# Олимпиада СПбГУ по информатике 2019/20 учебного года

| A | B | C | D | E | F | Sum |
|---|---|---|---|---|---|---|
| 100 | 100 | 100 | 40 | 45 | 0 | 385 |

## Task A ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

#define size(x) (int)(x).size()
#define all(x) (x).begin(),(x).end()
#define rall(x) (x).rbegin(),(x).rend()

void bye() {
    cout << "\n";
    exit(0);
}

struct Solution {

    void solve() {
        int n;
        cin >> n;
        int a = 0, b = 0;
        for (int i = 0; i < 1e6; i++) {
            int l = 0, r = n + 1;
            if (a < b) {
                swap(a, b);
            }
            while (r - l > 1) {
                int mid = (l + r) / 2;
                if (a + (n - mid) < b + mid) {
                    r = mid;
                } else {
                    l = mid;
                }
            }
            a += (n - l);
            b += l;
            a = min(a, b);
            b = 0;
        }
        cout << a;
    }
};

signed main() {
    #ifdef _TIM
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w",  stdout);
    #endif // _TIM
    int T = 1;
//    cin >> T;
    while (T--) {
        Solution kek;
        kek.solve();
    }
    bye();
}
```

## Task B ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

#define size(x) (int)(x).size()
#define all(x) (x).begin(),(x).end()
#define rall(x) (x).rbegin(),(x).rend()

const int INF = 1e9 + 1;
const double EPS = 1e-2;
const double PI = acos(-1);

void bye() {
    cout << "\n";
    exit(0);
}

struct Coord {
    double x, y;
    Coord() {}
    Coord(double _x, double _y) {
        x = _x, y = _y;
    }
};

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

struct Solution {

    bool eq(Coord a, Coord b) {
        return fabs(a.x - b.x) + fabs(a.y - b.y) < EPS;
    }

    double dist(Coord a, Coord b) {
        return hypot(a.x - b.x, a.y - b.y);
    }

    Coord rotateLeft(Coord v) {
        double alpha = 60.0 * PI / 180.0;
        return {v.x * cos(alpha) - v.y * sin(alpha),
                v.y * cos(alpha) + v.x * sin(alpha)};
    }

    void encrypt() {
        int n = 6;
        vector<Coord> g(n);
        for (int i = 0; i < n; i++) {
            cin >> g[i].x >> g[i].y;
        }
        sort(all(g));
        vector<Coord> res(3);
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                for (int k = 0; k < n; k++) {
                    if (i == j || j == k || i == k) {
                        continue;
                    }
                    Coord kek = {g[j].x - g[i].x, g[j].y - g[i].y};
                    kek = rotateLeft(kek);
                    if (dist(Coord(g[j].x + kek.x, g[j].y + kek.y), g[k]) < EPS) {
                        res[0] = g[i];
                        res[1] = g[j];
                        res[2] = g[k];
                    }
                }
            }
        }
        for (auto v : res) {
            cout << v.x << "␣" << v.y << "\n";
        }
    }

    void decrypt() {
        int n = 3;
        vector<Coord> g(6);
        for (int i = 0; i < n; i++) {
            cin >> g[i].x >> g[i].y;
        }
        Coord kek = {g[1].x - g[0].x, g[1].y - g[0].y};
        kek = rotateLeft(kek);
        for (int i = 3; i < 6; i++) {
            kek = rotateLeft(kek);
            g[i] = {g[i - 1].x + kek.x, g[i - 1].y + kek.y};
        }
        for (auto v : g) {
            cout << v.x << "␣" << v.y << "\n";
        }
    }

    void solve() {
        cout << fixed << setprecision(3);
        int n;
        cin >> n;
        if (n == 3) {
            decrypt();
        } else {
            encrypt();
        }
    }
};

signed main() {
    #ifdef _TIM
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w",  stdout);
    #endif // _TIM
    int T = 1;
//    cin >> T;
    while (T--) {
        Solution kek;
        kek.solve();
    }
    bye();
}
```

## Task C ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

#define size(x) (int)(x).size()
#define all(x) (x).begin(),(x).end()
#define rall(x) (x).rbegin(),(x).rend()

void bye() {
    cout << "\n";
    exit(0);
}

struct Solution {

    void solve() {
        string t;
        cin >> t;
        int n;
        cin >> n;
        vector<string> g(n);
        for (int i = 0; i < n; i++) {
            cin >> g[i];
        }
        int ans = 0;
        for (int i = 0; i < n; i++) {
            int now = size(t);
            for (int j = 0; j < size(g[i]); j++) {
                int r = j;
                int l = 0;
                int kekes = 0;
                while (l < size(t)) {
                    if (r < size(g[i]) && g[i][r] == t[l]) {
                        r++, l++;
                    } else {
                        l++, kekes++;
                    }
                }
                now = min(kekes, now);
            }
//            cout << now << "\n";
            ans += now;
        }
        cout << ans;
    }
};

