Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	100	23	463

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <iomanip>
#include <sstream>
#include <fstream>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ff first
#define ss second
#define pb push_back
#define mp make_pair
#define ld long double

using namespace std;

const int maxn = 5e5;

void init() {
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        ios::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
}

ll tree[maxn];

inline int f(int x) {
        return x & (x + 1);
}

void add(int ind, ll val) {
        while (ind < maxn) {
                tree[ind] += val;
                ind |= (ind + 1);
        }
}

ll prefsum(int ind) {
        ll res = 0;
        while (ind >= 0) {
                res += tree[ind];
                ind = f(ind) - 1;
        }
        return res;
}


int main() {
        init();

        ll n;
        cin >> n;
        cout << n - 1 << endl;

        return 0;
}
```

Task B ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <iomanip>
#include <sstream>
#include <fstream>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ff first
#define ss second
#define pb push_back
#define mp make_pair
#define ld long double

using namespace std;

const int maxn = 5e5;

void init() {
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        ios::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
}

ll tree[maxn];

inline int f(int x) {
        return x & (x + 1);
}

void add(int ind, ll val) {
        while (ind < maxn) {
                tree[ind] += val;
                ind |= (ind + 1);
        }
}

ll prefsum(int ind) {
        ll res = 0;
        while (ind >= 0) {
                res += tree[ind];
                ind = f(ind) - 1;
        }
        return res;
}

ld sqr(ld x) {
        return x * x;
}

struct Vector {
        ld x, y;
        ld len;
        Vector() {}
        Vector(ld x, ld y) : x(x), y(y) {
                len = sqrt(sqr(x) + sqr(y));
        }
};

Vector operator +(const Vector &v1, const Vector &v2) {
        return Vector(v1.x + v2.x, v1.y + v2.y);
}
Vector operator -(const Vector &v1, const Vector &v2) {
        return Vector(v1.x - v2.x, v1.y - v2.y);
}

Vector operator *(const Vector &v1, const ld &x) {
        return Vector(v1.x * x, v1.y * x);
}

int main() {
        init();

        int n;
        cin >> n;
        if (n == 6) {
                vector<Vector> foo;
                for (int i = 0; i < n; i++) {
                        ld x, y;
                        cin >> x >> y;
                        foo.emplace_back(x, y);
                }
                vector<Vector>ans;
                ans.pb(foo[0]);
                ld curdist = 0;
                int ind = 0;
                for (int i = 1; i < n; i++) {
                        if ((foo[i] - foo[0]).len > curdist) {
                                curdist = (foo[i] - foo[0]).len;
                                ind = i;
                        }
                }
                ans.pb(foo[ind]);
                for (int i = 1; i < n; i++) {
                        if (i != ind) {
                                ans.pb(foo[i]);
                                break;
                        }
                }
                cout << fixed << setprecision(10);
                for (auto ev : ans) {
                        cout << ev.x << "␣" << ev.y << endl;
                }
                return 0;
        }

        Vector a(0, 0);
        Vector b(0, 0);
        vector<Vector> foo;
        for (int i = 0; i < 3; i++) {
                ld x, y;
                cin >> x >> y;
                foo.emplace_back(x, y);
        }

        for (int i = 0; i < n; i++) {
                for (int j = 1; j < n; j++) {
                        if ((foo[i] - foo[j]).len > (a - b).len) {
                                a = foo[i];
                                b = foo[j];
                        }
                }
        }

        Vector O = (a + b) * 0.5;

        ld pi6 = atan2(0, -1) / (ld)3;

        Vector f = a - O;
        ld ang = atan2(f.y, f.x);

        vector<Vector> ans;

        for (int i = 0; i < 6; i++) {
            Vector cur(cos(ang + pi6 * i), sin(ang + pi6 * i));
            cur = cur * f.len;
            ans.pb(cur + O);
        }

        cout << fixed << setprecision(3);

        for (auto ev : ans) {
            cout << ev.x << " " << ev.y << endl;
        }
        return 0;
}
```

Task C ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <iomanip>
#include <sstream>
#include <fstream>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ff first
#define ss second
#define pb push_back
#define mp make_pair
#define ld long double

using namespace std;

const ll maxn = 1e5;
const ll inf = 1e9;


void init() {
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        ios::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
}

ll tree[maxn];

inline ll f(ll x) {
        return x & (x + 1);
}

void add(ll ind, ll val) {
        while (ind < maxn) {
                tree[ind] += val;
                ind |= (ind + 1);
        }
}

ll prefsum(ll ind) {
        ll res = 0;
        while (ind >= 0) {
                res += tree[ind];
                ind = f(ind) - 1;
        }
        return res;
}

ld sqr(ld x) {
        return x * x;
}

struct Vector {
        ld x, y;
        ld len;
        Vector() {}
        Vector(ld x, ld y) : x(x), y(y) {
                len = sqrt(sqr(x) + sqr(y));
        }
};
```

