Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	100	0	500

Task A ()

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

using namespace std;

int main()
{
    #ifdef HOME
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif // HOME

    int n;
    cin >> n;
    cout << n - 1;
}
```

Task B ()

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

using namespace std;
typedef long long ll;
#define f first
#define s second
#define mp(a, b) make_pair(a, b)
#define all(a) a.begin(), a.end()
#define size(a) (ll)a.size()

double get(pair<double, double> a, pair<double, double> b) {
    return sqrt((a.f − b.f) ∗ (a.f − b.f) + (a.s − b.s) ∗ (a.s − b.s));
}

int main()
{
    #ifdef HOME
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif // HOME

    int n;
    cin >> n;

    vector<pair<double, double>> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i].f >> a[i].s;
    }

    cout << fixed << setprecision(10) << "";
    if (n == 6) {
        vector<pair<double, double>> b(n);
        b[0] = a[0];
        vector<bool> used(n, 0);
        used[0] = 1;
        int last = 0;
        for (int i = 1; i < n; i++) {
            vector<pair<double, int>> v;
            for (int j = 0; j < n; j++) {
                if (!used[j]) {
                    v.push_back(mp(get(a[last], a[j]), j));
                }
            }
            sort(all(v));
            if (!v.empty()) {
                if (used[v[0].s]) {
                    if (size(v) > 1) {
                        b[i] = a[v[1].s];
                        last = v[1].s;
                        used[last] = 1;
                    }
                } else {
                    b[i] = a[v[0].s];
                    last = v[0].s;
                    used[last] = 1;
                }
            }
        }
        a = b;

//        for (int i = 0; i < n; i++) {
//            cout << a[i].f << " " << a[i].s << "\n";
//        }
//        cout << "\n";

        pair<double, double> o;
        o.f = (a[0].f + a[3].f) / 2;
        o.s = (a[0].s + a[3].s) / 2;
        cout << o.f << "␣" << o.s << "\n";
        cout << a[0].f << "␣" << a[0].s << "\n";
        cout << a[1].f << "␣" << a[1].s << "\n";
    } else {
        pair<double, double> o = a[0], A = a[1], B = a[2];
        cout << a[1].f << "␣" << a[1].s << "\n";
        cout << a[2].f << "␣" << a[2].s << "\n";
        pair<double, double> c, d, e, f;
        d.f = o.f - A.f, d.s = o.s - A.s;
        c.f = B.f + d.f, c.s = B.s + d.s;
        d.f += o.f, d.s += o.s;
        e.f = o.f - B.f, e.s = o.s - B.s;
        f.f = A.f + e.f, f.s = A.s + e.s;
        e.f += o.f, e.s += o.s;
        cout << c.f << "␣" << c.s << "\n";
        cout << d.f << "␣" << d.s << "\n";
        cout << e.f << "␣" << e.s << "\n";
        cout << f.f << "␣" << f.s << "\n";
    }
}
```

Task C ()

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

using namespace std;
typedef long long ll;
#define f first
#define s second
#define mp(a, b) make_pair(a, b)
#define all(a) a.begin(), a.end()
#define size(a) (ll)a.size()

int main()
{
    #ifdef HOME
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif // HOME

    string t;
    cin >> t;

    int n;
    cin >> n;

    int ans = 0;
    while (n--) {
        string s;
        cin >> s;
        int now = size(t);
        for (int i = 0; i < size(s); i++) {
            int pos = 0;
            int j = i;
            int cnt = 0;
            while (pos < size(t)) {
                if (t[pos] == s[j]) {
                    pos++, j++;
                } else {
                    cnt++;
                    pos++;
                }
            }
            now = min(now, cnt);
        }
        ans += now;
    }

