

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	20	6	0	326

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cmath>
#include <math.h>
#include <vector>
#include <map>
#include <set>
#include <string>
#include <queue>
#include <algorithm>

using namespace std;

void files(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
}

#define ll long long
#define mp make_pair
#define rt return
#define all(a) a.begin(), a.end()
#define pii pair<int, int>
#define pll pair<long long, long long>

const int INF = 2e9;
const long long SuperINF = 2e18, MOD = 1e9 + 7;
const double eps = 1e-6;

/*
vector<int> z_func(string &s){
    int n = s.size();

    vector<int> z(n);
    z[0] = n;
    int l = 0, r = 0;

    for (int i = 0; i < n; i++){
        if (i <= r){
            z[i] = min(r - i + 1, z[i - l]);
        }

        while (i + z[i] < n && s[z[i]] == s[z[i] + i]){
            z[i]++;
        }

        if (i + z[i] > r){
            r = i + z[i] - 1;
            l = i;
        }
    }
}
```

```

        return z;
    }
    const long long BASE = 67;
    const int MAX_BASE = 1e6;
    ll BASES[MAX_BASE];

    struct MyHash{
        vector<ll> pref;
        string s;

        MyHash(string &st){
            pref.push_back(0);
            for (int i = 0; i < st.size(); i++){
                ll sums = ((pref.back() * BASE) % MOD + st[i] - '0' + 1) % MOD;
                pref.push_back(sums);
            }

            ll get(int l, int r){
                return ((pref[r + 1] - (pref[l] * BASES[r - l + 1])%MOD) % MOD + MOD) % MOD;
            }
        };

    /*
    const int MAX_SIZE = 1e6;

    int main(){
        std::ios::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        //files();
        ll n;
        cin >> n;
        ll a = (n / 2);

        while ((n - a) / 2 > 0){
            a += (n - a) / 2;
        }

        cout << a;

    }

```

Task B ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cmath>
#include <math.h>
#include <vector>
#include <map>
#include <set>
#include <string>
#include <queue>
#include <algorithm>

using namespace std;

void files(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
}

#define ll long long
#define mp make_pair
#define rt return
#define all(a) a.begin(), a.end()
#define pii pair<int, int>
#define pll pair<long long, long long>

const int INF = 2e9;
const long long SuperINF = 2e18, MOD = 1e9 + 7;
const double eps = 1e-6;

/*
vector<int> z_func(string &s){
    int n = s.size();

    vector<int> z(n);
    z[0] = n;
    int l = 0, r = 0;

    for (int i = 0; i < n; i++){
        if (i <= r){
            z[i] = min(r - i + 1, z[i - l]);
        }

        while (i + z[i] < n && s[z[i]] == s[z[i] + i]){
            z[i]++;
        }

        if (i + z[i] > r){
            r = i + z[i] - 1;
            l = i;
        }
    }
    return z;
}

const long long BASE = 67
const int MAX_BASE = 1e6;
ll BASES[MAX_BASE];

struct MyHash{
    vector<ll> pref;
    string s;

    MyHash(string &st){
        pref.push_back(0);
        for (int i = 0; i < st.size(); i++){
            ll sums = ((pref.back() * BASE) % MOD + st[i] - '0' + 1) % MOD;
            pref.push_back(sums);
        }
    }
}
```

```

    ll get(int l, int r){
        return ((pref[r + 1] - (pref[l] * BASES[r - l + 1])%MOD) % MOD + MOD) % MOD;
    }
};

*/
const int MAX_SIZE = 1e6;

struct Vect{
    long double xx, yy;
    Vect(double a, double b){
        xx = a;
        yy = b;
    }
};

Vect operator+(const Vect a, const Vect b){
    return Vect(a.xx + b.xx, a.yy + b.yy);
}

Vect operator*(const Vect a, double k){
    return Vect(a.xx * k, a.yy * k);
}

void solve6(){
    vector<pair<double, double>> cords;
    double px = 0, py = 0;

    for (int i = 0; i < 6; i++){
        double xx1, yy1;
        cin >> xx1 >> yy1;
        cords.push_back(mp(xx1, yy1));
        px += xx1;
        py += yy1;
    }

    sort(all(cords));
    px /= 6;
    py /= 6;

    vector<pair<double, pair<double, double>>> sorted;

    double xx1 = cords[0].first - px, yy1 = cords[0].second - py;

    for (int i = 0; i < 6; i++){
        double xx2 = cords[i].first - px, yy2 = cords[i].second - py;
        double scalar = xx2 * xx1 + yy2 * yy1;
        double product = xx1 * yy2 - xx2 * yy1;
        double fi = atan2(product, scalar);
        sorted.push_back(mp(fi, cords[i]));
    }

    sort(all(sorted));
    cout.precision(35);
    cout << sorted[0].second.first << "\n" << sorted[0].second.second << "\n";
    cout << sorted[1].second.first << "\n" << sorted[1].second.second << "\n";
    cout << sorted[3].second.first << "\n" << sorted[3].second.second << "\n";
}

void solve3(){
    vector<pair<double, double>> arr;
    for (int i = 0; i < 3; i++){
        double xx, yy;
        cin >> xx >> yy;
        arr.push_back(mp(xx, yy));
    }
}

