

# 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 <bits/stdc++.h>

#define int long long
#define all(x) begin(x), end(x)
#define x first
#define y second

using namespace std;

void fast() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
#ifdef _offline
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif // _offline
}

signed main() {
    fast();
    int n;
    cin >> n;
    cout << n - 1;
    return 0;
}
```

## Task B ()

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

#define int long long
#define all(x) begin(x), end(x)
#define x first
#define y second

using namespace std;

typedef long double ld;

void fast() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    cout << fixed << setprecision(10);
#ifdef _offline
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif // _offline
}

const ld INF = 1e18;
const ld EPS = 1e-2;

struct pt {
    ld x, y;
};

vector<pt> a;
vector<bool> used;
vector<pt> res;
ld d;

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

ld vec(pt a, pt b) {
    return a.x * a.y + b.x * b.y;
}

signed main() {
    fast();
    int n;
    cin >> n;
    a.resize(n);
    used.resize(n);
    for(int i = 0; i < n; i++) {
        cin >> a[i].x >> a[i].y;
    }
    if(n == 6) {
        for(int i = 0; i < 3; i++) {
            cout << a[i].x << " " << a[i].y << '\n';
        }
        ld mn = INF;
        for(int i = 1; i < 6; i++) {
            mn = min(mn, dist(a[0], a[i]));
        }
        for(int i = 1; i < n; i++) {
            if(abs(dist(a[i], a[0]) - mn) <= EPS) {
                res.push_back(a[i]);
                res.push_back(a[0]);
            }
        }
        res.pop_back();
        pt v1 = {res[1].x - res[0].x, res[1].y - res[0].y};
        pt v2 = {res[2].x - res[1].x, res[2].y - res[1].y};
        for(int i = 0; i < 3; i++) {
            cout << res[i].x << "\u" << res[i].y << '\n';
        }
    }
}
```

```

else {
    ld d = sqrt((a[1].x - a[0].x) * (a[1].x - a[0].x) + (a[1].y - a[0].y) * (a[1].y - a[0].y))
    ;
    pt v1 = {-a[2].y + a[0].y, a[2].x - a[0].x};
    ld ln = sqrt(v1.x * v1.x + v1.y * v1.y);
    v1.x /= ln;
    v1.y /= ln;
    ln = sqrt(v1.x * v1.x + v1.y * v1.y);
    // cerr << ln << '\n';
    v1.x *= d;
    v1.y *= d;

    vector<pt> res = {{a[2].x + v1.x, a[2].y + v1.y}, {a[1].x + 2 * v1.x, a[1].y + 2 * v1.y},
        {a[0].x + v1.x, a[0].y + v1.y}};
    for(int i = 0; i < 3; i++) {
        cout << a[i].x << " " << a[i].y << '\n';
    }
    // for(int i = 0; i < 3; i++) {
    //     cerr << res[i].x << " " << res[i].y << '\n';
    // }
    // cerr << dist(res[1], res[0]) << " " << d << '\n';
    if(abs(dist(res[1], res[0]) - d) > EPS) {
    //     cerr << 1;
        reverse(all(a));
        ld d = sqrt((a[1].x - a[0].x) * (a[1].x - a[0].x) + (a[1].y - a[0].y) * (a[1].y - a
            [0].y));
        pt v1 = {-a[2].y + a[0].y, a[2].x - a[0].x};
        ld ln = sqrt(v1.x * v1.x + v1.y * v1.y);
        v1.x /= ln;
        v1.y /= ln;
        ln = sqrt(v1.x * v1.x + v1.y * v1.y);
        // cerr << ln << '\n';
        v1.x *= d;
        v1.y *= d;
        res = {{a[2].x + v1.x, a[2].y + v1.y}, {a[1].x + 2 * v1.x, a[1].y + 2 * v1.y}, {a[0].x
            + v1.x, a[0].y + v1.y}};
        reverse(all(res));
    }
    for(int i = 0; i < 3; i++) {
        cout << res[i].x << " " << res[i].y << '\n';
    }
    // cout << a[2].x + v1.x << " " << a[2].y + v1.y << '\n';
    // cout << a[1].x + 2 * v1.x << " " << a[1].y + 2 * v1.y << '\n';
    // cout << a[0].x + v1.x << " " << a[0].y + v1.y << '\n';
}
return 0;
}

