

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	100	18	518

Task A ()

```
#include <bits/stdc++.h>
#define int long long
using namespace std;

struct Solver {
    int n;

    Solver(int n): n(n) {}

    int solve() {
        int a = 0, b = 0;
        int res = 0;

        for(int i = 0; i < (int)(1e7); ++i) {
            int m = (n+a)/2;
            a += m-a;
            b += n-(m-a);

            if(a > b) swap(a, b);
            b = 0;

            res = max(res, a);
        }

        return res;
    }
};

const int local = 0, stress = 0;
int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);

    int tests = 1;

    while(tests--) {
        if(!stress) {
            int n;
            cin >> n;

            Solver solver(n);
            int res = solver.solve();
            cout << res << "\n";
        } else {}
    }

    return 0;
}
```

Task B ()

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
typedef long double ld;

const ld pi = atan2(0, -1);

struct Pnt {
    ld x, y;

    Pnt(ld x, ld y): x(x), y(y) {
    }
};

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

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

Pnt operator/(Pnt a, ld h) {
    return {a.x / h, a.y / h};
}

Pnt rot(Pnt a, ld phi) {
    ld x = a.x*cos(phi) - a.y*sin(phi);
    ld y = a.y*cos(phi) + a.x*sin(phi);

    return {x, y};
}

ld dist(Pnt a, Pnt b) {
    return sqrt(pow(a.x-b.x, 2) + pow(a.y-b.y, 2));
}

struct Encoder {
    vector<Pnt> p;

    Encoder(vector<Pnt> p): p(p) {
    }

    vector<Pnt> encode() {
        vector<Pnt> res;

        ld mx = 0;
        Pnt O(0, 0);
        Pnt v = p[0];

        for(Pnt u : p) {
            if(dist(v, u) > mx) {
                mx = dist(v, u);
                O = (v+u)/2;
            }
        }

        res.push_back(O);
        res.push_back(v);
        res.push_back({0, 0});

        return res;
    }
};

struct Decoder {
    vector<Pnt> p;

    Decoder(vector<Pnt> p): p(p) {
```

```

    }

    vector<Pnt> decode() {
        vector<Pnt> res;

        Pnt v = p[1];
        Pnt O = p[0];
        for(int i = 0; i < 6; ++i) {
            res.push_back(v);
            Pnt u = v-O;
            u = rot(u, pi/3);
            v = O + u;
        }

        return res;
    }
};

const int local = 0, stress = 0;
int32_t main() {
    if(local) freopen("input.txt", "r", stdin);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cout << fixed << setprecision(4);

    int tests = 1;

    while(tests--) {
        if(!stress) {
            int n;
            cin >> n;

            vector<Pnt> pnts(n, {0, 0});
            for(int i = 0; i < n; ++i)
                cin >> pnts[i].x >> pnts[i].y;

            if(n == 6) {
                Encoder encoder(pnts);
                vector<Pnt> code = encoder.encode();
                for(int i = 0; i < code.size(); ++i)
                    cout << code[i].x << " " << code[i].y << "\n";
            }

            else if(n == 3) {
                Decoder decoder(pnts);
                vector<Pnt> pnts = decoder.decode();
                for(auto i : pnts)
                    cout << i.x << " " << i.y << "\n";
            }

            } else {}
        }

    return 0;
}

```

Task C ()

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
typedef long double ld;

struct Solver {
    int n;
    vector<string> a;
    string t;

    Solver(string t, vector<string> a): t(t), a(a) {
        n = a.size();
    }

    int _solve(string s, int l) {
        int r = 0;
        int lo = l;

        int res = t.size();
        while((r < t.size()) && (l < s.size())) {
            if(s[l] == t[r]) {
                res--;
                l++;
                r++;
            }

            else {
                r++;
            }
        }

        return res;
    }

    // cout << "MDA: " << s << " " << lo << " " << " | " << res << "\n";
    return res;
}

int solve(string s) {
    int res = t.size();
    for(int i = 0; i < s.size(); ++i)
        res = min(res, _solve(s, i));

    // cout << s << " " << res << "\n";
    return res;
}

int solve() {
    int res = 0;
    for(string s : a)
        res += solve(s);

    return res;
}

};

const int local = 0, stress = 0;
int32_t main() {
    if(local) freopen("input.txt", "r", stdin);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cout << fixed << setprecision(4);

    ///-----///
    ///-----///

    int tests = 1;

    while(tests--) {
        if(!stress) {
            string t;
            cin >> t;

            int n;
            cin >> n;
```

```

        vector<string> a(n);
        for(int i = 0; i < n; ++i)
            cin >> a[i];

        Solver solver(t, a);
        int res = solver.solve();
        cout << res << "\n";
    } else {}
}

return 0;
}

