

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	6	0	346

Task A ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <set>
#include <map>
#include <cmath>
#include <string>
#include <iomanip>

using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int n;
    cin >> n;
    cout << n - 1;
    return 0;
}
```

Task B ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <set>
#include <map>
#include <cmath>
#include <string>
#include <iomanip>

using namespace std;

const double ANGLE = atan2(1, 1) * 4 / 3;

struct point {
    double x, y;
    point() {}
    point(double x, double y) : x(x), y(y) {};
};

point operator+(point a, point b) {
    return point(a.x + b.x, a.y + b.y);
}

point operator-(point a, point b) {
    return point(a.x - b.x, a.y - b.y);
}

point rotate(point p, double a) {
    double x = p.x * cos(a) - p.y * sin(a);
    double y = p.y * cos(a) + p.x * sin(a);
    return point(x, y);
}

bool operator!=(point a, point b) {
    return a.x != b.x || a.y != b.y;
}

bool operator==(point a, point b) {
    return a.x == b.x && a.y == b.y;
}

double dist(point a, point b) {
    return sqrt((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y));
}

void solve1(int n) {
    vector<point> v(n);
    for (int i = 0; i < n; i++)
        cin >> v[i].x >> v[i].y;
    point p1 = v[0];
    double mx_d = 0;
    int ind = 0;
    for (int i = 1; i < 6; i++)
        if (dist(v[0], v[i]) > mx_d)
            mx_d = dist(v[0], v[i]), ind = i;
    cout << v[0].x << " " << v[0].y << "\n" << v[ind].x << " " << v[ind].y << "\n";
    cout << (v[0].x + v[ind].x) / 2 << " " << (v[0].y + v[ind].y) / 2;
}

void solve2() {
    point a, b, c;
    cin >> a.x >> a.y >> b.x >> b.y >> c.x >> c.y;
    cout << a.x << " " << a.y << "\n";
    a = a - c;
    int q = 2;
    while (q--) {
        point p = rotate(a, ANGLE);
        point p1 = p + c;
        cout << p1.x << " " << p1.y << "\n";
        a = p;
    }
}
```

```

q = 2;
cout << b.x << "␣" << b.y << "\n";
b = b - c;
while (q--) {
    point p = rotate(b, ANGLE);
    point p1 = p + c;
    cout << p1.x << "␣" << p1.y << "\n";
    b = p;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int n;
    cin >> n;
    cout << fixed << setprecision(10);
    if (n == 6)
        solve1(6);
    else
        solve2();
    return 0;
}

```

Task C ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <set>
#include <map>
#include <cmath>
#include <string>
#include <iomanip>

using namespace std;

// #pragma warning (disable : 4996)

int fnd(string &s, int st_pos, char c) {
    for (int i = st_pos; i < s.length(); i++)
        if (s[i] == c)
            return i;
    return -1;
}

int solve(string t, string x) {
    int mx_l = 0, st_pos = 0;
    int n = t.length();
    for (int i = 0; i < x.length(); i++) {
        if (fnd(t, st_pos, x[i]) != -1) {
            int j = i;
            int tmp_len = 0;
            while (j < x.length()) {
                st_pos = fnd(t, st_pos, x[j]);
                if (st_pos == -1)
                    break;
                st_pos++;
                j++;
                tmp_len++;
            }
            st_pos = 0;
            mx_l = max(mx_l, tmp_len);
        }
    }
    return n - mx_l;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    // freopen("input.txt", "r", stdin);
    string t;
    cin >> t;
    int n;
    cin >> n;
    int ans = 0;
    for (int i = 0; i < n; i++) {
        string x;
        cin >> x;
        ans += solve(t, x);
    }
    cout << ans;
    return 0;
}
```

Task D ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <set>
#include <map>
#include <cmath>
#include <string>
#include <iomanip>
#include <climits>

#pragma warning (disable : 4996)

using namespace std;

vector<pair<int, int>> go(9);

int dif(pair<int, int> a, pair<int, int> b) {
    return abs(a.first - b.first) + abs(a.second - b.second);
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    //freopen("input.txt", "r", stdin);
    go[0] = { -1, -1 };
    go[1] = { -1, 0 };
    go[2] = { -1, 1 };
    go[3] = { 0, -1 };
    go[4] = { 0, 0 };
    go[5] = { 0, 1 };
    go[6] = { 1, -1 };
    go[7] = { 1, 0 };
    go[8] = { 1, 1 };
    int n, m;
    cin >> n >> m;
    int stx, sty, enx, eny;
    cin >> stx >> sty >> enx >> eny;
    stx--;
    sty--;
    enx--;
    eny--;
    vector<vector<pair<int, int>>> gr(n, vector<pair<int, int>>(m));
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> gr[i][j].first >> gr[i][j].second;
    vector<vector<int>> d(n, vector<int>(m, INT_MAX));
    d[stx][sty] = 0;
    set<pair<int, pair<int, int>>> s;
    s.insert({ 0, { stx, sty } });
    while (s.size()) {
        auto e = *s.begin();
        s.erase(s.begin());
        int w = e.first;
        pair<int, int> v = e.second;
        int vx = v.first, vy = v.second;
        for (int x = 0; x < n; x++) {
            for (int y = 0; y < m; y++) {
                int dist = abs(vx + gr[vx][vy].first - x) + abs(vy + gr[vx][vy].second - y);
                if (dist + w < d[x][y]) {
                    s.erase({ d[x][y], {x, y} });
                    d[x][y] = dist + w;
                    s.insert({ d[x][y], {x, y} });
                }
            }
        }
    }
    cout << d[enx][eny];
    return 0;
}
```

Task E ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <set>
#include <map>
#include <cmath>
#include <string>
#include <iomanip>
#include <climits>

#pragma warning (disable : 4996)

using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    //freopen("input.txt", "r", stdin);
    int n, m, b;
    cin >> n >> m >> b;
    vector<pair<int, int>> v(b);
    set<pair<int, int>> p1, p2;
    for (int i = 0; i < b; i++) {
        cin >> v[i].first >> v[i].second;
        p1.insert(v[i]);
        p2.insert({ v[i].first + n, v[i].second + m });
    }
    int q;
    q = 8119;
    while (q--) {
        if (p1.size() == 0 || p2.size() == 0) {
            cout << "!_";
            if (p1.size() == 0)
                cout << 1 << "_ " << 1 << endl;
            else
                cout << n + 1 << "_ " << m + 1 << endl;
            return 0;
        }
        cout << "?_";
        pair<int, int> t1 = *p1.begin(), t2 = *p2.begin();
        p1.erase(p1.begin());
        p2.erase(p2.begin());
        cout << t1.first << "_ " << t1.second << "_ " << t2.first << "_ " << t2.second <<
            endl;
        int x, y;
        cin >> x >> y;
        if (x > n)
            p2.insert({ x, y });
        else
            p1.insert({ x, y });
    }
    return 0;
}
```

Task F ()