```

```

}

vector<pair<double, double>> ans(6);
ans[0] = arr[0];
ans[1] = arr[1];
ans[3] = arr[2];

double f1_x = arr[1].first - arr[0].first, f1_y = arr[1].second - arr[0].second;
double d1_x = arr[2].first - arr[0].first, d1_y = arr[2].second - arr[0].second;
double a3_x = d1_x - f1_x, a3_y = d1_y - f1_y;
double px = (arr[0].first + arr[2].first) / 2, py = (arr[0].second + arr[2].second) / 2;
d1_x /= 2;
d1_y /= 2;

ans[2] = mp(arr[0].first + d1_x + f1_x, arr[0].second + d1_y + f1_y);

double v2_x = arr[1].first - px, v2_y = arr[1].second - py;
v2_x *= -1;
v2_y *= -1;

ans[4] = mp(px + v2_x, py + v2_y);

double v3_x = ans[2].first - px, v3_y = ans[2].second - py;
v3_x *= -1;
v3_y *= -1;

ans[5] = mp(px + v3_x, py + v3_y);

cout.precision(35);
for (auto x : ans)
    cout << x.first << " " << x.second << "\n";
}

int main() {
    std::ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    //files();
    int n;
    cin >> n;
    if (n == 6)
        solve6();
    else
        solve3();
}

```

Task C ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cmath>
#include <math.h>
#include <vector>
#include <map>
#include <set>
#include <string>
#include <queue>
#include <algorithm>

using namespace std;

void files(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
}

#define ll long long
#define mp make_pair
#define rt return
#define all(a) a.begin(), a.end()
#define pii pair<int, int>
#define pll pair<long long, long long>

const int INF = 2e9;
const long long SuperINF = 2e18, MOD = 1e9 + 7;
const double eps = 1e-6;

int get_solve(string &s, string &t){
    int maxx = t.size();
    int n = s.size();
    for (int i = 0; i < n; i++){
        int l1 = 0, l2 = 0;
        int j = 0;
        int c = 0;
        int cnt_bad = 0;
        while (1){
            if (i + l1 < n && l2 < t.size() && s[i + l1] == t[l2]){
                c++;
                l1++;
                l2++;
            }
            else{
                while (l2 < t.size() && s[i + l1] != t[l2]){
                    cnt_bad++;
                    l2++;
                }
            }
            if (l2 >= t.size() || i + l1 >= n)
                break;
        }
        maxx = min(maxx, (int)t.size() - l1);
    }
    return maxx;
}

void solve(){
    string obr;
    cin >> obr;
```

```

    int n;
    cin >> n;
    int ans = 0;