```

## Task C ()

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

#define int long long
#define all(x) begin(x), end(x)
#define x first
#define y second

using namespace std;

typedef long double ld;

void fast() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    cout << fixed << setprecision(10);
    #ifdef _offline
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif // _offline
}

const int INF = 1e18;
const ld EPS = 1e-9;

signed main() {
    fast();
    string s;
    cin >> s;
    int n;
    cin >> n;
    int ans = 0;
    for(int i = 0; i < n; i++) {
        string t;
        cin >> t;
        int mn = 0;
        for(int k = 0; k < t.size(); k++) {
            int cnt = 0, cur = k;
            for(int j = 0; j < s.size() && cur < t.size(); j++) {
                if(s[j] == t[cur]) {
                    cnt++;
                    cur++;
                }
            }
            mn = max(mn, cnt);
        }
        // cerr << mn << '\n';
        ans += s.size() - mn;
    }
    cout << ans;
    return 0;
}
```

## Task D ()

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

#define int long long
#define all(x) begin(x), end(x)
#define x first
#define y second
#define pii pair<int, int>

using namespace std;

typedef long double ld;

void fast() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    cout << fixed << setprecision(10);
#ifdef _offline
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif // _offline
}

const int INF = 1e18;
const ld EPS = 1e-9;

int n, m;

bool cor(int x, int y) {
    return x < n && x > -1 && y < m && y > -1;
}

signed main() {
    fast();
    cin >> n >> m;
    pii s, f;
    cin >> s.x >> s.y >> f.x >> f.y;
    s.x--;
    s.y--;
    f.x--;
    f.y--;
    vector<vector<pii>> a(n, vector<pii>(m));
    vector<vector<pii>> g(n * m);
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            cin >> a[i][j].x >> a[i][j].y;
        }
    }
    vector<int> dx = {-1, -1, -1, 0, 0, 1, 1, 1};
    vector<int> dy = {-1, 0, 1, -1, 1, -1, 0, 1};
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < m; j++) {
            for(int k = 0; k < n; k++) {
                for(int h = 0; h < m; h++) {
                    int tox = k;
                    int toy = h;
                    if(cor(tox, toy)) {
                        int w = 0;
                        tox -= a[i][j].x;
                        toy -= a[i][j].y;
                        w = abs(tox - i) + abs(toy - j);
                        assert(w >= 0);
                        // cerr << w << '\n';
                        g[i * m + j].push_back({k * m + h, w});
                    }
                }
            }
        }
    }
    set<pii> st;
    int str = s.x * m + s.y;
```

```

vector<int> d(n * m, INF);
d[str] = 0;
st.insert({0, str});
while(st.size()) {
    int u = st.begin()->y;
    st.erase(st.begin());
    for(auto v : g[u]) {
        if(d[v.x] > d[u] + v.y) {
            st.erase({d[v.x], v.x});
            d[v.x] = d[u] + v.y;
            st.insert({d[v.x], v.x});
        }
    }
}
// for(int i = 0; i < n; i++) {
//     for(int j = 0; j < m; j++) {
//         cerr << d[i * m + j] << " ";
//     }
//     cerr << '\n';
// }
cout << d[f.x * m + f.y];
return 0;
}

```

## Task E ()

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

#define int long long
#define all(x) begin(x), end(x)
#define x first
#define y second
#define pii pair<int, int>

using namespace std;

typedef long double ld;

void fast() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    cout << fixed << setprecision(10);
#ifdef _offline
    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);
#endif // _offline
}

const int INF = 1e18;
const ld EPS = 1e-9;

signed main() {
    fast();
    int n, m, b;
    cin >> n >> m >> b;
    if(b == 1) {
        int x, y;
        cin >> x >> y;
        cout << "?_ " << x << "_ " << y << "_ " << x + n << "_ " << y << endl;
        int tx, ty;
        cin >> tx >> ty;
        if(pii{x, y} != pii{tx, ty}) {
            cout << "!_1_1" << endl;
        }
        else {
            cout << "!_ " << n + 1 << "_1" << endl;
        }
    }
    return 0;
}
```

## Task F ()