

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	100	18	458

Task A ()

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

using namespace std;
#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
#define endl "\n"

// #define MULTITEST

#define double long double
#define int long long

#ifdef LOCAL
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

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

int32_t main() {
#ifdef LOCAL
    fin.open("in.txt");
#endif
    ON_START
#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; i++) {
#endif
        solve();
#ifdef MULTITEST
    }
#endif
    return 0;
}
```

Task B ()

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

using namespace std;
#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
#define endl "\n"

// #define MULTITEST

#define double long double
#define int long long

#ifdef LOCAL
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

struct point_t {
    double x, y;

    point_t sym(point_t center) {
        point_t rep = {x - center.x, y - center.y};
        rep = {-rep.x, -rep.y};
        rep = {rep.x + center.x, rep.y + center.y};
        return rep;
    }
};

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

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

point_t operator/(point_t a, double b) {
    return {a.x / b, a.y / b};
}

bool half(point_t a) {
    return a.y < 0 || (a.y == 0 && a.x < 0);
}

double cross(point_t a, point_t b) {
    return a.x * b.y - a.y * b.x;
}

void solve() {
    int k;
    cin >> k;
    // TODO: random order
    cout.precision(100);
    if (k == 6) {
        point_t pts[6];
        for (auto &pt : pts) {
            cin >> pt.x >> pt.y;
        }
        point_t f = (pts[0] + pts[1]) / 2;
        for (auto &pt : pts) {
            pt.x -= f.x;
            pt.y -= f.y;
        }

        sort(pts, pts + 6, [](point_t a, point_t b) {
            bool ha = half(a);
            bool hb = half(b);
            if (ha != hb) {
                return ha < hb;
            }
        });

        return cross(a, b) > 0;
    }
}
```

```

    });

    for (int i = 0; i < 3; i++) {
        cout << pts[i].x + f.x << "┘" << pts[i].y + f.y << endl;
    }
    return;
}

point_t res[6];
for (int i = 0; i < 3; i++) {
    cin >> res[i].x >> res[i].y;
}
for (int i = 0; i < 3; i++) {
    res[i + 3] = res[i].sym(res[0] + (res[2] - res[1]));
}

for (auto p : res) {
    cout << p.x << "┘" << p.y << endl;
}
}

int32_t main() {
#ifdef LOCAL
    fin.open("in.txt");
#endif
    ON_START
#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; i++) {
#endif
        solve();
#ifdef MULTITEST
    }
#endif
    return 0;
}

```

Task C ()

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

using namespace std;
#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
#define endl "\n"

// #define MULTITEST

#define double long double
#define int long long

#ifdef LOCAL
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

string t;
int calc(const string &s) {
    int n = s.length();
    int res = t.length();
    for (int start = 0; start < n; start++) {
        int cres = 0;

        int i = start;
        for (char ch : t) {
            if (i < n && s[i] == ch) {
                i++;
            } else {
                cres++;
            }
        }

        if (res > cres) {
            res = cres;
        }
    }
    return res;
}

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

int32_t main() {
#ifdef LOCAL
    fin.open("in.txt");
#endif
    ON_START
#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; i++) {
#endif
        solve();
#ifdef MULTITEST
    }
#endif
    return 0;
}
```

Task D ()

```
#ifndef LOCAL
#pragma GCC optimize("Ofast")
#endif
#include <bits/stdc++.h>

using namespace std;
#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
#define endl "\n"

// #define MULTITEST

#define double long double
// #define int long long

#ifdef int
#define IMAX INT64_MAX
#else
#define IMAX INT32_MAX
#endif

#ifdef LOCAL
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

struct point_t {
    int x, y;
};

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

bool operator!=(point_t a, point_t b) {
    return !(a == b);
}

struct cmp_t {
    bool operator()(pair<point_t, int> a, pair<point_t, int> b) {
        if (a.second != b.second) {
            return a.second < b.second;
        }
        if (a.first.x != b.first.x) {
            return a.first.x < b.first.x;
        }
        return a.first.y < b.first.y;
    }
};

void solve() {
    int n, m;
    cin >> n >> m;
    point_t s{}, f{};
    cin >> s.x >> s.y >> f.x >> f.y;
    s.x--;
    s.y--;
    f.x--;
    f.y--;
    point_t mp[n][m];
    int dist[n][m];
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> mp[i][j].x >> mp[i][j].y;
            dist[i][j] = IMAX;
        }
    }

    cmp_t cmp;
    priority_queue<pair<point_t, int>, vector<pair<point_t, int>>, cmp_t> q(cmp);
    q.emplace(s, 0);
    dist[s.x][s.y] = 0;
}
```

