

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	0	100	100	45	0	345

Task A ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>

#define int long long
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()

using namespace std;

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

signed main(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
}
```

Task B ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
#include <queue>

// #define int long long
#define double long double
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define pb push_back

const int INF = 2e9;

using namespace std;

double getDis(pair<double, double> one, pair<double, double> two){
    return abs(one.first - two.first) * abs(one.first - two.first) + abs(one.second - two.
        second) * abs(one.second - two.second);
}

double get(pair<double, double> one, pair<double, double> two){
    return one.first * two.second - one.second * two.first;
}

double get1(pair<double, double> one, pair<double, double> two){
    return one.first * two.first + two.second * one.second;
}

void solve(){
    int n;
    cin >> n;
    vector<pair<double, double>> a(n);
    for (int i = 0; i < n; i++){
        cin >> a[i].first >> a[i].second;
    }

    if (n == 6){
        double maxD = 0;
        int id = 0;
        for (int i = 0; i < n; i++){
            for (int j = i + 1; j < n; j++){
                pair<int, int> p = { a[j].first - a[i].first, a[j].second - a[i].
                    second };
                int left = 0, right = 0;
                for (int c = 0; c < n; c++){
                    if (c == i || c == j) continue;
                    if (get(p, { a[c].first - a[i].first, a[c].second - a[i].
                        second }) > 0) left++;
                    else right++;
                }
                if (left == right) id = i;
            }
        }
        id = 0;
        for (int i = 0; i < 3; i++, id = (id - 1 + n) % n){
            cout << a[id].first << " " << a[id].second << '\n';
        }
    }
    else{
        double oneX = a[1].first - a[0].first;
        double oneY = a[1].second - a[0].second;

        vector<pair<double, double>> res = a;

        pair<double, double> p1 = {a[1].first - a[0].first, a[1].second - a[0].second};
        pair<double, double> p2 = {a[2].first - a[1].first, a[2].second - a[1].second };
        pair<double, double> p3 = {-oneX, oneY};

        if (get(p2, p1) * get(p2, p3) < 0) res.pb({ a[0].first - oneX, a[0].second + oneY
            });
    }
}
```

```

        else res.pb({ a[0].first + oneX, a[0].second - oneY });

        double twoX = a[2].first - a[1].first;
        double twoY = a[2].second - a[1].second;

        res.pb({ res.back().first + twoX, res.back().second + twoY });
        res.pb({ res.back().first + oneX, res.back().second + oneY });

        swap(res[3], res[5]);
        for (auto& p : res) cout << p.first << "␣" << p.second << "\n";
    }
}

signed main(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    cout.precision(12);
    cout << fixed;
    solve();
}

```

Task C ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>

#define int long long
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()

using namespace std;

void solve(){
    string t;
    cin >> t;
    int n;
    cin >> n;
    int ans = 0;
    for (int i = 0; i < n; i++){
        string s;
        cin >> s;
        int res = 0;

        vector<int> last(t.size());
        vector<int> now(t.size());
        for (int j = 0; j < s.size(); j++){
            int mx = 0;
            for (int c = 0; c < t.size(); c++){
                if (t[c] != s[j]){
                    mx = max(mx, last[c]);
                    now[c] = 0;
                    continue;
                }
                if (j && c) now[c] = mx + 1;
                else now[c] = 1;
                mx = max(mx, last[c]);
            }
            res = max(res, *max_element(all(now)));
            swap(now, last);
        }
        ans += (t.size() - res);
    }
    cout << ans;
}

signed main(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
}
```

Task D ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
#include <queue>

// #define int long long
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()

const int INF = 2e9;

using namespace std;

void solve(){
    int n, m;
    cin >> n >> m;
    int startX, startY, endX, endY;
    cin >> startX >> startY >> endX >> endY;
    startX--; startY--; endX--; endY--;

    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].first >> a[i][j].second;
        }
    }

    priority_queue<pair<int, pair<int, int>>> q;
    vector<vector<int>> d(n, vector<int>(m, INF));
    d[startX][startY] = 0;
    q.push({ 0, { startX, startY } });
    int plus = min(n, m) < 300 ? 13 : 5;
    while (!q.empty()){
        auto p = q.top();
        q.pop();
        int x = p.second.first;
        int y = p.second.second;
        if (-p.first > d[x][y]) continue;
        for (int i = max(0, x - plus); i < min(n, x + plus); i++){
            for (int j = max(0, y - plus); j < min(m, y + plus); j++){
                if (i < 0 || j < 0 || i >= n || j >= m) continue;
                int one = abs(x + a[x][y].first - i);
                int two = abs(y + a[x][y].second - j);
                if (d[i][j] > one + two + d[x][y]){
                    d[i][j] = one + two + d[x][y];
                    q.push({ -(one + two + d[x][y]), { i, j } });
                }
            }
        }
    }

    int ans = d[endX][endY];
    for (int i = 0; i < n; i++){
        for (int j = 0; j < m; j++){
            int one = abs(-endX + a[i][j].first + i);
            int two = abs(-endY + a[i][j].second + j);
            ans = min(ans, one + two + d[i][j]);
        }
    }
    cout << ans;
}

signed main(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    solve();
}
```

Task E ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
#include <queue>
#include <set>
#include <map>
#include <assert.h>

#define int long long
#define double long double
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()
#define pb push_back

const int INF = 2e9;

using namespace std;

void solve(){
    int n, m, b;
    cin >> n >> m >> b;
    vector<pair<int, int>> a(b);
    for (int i = 0; i < b; i++){
        cin >> a[i].first >> a[i].second;
        a[i].first--; a[i].second--;
    }

    int x = 0;
    int mm = 1 << (b - 1);

    queue<int> q;
    set<int> used;
    map<int, int> mp;

    for (int i = 0; i < mm; i++, x += 2 * m){
        q.push(x);
        q.push(x + m);
    }

    set<int> good;

    while (!q.empty()){
        int one, two;
        while (true){
            one = q.front();
            q.pop();
            if (!used.count(one)) break;
        }
        while (true){
            two = q.front();
            q.pop();
            if (!used.count(two)) break;
        }
        assert(!used.count(one) && !used.count(two));
        int d = mp[one];
        cout << "?_ " << a[d].first << "_ " << one + a[d].second << "_ ";
        d = mp[two];
        cout << a[d].first << "_ " << two + a[d].second << endl;
        mp[one]++; mp[two]++;

        if (mp[one] == b) good.insert(one);
        else q.push(one);
        if (mp[two] == b) good.insert(two);
        else q.push(two);

        pair<int, int> p;
        cin >> p.first >> p.second;
        x = p.second / m * m;
        used.insert(x);
        good.erase(x);
    }
}
```

```

        if (!good.empty()){
            int x = *good.begin();
            cout << "!_0_" << x << endl;
            return;
        }
    }
}

signed main(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    cout.precision(12);
    cout << fixed;
    solve();
}

```

Task F ()