signed main() {
    #ifdef _TIM
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w",  stdout);
    #endif // _TIM
    int T = 1;
//    cin >> T;
    while (T--) {
        Solution kek;
        kek.solve();
    }
    bye();
}
```

# Task D ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

#define size(x) (int)(x).size()
#define all(x) (x).begin(),(x).end()
#define rall(x) (x).rbegin(),(x).rend()

const int INF = 1e9 + 1;

void bye() {
    cout << "\n";
    exit(0);
}

struct Coord {
    int x, y;
};

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

struct Solution {

    int n, m;
    vector<vector<Coord> > g;
    vector<vector<int> > d;
    Coord s, t;

    bool check(Coord v) {
        return v.x < n && v.y < m && v.x >= 0 && v.y >= 0;
    }

    void solve() {
        cin >> n >> m;
        cin >> s.x >> s.y >> t.x >> t.y;
        --s.x, --s.y, --t.x, --t.y;
        g.resize(n, vector<Coord>(m));
        d.resize(n, vector<int>(m, INF));
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                cin >> g[i][j].x >> g[i][j].y;
            }
        }
        set<pair<int, Coord> > q;
        d[s.x][s.y] = 0;
        q.insert({0, s});
        while (!q.empty()) {
            auto v = (q.begin() -> second);
            q.erase(q.begin());
            for (int tx = -100; tx <= 100; tx++) {
                for (int ty = -100; ty <= 100; ty++) {
                    Coord to = {v.x + g[v.x][v.y].x + tx, v.y + g[v.x][v.y].y + ty};
                    if (check(to) && d[to.x][to.y] > d[v.x][v.y] + abs(tx) + abs(ty)) {
                        q.erase({d[to.x][to.y], to});
                        d[to.x][to.y] = d[v.x][v.y] + abs(tx) + abs(ty);
                        q.insert({d[to.x][to.y], to});
                    }
                }
            }
        }
        cout << d[t.x][t.y];
    }
};

signed main() {
    #ifdef _TIM
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w",  stdout);
    #endif // _TIM
    int T = 1;
//    cin >> T;
    while (T--) {
        Solution kek;
        kek.solve();
    }
    bye();
}
```

# Task E ()

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

using namespace std;

typedef long long ll;
typedef long double ld;

#define size(x) (ll)(x).size()
#define all(x) (x).begin(),(x).end()
#define rall(x) (x).rbegin(),(x).rend()

const ll INF = 1e9 + 1;
const double EPS = 1e-2;
const double PI = acos(-1);

void bye() {
    cout << "\n";
    exit(0);
}

struct Coord {
    ll x, y;
};

struct Solution {

    ll n, m, B;

    Coord ask(Coord a, Coord b) {
        a.x++, a.y++, b.x++, b.y++;
        cout << "?␣" << a.x << "␣" << a.y << "␣" << b.x << "␣" << b.y << endl;
        Coord v;
        cin >> v.x >> v.y;
        v.x--, v.y--;
        return v;
    }

    void solve() {
        cin >> n >> m >> B;
        vector<Coord> ord(B);
        for (ll i = 0; i < B; i++) {
            cin >> ord[i].x >> ord[i].y;
            ord[i].x--, ord[i].y--;
        }
        set<ll> alive;
        int r = 0;
        for (ll i = 0; i < (1 << B); i += 2) {
            alive.insert(i);
            alive.insert(i + 1);
            Coord del = ask({i * n + ord[0].x, ord[0].y}, {(i + 1) * n + ord[0].x, ord[0].y});
            alive.erase(del.x / n);
        }
        for (ll q = 1; q < B; q++) {
            set<ll> now;
            int l = 0;
            while (size(alive) >= 2) {
                ll f = (*alive.begin());
                alive.erase(alive.begin());
                ll s = (*alive.begin());
                alive.erase(alive.begin());
                Coord cf = {f * n + ord[q].x, ord[q].y};
                Coord sf = {s * n + ord[q].x, ord[q].y};
                now.insert(s);
                now.insert(f);
                Coord del = ask(cf, sf);
                now.erase(del.x / n);
                alive.erase(del.x / n);
            }
            swap(alive, now);
        }
        cout << "!␣" << (*alive.begin()) * n + 1 << "␣" << 1 << endl;
    }
};

signed main() {
    #ifdef _TIM
//        freopen("input.txt", "r", stdin);
//        freopen("output.txt", "w",  stdout);
    #endif // _TIM
    ll T = 1;
//    cin >> T;
    while (T--) {
        Solution kek;
        kek.solve();
    }
    bye();
}
```

**Task F ()**