```

Task D ()

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
typedef long double ld;

const int inf = 1e18;

struct Pos {
    int d, x, y, start;

    Pos(int d, int x, int y, int start): d(d), x(x), y(y), start(start) {}
};

bool operator<(Pos a, Pos b) {
    if(a.d != b.d) return a.d < b.d;
    if(a.x != b.x) return a.x < b.x;
    if(a.y != b.y) return a.y < b.y;
    return false;
}

struct Solver {
    vector<vector<pair<int, int>>> a;
    vector<vector<int>> d;
    int x0, y0, x1, y1;
    int n, m;

    Solver(vector<vector<pair<int, int>>> a, int x0, int y0, int x1, int y1):
        a(a), x0(x0), y0(y0), x1(x1), y1(y1) {
        n = a.size();
        m = a[0].size();

        d.assign(n, vector<int>(m, inf));
    }

    int solve() {
        set<Pos> q;
        q.insert({0, x0, y0, 1});

        while(!q.empty()) {
            Pos v = *q.begin();
            q.erase(q.begin());
            int x = v.x, y = v.y;
            if(d[x][y] != inf) continue;
            d[x][y] = v.d;

            int mx = (v.d < 3 ? 2 : 1);

            /// With
            for(int dx = -mx; dx <= mx; ++dx) {
                for(int dy = -mx; dy <= mx; ++dy) {
                    if(dx == 0 && dy == 0) continue;
                    if(x+dx >= n) continue;
                    if(x+dx < 0) continue;
                    if(y+dy >= m) continue;
                    if(y+dy < 0) continue;

                    int dr = abs(dx-a[x][y].first);
                    int dc = abs(dy-a[x][y].second);

                    if(d[x+dx][y+dy] == inf) q.insert({v.d+dr+dc, x+dx, y+dy, 0});
                }
            }

            /// Without
            if(!v.start) {
                for(int dx = -mx; dx <= mx; ++dx) {
                    for(int dy = -mx; dy <= mx; ++dy) {
                        if(dx == 0 && dy == 0) continue;
                        if(x+dx >= n) continue;
                        if(x+dx < 0) continue;
```

```

        if(y+dy >= m) continue;
        if(y+dy < 0) continue;

        if(d[x+dx][y+dy] == inf) q.insert({v.d+abs(dx)+abs(dy), x+dx, y+dy, 0});
    }
}

return d[x1][y1];
};

const int local = 0, stress = 0;
int32_t main() {
    if(local) freopen("input.txt", "r", stdin);
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cout << fixed << setprecision(4);

    ///-----///
    ///-----///

    int tests = 1;

    while(tests--) {
        if(!stress) {
            int n, m;
            cin >> n >> m;

            int x0, y0, x1, y1;
            cin >> x0 >> y0 >> x1 >> y1;
            x0--; y0--; x1--; y1--;

            vector<vector<pair<int, int>>> a(n, vector<pair<int, int>> (m, {0, 0}));
            for(int i = 0; i < n; ++i)
                for(int j = 0; j < m; ++j)
                    cin >> a[i][j].first >> a[i][j].second;

            Solver solver(a, x0, y0, x1, y1);
            int res = solver.solve();
            cout << res << "\n";