```

    if (n == 1 || m == 1) {
        while (!q.empty()) {
            auto top = q.top();
            q.pop();
            auto v = top.first;
            if (top.second > dist[v.x][v.y]) {
                continue;
            }
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < m; j++) {
                    int d = abs(i - (v.x + mp[v.x][v.y].x)) + abs(j - (v.y + mp[v.x][v.y].y)) +
                        top.second;
                    if (dist[i][j] > d) {
                        dist[i][j] = d;
                        q.emplace(point_t{i, j}, d);
                    }
                }
            }
        }
    } else {
        while (!q.empty()) {
            auto top = q.top();
            q.pop();
            auto v = top.first;
            if (top.second > dist[v.x][v.y]) {
                continue;
            }
            for (int i = v.x - 1; i <= v.x + 1; i++) {
                if (i < 0) {
                    continue;
                }
                if (i >= n) {
                    break;
                }
                for (int j = v.y - 1; j <= v.y + 1; j++) {
                    if (j < 0) {
                        continue;
                    }
                    if (j >= m) {
                        break;
                    }
                    int d = abs(i - (v.x + mp[v.x][v.y].x)) + abs(j - (v.y + mp[v.x][v.y].y)) +
                        top.second;
                    if (v != s) {
                        d = min(d, abs(i - v.x) + abs(j - v.y) + top.second);
                    }
                    if (dist[i][j] > d) {
                        dist[i][j] = d;
                        q.emplace(point_t{i, j}, d);
                    }
                }
            }
        }
    }
}

cout << dist[f.x][f.y] << endl;
}

int32_t main() {
#ifdef LOCAL
    fin.open("in.txt");
#endif
    ON_START
#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; i++) {
#endif
        solve();
#ifdef MULTITEST
    }
#endif
    return 0;
}

```


Task E ()

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

using namespace std;
#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
#define endl "\n"

// #define MULTITEST

#define double long double
#define int long long

#ifdef LOCAL
// #define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

void solve() {
    int tp[20];
    tp[0] = 1;
    for (int i = 1; i < 20; i++) {
        tp[i] = tp[i - 1] * 2;
    }

    int n, m, b;
    cin >> n >> m >> b;
    pair<int, int> pos[b];
    for (int i = 0; i < b; i++) {
        cin >> pos[i].first >> pos[i].second;
        pos[i].first--;
        pos[i].second--;
    }
    bool damaged[20000];
    memset(damaged, false, sizeof(damaged));
    for (int i = 0; i < b; i++) {
        int last = 0;
        for (int asd = 0; asd < tp[b - i - 1]; asd++) {
            while (damaged[last]) {
                last++;
            }
            int l1 = last;
            last++;
            while (damaged[last]) {
                last++;
            }
            (cout << "?_ " << l1 * n + pos[i].first << "_ " << pos[i].second << "_ " << last * n +
             pos[i].first << "_ " << pos[i].second << endl).flush();
            last++;
            int x, y;
            cin >> x >> y;
            damaged[x / n] = true;
        }
    }

    for (int i = 0;; i++) {
        if (!damaged[i]) {
            (cout << "!_ " << i * n << "_0\n").flush();
            break;
        }
    }
}

int32_t main() {
#ifdef LOCAL
    fin.open("in.txt");
#endif
    ON_START
#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; i++) {
#endif
        solve();
    }
```

```
#ifdef MULTITEST
}
#endif
return 0;
}
```

[illegible]


```

for (int i = 0; i < n; i++) {
    p[i] = 0;
}

vector<int> _gr[n];

gr = _gr;

set<vector<pair<int, int>>> st;

int num[n];
for (int i = 0; i < n; i++) {
    num[i] = i;
}

while (p[0] == 0) {
    do {
        vector<pair<int, int>> t;
        for (int i = 1; i < n; i++) {
            int a = num[p[i]];

            int b = num[i];

            t.emplace_back(min(a, b), max(a, b));
        }

        sort(t.begin(), t.end());

        st.insert(t);
    } while (next_permutation(num, num + n));

    for (int i = n - 1; i >= 0; i--) {
        p[i]++;

        if (p[i] == i) {
            p[i] = 0;
        } else {
            break;
        }
    }
}

int res[m];
for (int i = 0; i < m; i++) {
    res[i] = 0;
}

for (const vector<pair<int, int>> &v : st) {
    for (int i = 0; i < n; i++) {
        gr[i].clear();
    }

    for (auto pa : v) {
        gr[pa.first].push_back(pa.second);
        gr[pa.second].push_back(pa.first);
    }

    int dist = 0;
    for (int i = 0; i < n; i++) {
        int d[n];

        for (int j = 0; j < n; j++) {
            d[j] = IMAX;
        }
    }
}

```

```

d[i] = 0;
deque<int> q;
q.push_back(i);

while (!q.empty()) {
    int f = q.front();
    q.pop_front();

    for (int g : gr[f]) {
        if (d[g] == IMAX) {
            d[g] = d[f] + 1;
            q.push_back(g);
        }
    }
}

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

res[dist / 2 - 1]++;
}

for (int i = 0; i < m; i++) {
    cout << res[i] << (i == m - 1 ? endl : "\n");
}
}

int32_t main() {
#ifdef LOCAL
    fin.open("in.txt");
#endif
    ON_START
#ifdef MULTITEST
    int t;

    cin >> t;
    for (int i = 0; i < t; i++) {
#ifdef MULTITEST
    }
#endif
        return 0;
    }
}

```