    for (int i = 0; i < n; i++){
        string s;
        cin >> s;
        ans += get_solve(s, obr);
    }

    cout << ans;
}

int main(){
    std::ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    //files();

    solve();
}

```

Task D ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cmath>
#include <math.h>
#include <vector>
#include <map>
#include <set>
#include <string>
#include <queue>
#include <algorithm>

using namespace std;

void files(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
}

#define pragma Optimize("se4,_loops-noroll")

#define ll long long
#define mp make_pair
#define rt return
#define all(a) a.begin(), a.end()
#define pii pair<int, int>
#define pll pair<long long, long long>

const int INF = 1000;
const long long SuperINF = 2e18, MOD = 1e9 + 7;
const double eps = 1e-6;

const int MAX_SIZE = 50;

int n, m;
int dist[MAX_SIZE][MAX_SIZE][MAX_SIZE][MAX_SIZE], dy[MAX_SIZE][MAX_SIZE], dx[MAX_SIZE][MAX_SIZE];

void solve(){
    cin >> n >> m;
    /*
    if (n == 3 && m == 3){
        cout << 1;
        return;
    }

    if (n == 3 && m == 5){
        cout << 4;
        return;
    }
    */

    int s_x, s_y, f_x, f_y;
    cin >> s_y >> s_x >> f_y >> f_x;

    s_x--;
    s_y--;
    f_y--;
    f_x--;

    for (int i = 0; i < n; ++i){
        for (int j = 0; j < m; ++j){
            cin >> dy[i][j] >> dx[i][j];
        }
    }

    for (int i = 0; i < n; ++i){
        for (int j = 0; j < m; ++j){
            for (int i1 = 0; i1 < n; ++i1)
```

```

        for (int j1 = 0; j1 < m; ++j1)
            dist[i][j][i1][j1] = INF;
    }

    for (int i = 0; i < n; ++i){
        for (int j = 0; j < m; ++j)
            dist[i][j][i][j] = 0;
    }

    for (int i = 0; i < n; ++i){
        for (int j = 0; j < m; ++j){

            bool flag = false;
            for (int i_s = 0; i_s < n; ++i_s){
                for (int j_s = 0; j_s < m; ++j_s){

                    for (int i_f = 0; i_f < n; ++i_f){
                        for (int j_f = 0; j_f < m; ++j_f){

                            if (i_s == i_f && j_s == j_f)
                                continue;

                            /*
                            ll ty = i_s + dy[i_s][j_s];
                            ll tx = j_s + dx[i_s][j_s];

                            ll will_ans = abs(ty - i) + abs(tx - j);

                            ll ty2 = i + dy[i][j];
                            ll tx2 = j + dx[i][j];

                            will_ans += abs(ty2 - i_f) + abs(tx2 - j_f
                                );
                            */

                            int will_ans = 0;
                            int ty = i_s + dy[i_s][j_s];
                            int tx = j_s + dx[i_s][j_s];
                            will_ans = abs(i_f - ty) + abs(j_f - tx);

                            will_ans = min(will_ans, dist[i_s][j_s][i
                                ][j] + dist[i][j][i_f][j_f]);
                            if (will_ans < dist[i_s][j_s][i_f][j_f]){
                                dist[i_s][j_s][i_f][j_f] =
                                    will_ans;
                            }
                            //dist[i_s][j_s][i_f][j_f] = min(dist[i_s
                                ][j_s][i_f][j_f], );
                        }
                    }
                }
            }
        }

        cout << dist[s_y][s_x][f_y][f_x];
    }

    int main(){
        std::ios::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
    }
}

```

```
// files();  
solve();
```

```
}
```

Task E ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <cmath>
#include <math.h>
#include <vector>
#include <map>
#include <set>
#include <string>
#include <queue>
#include <algorithm>

using namespace std;

void files(){
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
}

#define pragma Optimize("se4, _loops-noroll")

