

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;

signed main() {
    int n;
    cin >> n;
    cout << n - 1 << '\n';
}
```

Task B ()

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

using namespace std;

bool eq(double a, double b) {
    return abs(a - b) < 0.0001;
}

double dist(pair<double, double> a, pair<double, double> b) {
    double x = a.first - b.first;
    double y = a.second - b.second;
    return sqrt(x * x + y * y);
}

pair<double, double> inv(pair<double, double> a, pair<double, double> mid) {
    pair<double, double> dif;
    dif.first = mid.first - a.first;
    dif.second = mid.second - a.second;
    pair<double, double> ans;
    ans.first = 2 * dif.first + a.first;
    ans.second = 2 * dif.second + a.second;
    return ans;
}

signed main() {
    cout.precision(20);
    int n;
    cin >> n;
    if (n == 6) {
        vector<pair<double, double>> six(n);
        for (int i = 0; i < n; ++i) {
            cin >> six[i].first >> six[i].second;
        }
        int fst = 0;
        for (int i = 0; i < n; ++i) {
            if (six[i] < six[fst])
                fst = i;
        }
        int scnd = -1;
        for (int i = 0; i < n; ++i) {
            if (i != fst) {
                if (scnd == -1 || dist(six[fst], six[i]) < dist(six[fst], six[scnd])) {
                    scnd = i;
                }
            }
        }
        int third = -1;
        for (int i = 0; i < n; ++i) {
            if (i != fst && i != scnd) {
                if (third == -1 || dist(six[fst], six[i]) < dist(six[fst], six[third])) {
                    third = i;
                }
            }
        }
        cout << six[scnd].first << '\u' << six[scnd].second << '\n';
        cout << six[fst].first << '\u' << six[fst].second << '\n';
        cout << six[third].first << '\u' << six[third].second << '\n';
    }
    if (n == 3) {
        pair<double, double> fst, scnd, third;
        vector<pair<double, double>> ans(6);
        pair<double, double> mid;
        cin >> scnd.first >> scnd.second;
        cin >> fst.first >> fst.second;
        cin >> third.first >> third.second;
        mid.first = scnd.first + third.first - fst.first;
        mid.second = scnd.second + third.second - fst.second;
        ans[0] = scnd;
        ans[3] = inv(scnd, mid);
        ans[1] = fst;
        ans[4] = inv(fst, mid);
        ans[2] = third;
```

```

    ans[5] = inv(third, mid);
    for (int i = 0; i < 6; ++i) {
        cout << ans[i].first << ' ' << ans[i].second << '\n';
    }
}

```

Task C ()

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

using namespace std;

signed main() {
    string s;
    cin >> s;
    int n;
    cin >> n;
    vector<string> ss(n);
    for (auto& i : ss) {
        cin >> i;
    }
    vector<char> st1;
    for (int i = 0; i < s.size(); ++i) {
        st1.push_back(s[s.size() - i - 1]);
    }
    int ans = 0;
    for (int i = 0; i < n; ++i) {
        int res = 0;
        for (int j = 0; j < ss[i].size(); ++j) {
            vector<char> st = st1;
            int cnt = 0;
            while (!st.empty() && (j + cnt < ss[i].size())) {
                if (st.back() == ss[i][j + cnt]) {
                    ++cnt;
                }
                st.pop_back();
            }
            res = max(cnt, res);
        }
        ans += s.size() - res;
    }
    cout << ans << '\n';
}
```

Task D ()

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

using namespace std;

const int N = 1010;
vector<pair<int, int>> g[N * N];
int d[N * N];
pair<int, int> mtr[N][N];

void dfs(int s) {
    set<pair<int, int>> q;
    q.insert(make_pair(1, s));
    d[s] = 1;
    while (!q.empty()) {
        pair<int, int> v = *q.begin();
        q.erase(q.begin());
        //cout << v << endl;
        for (auto i : g[v.second]) {
            if (d[i.first] == 0 || d[i.first] > d[v.second] + i.second) {
                if (d[i.first] > d[v.second] + i.second)
                    q.erase(q.find(make_pair(d[i.first], i.first)));
                d[i.first] = d[v.second] + i.second;
                q.insert(make_pair(d[i.first], i.first));
            }
        }
    }
}

signed main() {
    int n, m;
    cin >> n >> m;
    int ar, ac, br, bc;
    cin >> ar >> ac >> br >> bc;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            cin >> mtr[i][j].first >> mtr[i][j].second;
        }
    }
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            for (int ni = 0; ni < n; ++ni) {
                for (int nj = 0; nj < m; ++nj) {
                    int ind = i * N + j;
                    int jnd = ni * N + nj;
                    g[ind].push_back(make_pair(jnd, abs(i + mtr[i][j].first - ni) + abs(j + mtr[i][j].second - nj)));
                }
            }
        }
    }
    dfs((ar - 1) * N + ac - 1);
    cout << d[(br - 1) * N + bc - 1] - 1 << endl;
}
```

Task E ()

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

using namespace std;

int n, m, b;
struct rect {
    int i;
    int cnt = 0;
    rect(int i) : i(i), cnt(0) {}
    pair<int, int> get(pair<int, int> p) {
        return make_pair(i * n + p.first, p.second - m + 1);
    }
};

bool operator < (const rect& a, const rect& b) {
    return a.cnt > b.cnt || (a.cnt == b.cnt && a.i > b.i);
}

int cnt[10000];

rect getR(priority_queue<rect>& q) {
    rect ans = q.top();
    // cout << ans.i << endl;
    q.pop();
    while (cnt[ans.i] == -1) {
        ans = q.top();
        // cout << ans.i << endl;
        q.pop();
    }
    return ans;
}

signed main() {
    cin >> n >> m >> b;
    vector<pair<int, int>> ps(b);
    for (int i = 0; i < b; ++i) {
        cin >> ps[i].first >> ps[i].second;
        ps[i].first -= 1;
        ps[i].second -= 1;
    }
    priority_queue<rect> q;
    for (int i = 0; i < (1 << b); ++i) {
        q.push(rect(i));
    }
    bool run = true;
    while (run) {
        rect fst = getR(q);
        rect scnd = getR(q);
        auto fstxy = fst.get(ps[fst.cnt]);
        auto scndxy = scnd.get(ps[scnd.cnt]);
        cout << '?' << ' ' << fstxy.first << ' ' << fstxy.second << ' ' << scndxy.first << ' ' <<
            scndxy.second << endl;
        int x, y;
        cin >> x >> y;
        fst.cnt += 1;
        scnd.cnt += 1;
        x /= n;
        if (-m < y && y <= 0 && 0 <= x && x < 10000) {
            cnt[x] = -1;
            //cout << x << endl;
        }
        if (cnt[fst.i] != -1) {
            if (fst.cnt == b) {
                auto fstc = fst.get(make_pair(0, 0));
                cout << '!' << ' ' << fstc.first << ' ' << fstc.second << endl;
                return 0;
            } else {
                q.push(fst);
            }
        }
        if (cnt[scnd.i] != -1) {
            if (scnd.cnt == b) {

```

```

        auto scndc = scnd.get(make_pair(0, 0));
        cout << '!' << ' ' << scndc.first << ' ' << scndc.second << endl;
        return 0;
    } else {
        q.push(scnd);
    }
}
}
}

```

Task F ()