> a;
vector <vector <int>> used, d;

int get_dist(pt a, pt r, pt b) {
    return abs(a.x - r.x - b.x) + abs(a.y - r.y - b.y);
}

void solve(int num_test) {
    cin >> n >> m;
    cin >> s.x >> s.y >> f.x >> f.y;
    a.clear();
    a.resize(n + 2, vector <pt> (m + 2));
    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m; ++j) {
            cin >> a[i][j].x >> a[i][j].y;
        }
    }
    used.clear();
    used.assign(n + 2, vector <int> (m + 2, 0));
    d.assign(n + 2, vector <int> (m + 2, INF));
    used[s.x][s.y] = 1;
    d[s.x][s.y] = 0;
    set <pair <int, pt>> q;
    q.insert({d[s.x][s.y], s});
//    int cnt = 0;
    for (; q.size(); ) {
//        if (++cnt > 5) {
//            break;
//        }
        pt cur = q.begin()->se;
        q.erase(q.begin());
//        cout << cur.x << " " << cur.y << "\n";
//        cout << d[cur.x][cur.y] << "\n";
        if (cur == f) {
            break;
        }
        for (int i = 1; i <= n; ++i) {
            for (int j = 1; j <= m; ++j) {
                if (!used[i][j]) {
                    pt next = {i, j};
                    int dist = get_dist(next, a[cur.x][cur.y], cur);
                    assert(dist >= 0);
                    if (d[cur.x][cur.y] + dist < d[i][j]) {
                        q.erase({d[i][j], next});
                        d[i][j] = d[cur.x][cur.y] + dist;
                        q.insert({d[i][j], next});
                    }
                }
            }
        }
    }
    cout << d[f.x][f.y] << "\n";
}

signed main() {
    set_program();
    int t;
    #ifdef DEBUG
        t = 2;
    #else
        t = 1;
    #endif // DEBUG
    for (int test = 1; test <= t; ++test) {
        solve(test);
//        cout << "\n";
    }
    return 0;
}
```

Task E ()

```
//#define DEBUG

#include <bits/stdc++.h>
#define x first
#define y second
#define fi first
#define se second
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define out(t, a) copy(all(a), ostream_iterator<t>(cout, "␣")); cout << "\n";
#define ldout cout << fixed << setprecision(15)
//#define int long long

using namespace std;

void set_program() {
    #ifdef DEBUG
        freopen("in.txt", "r", stdin);
        freopen("out.txt", "w", stdout);
    #endif // DEBUG
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
}

typedef long long ll;
typedef long double ld;
typedef pair <int, int> pt;

const ld EPS = 1e-3;
const ld EPS2 = 1e-8;
const int LOGN = 20;
const int MOD = 1e9 + 7;
const int INF = 1e9;
const int MAXN = 1e5;

pt ask(pt a, pt b) {
    cout << "?␣" << a.x << "␣" << a.y << "␣" << b.x << "␣" << b.y << endl;
    pt res;
    cin >> res.x >> res.y;
    return res;
}

void guess(pt crd, pt in, int n, int m) {
    cout << "!␣" << in.x - crd.x + 1 << "␣" << in.y - crd.y + 1 << endl;
    exit(0);
}

void solve1(int n, int m, int b, vector <pt>& a) {
    pt p1 = {0, 0};
    pt p2 = {100, 100};
    pt res = ask(p1, p2);
    if (res != p1) {
        guess(a[0], p1, n, m);
    } else {
        guess(a[0], p2, n, m);
    }
}

pair <pt, pt> solve2(int n, int m, int bb, vector <pt>& arr, int sx, int sy)  {
    pt p1, p2;
    p1 = {sx + 0, sy + 0};
    p2 = {sx + 100, sy + 100};
    pt res1 = ask(p1, p2);
    pt p3, p4;
    p3 = {sx + 200, sy + 200};
    p4 = {sx + 300, sy + 300};
    pt res2 = ask(p3, p4);
    pt a;
    pt b;
    if (res1 != p1) {
        a = p1;
    } else {
        a = p2;
    }
    if (res2 != p3) {
        b = p3;
    } else {
        b = p4;
    }
    int dx = arr[1].x - arr[0].x;
    int dy = arr[1].y - arr[0].y;
    pt c, d;
    c = {a.x + dx, a.y + dy};
    d = {b.x + dx, b.y + dy};
    pt res3 = ask(c, d);
//    pt luloc;
//    luloc.x = min(arr[0].x, arr[1].x);
//    luloc.y = min(arr[0].y, arr[1].y);
    if (res3 != c) {
        return {a, c};
//        pt lu;
//        lu.x = min(a.x, c.x);
//        lu.y = min(a.y, c.y);
//        guess(luloc, lu, n, m);
    } else {
        return {b, d};
//        pt lu;
//        lu.x = min(b.x, d.x);
//        lu.y = min(b.y, d.y);
//        guess(luloc, lu, n, m);
    }
}