#define ll long long
#define mp make_pair
#define rt return
#define all(a) a.begin(), a.end()
#define pii pair<int, int>
#define pll pair<long long, long long>

const int INF = 1000;
const long long SuperINF = 2e18, MOD = 1e9 + 7;
const double eps = 1e-6;

const int MAX_SIZE = 50;

/*
ll MAX_N = 1e18, MAX_M = 1e18;

vector<pii> blacks;
set<ll> good;
map<ll, bool> bad;
int n, m, b;
ll H_SZ, W_SZ;
ll YY = -1e17, XX = -1e17;

void draw(ll id, pll cord){
    ll i = id / W_SZ;
    ll j = id % W_SZ;

    ll yy = YY + (i * n) + cord.first;
    ll xx = XX + j * m + cord.second;

    cout << yy << " " << xx;
}

ll to_id(ll ii, ll jj){
    ll yy = ii / n;
    ll xx = jj / m;

    return yy * W_SZ + xx;
}

void made_bad(ll id){
    bad[id] = 1;
}
```

```

        if (good.count(id) != 0){
            good.erase(id);
        }
    }

pair<ll, ll> to_cord(ll id_win){
    ll ii = id_win / W_SZ;
    ll jj = id_win % W_SZ;

    ll cord_yy = ii * n + YY;

    ll cord_xx = jj * m + XX;
    return mp(cord_yy, cord_xx);
}

void solve(){

    cin >> n >> m >> b;

    W_SZ = 2e18 / m;
    H_SZ = 2e18 / n;

    for (int i = 0; i < b; i++){
        ll ii, jj;
        cin >> ii >> jj;
        blacks.push_back(mp(ii - 1, jj - 1));
    }

    for (ll cnts = 0; cnts < (1 << 12); cnts++){
        while (bad[cnts]){
            cnts++;
        }

        cout << "? ";
        draw(cnts, blacks[0]);
        good.insert(cnts);
        cout << " ";
        cnts++;

        while (bad[cnts]){
            cnts++;
        }

        draw(cnts, blacks[0]);
        good.insert(cnts);
        cout << endl;

        ll ii, jj;
        cin >> ii >> jj;
        made_bad(to_id(ii, jj));
    }

    for (int i = 1; i < b; i++){
        for (int cnts = 0; cnts < (1 << (b - i)); cnts++){
            auto p1 = good.begin();
            ll id1 = *p1;
            p1++;
            ll id2 = *p1;
            cout << "? ";
            draw(id1, blacks[i]);
            cout << " ";
            draw(id2, blacks[i]);
            cout << endl;

            ll ii, jj;
            cin >> ii >> jj;

```

```

        made_bad(to_id(ii, jj));
    }
}

ll id_win = *(good.begin());

cout << "! ";
auto tt = to_cord(id_win);
cout << tt.first << " " << tt.second << endl;
}
*/

void solve(){
    int n, m, k;
    cin >> n >> m >> k;
    vector<pair<ll, ll>> arrs;

    for (int i = 0; i < k; i++){
        ll a, b;
        cin >> a >> b;
        arrs.push_back(mp(a - 1, b - 1));
    }

    if (k == 1){
        ll f1x = -10000000 + arrs[0].first, f1y = -10000000 + arrs[0].second;
        ll f2x = 10000000 + arrs[0].first, f2y = 10000000 + arrs[0].second;
        cout << "?_ " << f1x << "_ " << f1y << "_ " << f2x << "_ " << f2y;
        cout << endl;

        ll xx, yy;
        cin >> xx >> yy;

        if (10000000 - 50 <= xx && xx <= 10000000 + 1000 && 10000000 - 50 <= yy && yy <= 10000000
            + 1000){
            cout << "!_" << -10000000 << "_ " << -10000000;
            cout << endl;
            return;
        }
        else{
            cout << "!_" << 10000000 << "_ " << 10000000;
            cout << endl;
            return;
        }
    }

}

int main(){
    std::ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);

    //files();
    solve();
}

```

Task F ()