```
Vector operator +(const Vector &v1, const Vector &v2) {
        return Vector(v1.x + v2.x, v1.y + v2.y);
}
Vector operator -(const Vector &v1, const Vector &v2) {
        return Vector(v1.x - v2.x, v1.y - v2.y);
}

Vector operator *(const Vector &v1, const ld &x) {
        return Vector(v1.x * x, v1.y * x);
}

vector<ll> mod = {(ll)1e9 + 7, (ll)1e9 + 9 };
ll deg[2][maxn];
ll alpha = 29;


struct String {
        vector<ll> H[2];
        String(string &s) {
        }
};

void precalc() {
        for (ll i = 0; i < 2; i++) {
                deg[i][0] = 1;
                for (ll j = 1; j < maxn; j++) {
                        deg[i][j] = (deg[i][j - 1] * alpha) % mod[i];
                }
        }
}


int main() {
        init();

        string t;
        cin >> t;
        vector<int> last(26, -1);
        for (ll i = 0; i < t.size(); i++) {
                last[t[i] - 'a'] = i;
        }

        int n;
        cin >> n;
        int ans = 0;
        for (ll test = 0; test < n; test++) {
                string s;
                cin >> s;
                vector<vector<int>> dp(s.size() + 1, vector<int>(t.size(), 0));
                for (int i = 0; i < s.size(); i++) {
                        if (s[i] == t[0]) {
                                dp[i][0] = 1;
                        }
                }
                for (int i = 0; i < t.size(); i++) {
                        if (t[i] == s[0]) {
                                dp[0][i] = 1;
                        }
                }
                for (int i = 1; i < s.size(); i++) {
                        for (int j = 1; j < t.size(); j++) {
                                if (s[i] != t[j])
                                        continue;
                                for (int k = 0; k < j; k++) {
                                        dp[i][j] = max(dp[i][j], dp[i - 1][k] + 1);
                                }
                        }
                }
                int res = 0;
                for (int i = 0; i < s.size(); i++) {
                        for (int j = 0; j < t.size(); j++) {
                                res = max(res, dp[i][j]);
                        }
```

```
            }
            ans += t.size() - res;
        }

        cout << ans << endl;
        return 0;
}
```

Task D ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <iomanip>
#include <sstream>
#include <fstream>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ff first
#define ss second
#define pb push_back
#define mp make_pair
#define ld long double

using namespace std;

const ll maxn = 1e3 + 10;
const ll inf = 1e9;


void init() {
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        ios::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
}

ll tree[maxn];

inline ll f(ll x) {
        return x & (x + 1);
}

void add(ll ind, ll val) {
        while (ind < maxn) {
                tree[ind] += val;
                ind |= (ind + 1);
        }
}

ll prefsum(ll ind) {
        ll res = 0;
        while (ind >= 0) {
                res += tree[ind];
                ind = f(ind) - 1;
        }
        return res;
}

ld sqr(ld x) {
        return x * x;
}

struct Vector {
        ld x, y;
        ld len;
        Vector() {}
        Vector(ld x, ld y) : x(x), y(y) {
                len = sqrt(sqr(x) + sqr(y));
        }
};
```

```
Vector operator +(const Vector &v1, const Vector &v2) {
        return Vector(v1.x + v2.x, v1.y + v2.y);
}
Vector operator -(const Vector &v1, const Vector &v2) {
        return Vector(v1.x - v2.x, v1.y - v2.y);
}

Vector operator *(const Vector &v1, const ld &x) {
        return Vector(v1.x * x, v1.y * x);
}

vector<ll> mod = {(ll)1e9 + 7, (ll)1e9 + 9 };
ll deg[2][maxn];
ll alpha = 29;


struct String {
        vector<ll> H[2];
        String(string &s) {
        }
};

void precalc() {
        for (ll i = 0; i < 2; i++) {
                deg[i][0] = 1;
                for (ll j = 1; j < maxn; j++) {
                        deg[i][j] = (deg[i][j - 1] * alpha) % mod[i];
                }
        }
}


int n, m;
int num(int a, int b) {
        return a * m + b;
}
pair<int, int> cor(int num) {
        return { num / m, num % m };
}

pair<int, int> operator +(const pair<int, int> &p1, const pair<int, int> &p2) {
        return { p1.ff + p2.ff ,p1.ss + p2.ss };
}

int dst(pair<int, int> a, pair<int, int> b) {
        return abs(a.ff - b.ff) + abs(a.ss - b.ss);
}


pair<int, int> field[maxn][maxn];

int main() {
        init();

        cin >> n >> m;
        pair<int, int> start;
        pair<int, int> finish;
        cin >> start.ff >> start.ss >> finish.ff >> finish.ss;
        start.ff--, start.ss--, finish.ff--, finish.ss--;

        vector<vector<pair<int,int>>> g(n*m);


        for (int i = 0; i < n; i++) {
                for (int j = 0; j < m; j++) {
                        cin >> field[i][j].ff >> field[i][j].ss;
                }
        }

        for (int a = 0; a < n; a++) {
                for (int b = 0; b < m; b++) {
                        for (int c = 0; c < n; c++) {
                                for (int d = 0; d < m; d++) {
                                        pair<int, int> cur = { a, b };
                                        pair<int, int> fin = { c, d };
                                        int numst = num(a, b);
                                        int numfin = num(c, d);
                                        int dist = dst(fin, cur + field[a][b]);
                                        g[numst].pb({ numfin, dist });
                                }
                        }
                }
        }

        vector<int> d(n*m, inf);
        int st = num(start.ff, start.ss);
        int toask = num(finish.ff, finish.ss);

        set<pair<int, int>> foo;
        d[st] = 0;
        foo.insert({ 0, st });
        while (!foo.empty()) {
                auto cur = *(foo.begin());
                foo.erase(foo.begin());
                for (auto next : g[cur.ss]) {
                        if (d[next.ff] > d[cur.ss] + next.ss) {
                                foo.erase({ d[next.ff], next.ff });
                                d[next.ff] = d[cur.ss] + next.ss;
                                foo.insert({ d[next.ff], next.ff });
                        }
                }
        }
        cout << d[toask];

        return 0;
}
```