        } else {}
    }

    return 0;
}

```

Task E ()

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
typedef long double ld;

const int inf = 1e18;

struct Pos {
    int x, y;

    Pos(int x, int y): x(x), y(y) {}
};

bool operator<(Pos a, Pos b) {
    if(a.x != b.x) return a.x < b.x;
    if(a.y != b.y) return a.y < b.y;
    return false;
}

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

struct Solver {
    int n, m, b;
    vector<Pos> a;
    vector<Pos> active;
    vector<Pos> to_del;

    Solver(int n, int m, vector<Pos> a): n(n), m(m), a(a) {
        b = a.size();
    }

    void color(Pos a, Pos b) {
        cout << "?_" << a.x << "_" << a.y << "_" << b.x << "_" << b.y << endl;
        int s, t;
        cin >> s >> t;

        to_del.push_back({s, t});
    }

    void solve(int bi) {
        Pos v = a[bi];

        for(int i = 0; i+1 < active.size(); i += 2) {
            Pos u0 = v + active[i+0];
            Pos u1 = v + active[i+1];

            color(u0, u1);
        }

        vector<Pos> _active;
        for(int i = 0; i < active.size(); ++i) {
            int l = active[i].x, r = active[i].x + n;

            int flag = 1;
            for(Pos d : to_del)
                if(l <= d.x && d.x < r) flag = 0;

            cout << "MDA " << bi << ": " << active[i].x << " " << active[i].y << " | " << flag
            << "\n";
            if(flag && _active.size() < pow(2, b-bi-1)) _active.push_back(active[i]);
        }

        active = _active;
        to_del.clear();
    }

    void solve() {
        /// Init
        for(int i = 0; i < pow(2, b); ++i)
            active.push_back({n*i, 0});
    }
};
```

```

        for(int i = 0; i < b; ++i) {
            solve(i);
        }

        assert(active.size() > 0);
        cout << "!_" << active[0].x << "_" << active[1].y << endl;
    }
};

const int local = 0, stress = 0;
int32_t main() {
    // if(local) freopen("input.txt", "r", stdin);
    // ios_base::sync_with_stdio(0);
    // cin.tie(0); cout.tie(0);
    // cout << fixed << setprecision(4);

    ///-----///
    ///-----///

    int tests = 1;

    while(tests--) {
        if(!stress) {
            int n, m;
            cin >> n >> m;
            int b;
            cin >> b;

            vector<Pos> a(b, {0, 0});
            for(int i = 0; i < b; ++i)
                cin >> a[i].x >> a[i].y;

            for(int i = 0; i < b; ++i)
                a[i].x--, a[i].y--;

            Solver solver(n, m, a);
            solver.solve();

        } else {}
    }

    return 0;
}

```

Task F ()

```
#include <bits/stdc++.h>
#define int long long
using namespace std;
typedef long double ld;

const int inf = 1e18;

void write(vector<pair<int, vector<int>>>> a, string fname) {
    ofstream out(fname);
    out << a.size() << "\n";
    for(auto i : a) {
        out << i.first << "\n";
        for(auto j : i.second)
            out << j << ",\n";
        out << "\n";
    }
    out.close();
}

vector<pair<int, vector<int>>>> read(string fname) {
    ifstream inp(fname);
    int k;
    inp >> k;

    vector<pair<int, vector<int>>>> a;

    for(int i = 0; i < k; ++i) {
        int n;
        inp >> n;

        int m = (n*n*n - n) / 6;
        vector<int> b(m, 0);
        for(int j = 0; j < m; ++j)
            inp >> b[j];

        a.push_back({n, b});
    }

    return a;
}

struct Checker {
    int n;
    vector<vector<int>>> g;
    vector<int> clr;

    Checker(vector<vector<int>>> g): g(g) {
        n = g.size();
        clr.assign(n, 0);
    }

    int dfs(int v, int p) {
        clr[v] = 1;

        for(int u : g[v]) {
            if(u == p) continue;
            if(clr[u] == 1) return 1;
            if((clr[u] == 0) && dfs(u, v)) return 1;
        }

        clr[v] = 2;
        return 0;
    }

    int check() {
        if(dfs(0, -1)) return 0;
        for(int i = 0; i < n; ++i)
            if(clr[i] == 0) return 0;

        return 1;
    }
}
```

```

};

struct DP_solver {
    vector<vector<int>>> g;
    int n, res;
    vector<int> dp, sz;

    DP_solver(vector<vector<int>>> g): g(g) {
        n = g.size();
        res = 0;

        dp.assign(n, 0);
        sz.assign(n, 0);
    }

    void dfs(int v, int p) {
        sz[v] = 1;

        for(int u : g[v]) {
            if(u == p) continue;
            dfs(u, v);
            sz[v] += sz[u];
        }

        dp[v] += sz[v] - 1;
        for(int u : g[v]) {
            if(u == p) continue;
            dp[v] += dp[u];

            res += (dp[u] + sz[u]) * ((sz[v] - 1) - sz[u]);
        }

        // cout << "V: " << v << " " << dp[v] << " " << sz[v] << "\n";
        res += dp[v];
    }

    int get_dp() {
        dfs(0, -1);
        return res;
    }
};

struct Stupid {
    int n, m;

    Stupid(int n): n(n), m((n*n*n-n)/6) {
    }

    vector<vector<int>>> get_g(int mask) {
        vector<vector<int>>> g(n);

        for(int v = 0; v < n; ++v) {
            if(mask % n != v) {
                g[v].push_back(mask % n);
                g[mask % n].push_back(v);
            }

            mask /= n;
        }

        for(int v = 0; v < n; ++v) {
            sort(g[v].begin(), g[v].end());
            g[v].erase(unique(g[v].begin(), g[v].end()), g[v].end());
        }

        return g;
    }

    vector<int> solve() {
        set<vector<vector<int>>>> G;

        for(int mask = 0; mask < pow(n, n); ++mask) {
            vector<vector<int>>> g = get_g(mask);

```