void solve(int num_test) {
    int n, m, b;
    cin >> n >> m >> b;
    vector <pt> a(b);
    for (auto& i : a) {
        cin >> i.x >> i.y;
    }
    if (b == 1) {
        solve1(n, m, b, a);
        exit(0);
    }
    if (b == 2) {
        pair <pt, pt> res = solve2(n, m, b, a, 0, 0);
        pt p1 = res.fi;
        pt p2 = res.se;
//        exit(0);
        pt lu;
        lu.x = min(p1.x, p2.x);
        lu.y = min(p1.y, p2.y);
        pt luloc;
        luloc.x = min(a[0].x, a[1].x);
        luloc.y = min(a[0].y, a[1].y);
        guess(luloc, lu, n, m);
        exit(0);
    }
}

signed main() {
    set_program();
    int t;
    #ifdef DEBUG
        t = 1;
    #else
        t = 1;
    #endif // DEBUG
    for (int test = 1; test <= t; ++test) {
        solve(test);
//        cout << "\n";
    }
    return 0;
}
```

Task F ()

```
//#define DEBUG

#include <bits/stdc++.h>
#define x first
#define y second
#define fi first
#define se second
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define out(t, a) copy(all(a), ostream_iterator<t>(cout, "␣")); cout << "\n";
#define ldout cout << fixed << setprecision(15)
#define int long long

using namespace std;

void set_program() {
    #ifdef DEBUG
        freopen("in.txt", "r", stdin);
        freopen("out.txt", "w", stdout);
    #endif // DEBUG
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
}

typedef long long ll;
typedef long double ld;
typedef pair <int, int> pt;

const ld EPS = 1e-3;
const ld EPS2 = 1e-8;
const int LOGN = 20;
const int MOD = 1e9 + 7;
const int INF = 1e9;
const int MAXN = 26;

int n;
int gr[MAXN][MAXN];
int deg[MAXN];

int dfs(int u, int parent, int depth) {
    int res = depth;
    for (int v = 0; v < n; ++v) {
        if (gr[u][v] && v != parent) {
            res += dfs(v, u, depth + 1);
        }
    }
    return res;
}

int bit(int mask, int i) {
    return ((mask >> i) & 1);
}

void print_gr(int n) {
    for (int i = 0; i < n; ++i) {
        cout << i + 1 << ":␣";
        for (int j = 0; j < n; ++j) {
            if (gr[i][j]) {
                cout << j + 1 << "␣";
            }
        }
        cout << endl;
    }
    cout << endl;
}

int col[MAXN];
bool cycle;

void find_cycle(int u, int parent = -1) {
    col[u] = 1;
    for (int v = 0; v < n; ++v) {
        if (gr[u][v] && col[v] == 1 && v != parent) {
            cycle = 1;
        }
        if (gr[u][v] && col[v] == 0) {
            find_cycle(v, u);
        }
    }
    col[u] = 2;
}

void solve(int num_test) {
//    int n;
//    cin >> n;
//    cout << (n * n * n - n / 6) << "\n";
//    return;
    int m;
    cin >> n >> m;
    int sz = n * n;
    vector <int> ans(m + 1, 0);
//    return;
    for (int mask = 0; mask < (1 << sz); ++mask) {
        if (__builtin_popcount(mask) != n - 1) {
            continue;
        }
        memset(gr, 0, sizeof(gr));
        memset(deg, 0, sizeof(deg));
        bool bad = 0;
        for (int i = 0; i < sz; ++i) {
            if (bit(mask, i)) {
                int u = i / n;
                int v = i % n;
                if (u >= v) {
                    bad = 1;
                    break;
                }
                gr[u][v] = gr[v][u] = 1;
                ++deg[u], ++deg[v];
            }
        }
        for (int i = 0; i < n; ++i) {
            if (!deg[i]) {
                bad = 1;
                break;
            }
        }
        memset(col, 0, sizeof(col));
        cycle = 0;
        for (int i = 0; i < n; ++i) {
            if (!col[i]) {
                find_cycle(i);
            }
        }
        if (bad || cycle) {
            continue;
        }
//        cout << mask << endl;
//        cout << "ok" << endl;
//        print_gr(n);
        int res = 0;
        for (int i = 0; i < n; ++i) {
            res += dfs(i, -1, 0);
        }
//        cout << "res = " << res / 2 << endl;
        ++ans[res / 2];
    }
//    cout << "ans: ";
    for (int i = 1; i <= m; ++i) {
        cout << ans[i] << "␣";
    }
    cout << "\n";
}

signed main() {
    set_program();
    int t;
    #ifdef DEBUG
        t = 6;
    #else
        t = 1;
    #endif // DEBUG
    for (int test = 1; test <= t; ++test) {
        solve(test);
//        cout << "\n";
    }
    return 0;
}
```