    cout << ans;
}
```

Task D ()

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

using namespace std;
typedef long long ll;
#define f first
#define s second
#define mp(a, b) make_pair(a, b)
#define all(a) a.begin(), a.end()
#define size(a) (ll)a.size()

int main()
{
    #ifdef HOME
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif // HOME

    int n, m, x1, y1, x2, y2;
    cin >> n >> m >> x1 >> y1 >> x2 >> y2;
    x1--, y1--, x2--, y2--;

    vector<vector<pair<int, int>>> a(n, vector<pair<int, int>> (m));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> a[i][j].f >> a[i][j].s;
        }
    }

    const int INF = 1e9;
    int dist[n][m][2];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            dist[i][j][0] = INF;
            dist[i][j][1] = INF;
        }
    }

    set<pair<int, pair<int, pair<int, int>>>> q;
    q.insert(mp(0, mp(0, mp(x1, y1))));
    dist[x1][y1][0] = 0;

    const int nx[] = {-1, -1, 0, 1, 1, 1, 0, -1};
    const int ny[] = {0, 1, 1, 1, 0, -1, -1, -1};

    while (!q.empty()) {
        int x = q.begin()->s.s.f, y = q.begin()->s.s.s, k = q.begin()->s.f;
        q.erase(q.begin());

        for (int i = 0; i < 8; i++) {
            int dx = x + nx[i], dy = y + ny[i];
            if (0 <= dx && dx < n && 0 <= dy && dy < m) {
                int cost;
                if (k == 0) {
                    cost = abs(a[x][y].f - nx[i]) + abs(a[x][y].s - ny[i]);
                } else {
                    cost = abs(nx[i]) + abs(ny[i]);
                }
                if (a[x][y].f == nx[i] && a[x][y].s == ny[i]) cost = 0;
                int kk = (cost ? 1 : 0);
                if (dist[dx][dy][kk] > dist[x][y][k] + cost) {
                    q.erase(mp(dist[dx][dy][kk], mp(kk, mp(dx, dy))));
                    dist[dx][dy][kk] = dist[x][y][k] + cost;
                    q.insert(mp(dist[dx][dy][kk], mp(kk, mp(dx, dy))));
                }
                if (cost == 0) {
                    for (int j = 0; j < 8; j++) {
                        if (nx[i] == -nx[j] && ny[i] == -ny[j]) continue;
                        int ddx = dx + nx[j], ddy = dy + ny[j];
                        if (0 <= ddx && ddx < n && 0 <= ddy && ddy < m) {
                            kk = 1;
                            cost = abs(nx[j]) + abs(ny[j]);
                            if (dist[ddx][ddy][kk] > dist[x][y][k] + cost) {
                                q.erase(mp(dist[ddx][ddy][kk], mp(kk, mp(ddx, ddy))));
                                dist[ddx][ddy][kk] = dist[x][y][k] + cost;
                                q.insert(mp(dist[ddx][ddy][kk], mp(kk, mp(ddx, ddy))));
                            }
                        }
                    }
                }
            }
        }
    }

//    for (int i = 0; i < n; i++) {
//        for (int j = 0; j < m; j++) cout << dist[i][j][0] << " ";
//        cout << "\n";
//    }
//    cout << "\n";
//    for (int i = 0; i < n; i++) {
//        for (int j = 0; j < m; j++) cout << dist[i][j][1] << " ";
//        cout << "\n";
//    }

    cout << min(dist[x2][y2][0], dist[x2][y2][1]);
}
```

Task E ()

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

using namespace std;
typedef long long ll;
#define f first
#define s second
#define mp(a, b) make_pair(a, b)
#define all(a) a.begin(), a.end()
#define size(a) (ll)a.size()

int main()
{
//    #ifdef HOME
//        freopen("input.txt", "r", stdin);
//        freopen("output.txt", "w", stdout);
//    #endif // HOME

    ll n, m, b;
    cin >> n >> m >> b;

    vector<pair<ll, ll>> a(b);
    for (int i = 0; i < b; i++) {
        cin >> a[i].f >> a[i].s;
        a[i].f--, a[i].s--;
    }

    sort(all(a));
    set<ll> alive;
    map<pair<ll, ll>, ll> id;
    map<ll, pair<ll, ll>> ptr;
    ll x = a[0].f, y = a[0].s;
    ll cnt = 0;
    for (int i = 0; i < (1 << (b - 1)); i++) {
        cout << "?␣" << x << "␣" << y << "␣" << x + n << "␣" << y << endl;
        id[mp(x, y)] = cnt;
        ptr[cnt] = mp(cnt * n, 0);
        alive.insert(cnt);
        cnt++;
        id[mp(x + n, y)] = cnt;
        ptr[cnt] = mp(cnt * n, 0);
        alive.insert(cnt);
        cnt++;
        x += n + n;
        ll kx, ky;
        cin >> kx >> ky;
        ll kid = id[mp(kx, ky)];
        alive.erase(kid);
    }

    for (int i = 1; i < b; i++) {
        set<ll> add = alive;
        for (int j = 0; j < (1 << (b - i - 1)); j++) {
            ll id1 = *add.begin();
            add.erase(add.begin());
            ll id2 = *add.begin();
            add.erase(add.begin());
            ll x1 = ptr[id1].f + a[i].f, y1 = ptr[id1].s + a[i].s;
            ll x2 = ptr[id2].f + a[i].f, y2 = ptr[id2].s + a[i].s;
            cout << "?␣" << x1 << "␣" << y1 << "␣" << x2 << "␣" << y2 << endl;
            id[mp(x1, y1)] = id1;
            id[mp(x2, y2)] = id2;
            ll kx, ky;
            cin >> kx >> ky;
            ll kid = id[mp(kx, ky)];
            alive.erase(kid);
        }
    }

    ll ans = *alive.begin();
    cout << "!␣" << ptr[ans].f << "␣" << ptr[ans].s << endl;
}
```

Task F ()