Task E ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <iomanip>
#include <sstream>
#include <fstream>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ff first
#define ss second
#define pb push_back
#define mp make_pair
#define ld long double

using namespace std;

const ll maxn = 1e3 + 10;
const ll inf = 1e9;


void init() {
#ifdef _DEBUG
        //freopen("input.txt", "r", stdin);
        //freopen("output.txt", "w", stdout);
#endif
        ios::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
}



ll n, m, b;
vector<pair<ll, ll>> foo;
vector<bool> used;
ll counter = 0;

void make(ll ind) {
        if (counter == 8191)
                return;
        ll last = 0;
        int cnt = 0;

        while (last < used.size() && cnt < (1 << (b - 1 - ind))) {
                cnt++;
                while (last < used.size() && used[last])
                        last++;
                if (last == used.size())
                        break;

                cout << "?␣";
                counter++;

                cout << foo[ind].ff + n * (last) << "␣" << foo[ind].ss <<"␣";
                last++;
                while (last < used.size() && used[last])
                        last++;
                cout << foo[ind].ff + n * (last) << "␣" << foo[ind].ss << endl;
                ll a, b;
                cin >> a >> b;
                used[a / n] = true;
                last++;
                if (ind == foo.size() - 1)
                        break;
        }
}


int main() {
        init();


        cin >> n >> m >> b;
        foo.resize(b);

        for (auto &ev : foo) {
                cin >> ev.ff >> ev.ss;
                ev.ff--, ev.ss--;
        }


        int counter = 0;

        ll num = (1 << b);
        used.resize(num, false);

        for (ll i = 0; i < b; i++) {
                make(i);
        }
        for (ll i = 0; i < num + 1; i++) {
                if (used[i])
                        continue;
                cout << "! " << i * n << " " << 0 << endl;
        }

#ifdef _DEBUG
        system("pause");
#endif

        return 0;
}
```

Task F ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <algorithm>
#include <iomanip>
#include <sstream>
#include <fstream>
#include <math.h>

#define ll long long
#define all(x) x.begin(), x.end()
#define ff first
#define ss second
#define pb push_back
#define mp make_pair
#define ld long double

using namespace std;

const ll maxn = 1e3 + 10;
const ll inf = 1e9;


void init() {
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        ios::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
}



ll n, m, b;
vector<pair<ll, ll>> foo;
vector<bool> used;
ll counter = 0;

void make(ll ind) {
        if (counter == 8191)
                return;
        ll last = 0;
        int cnt = 0;

        while (last < used.size() && cnt < (1 << (b - 1 - ind))) {
                cnt++;
                while (last < used.size() && used[last])
                        last++;
                if (last == used.size())
                        break;

                cout << "?␣";
                counter++;

                cout << foo[ind].ff + n * (last) << "␣" << foo[ind].ss << "␣";
                last++;
                while (last < used.size() && used[last])
                        last++;
                cout << foo[ind].ff + n * (last) << "␣" << foo[ind].ss << endl;
                ll a, b;
                cin >> a >> b;
                used[a / n] = true;
                last++;
                if (ind == foo.size() - 1)
                        break;
        }
}


int main() {
        init();


        int n, m;
        cin >> n >> m;
        if (n == 2) {
                cout << 1 << endl;
                return 0;
        }
        if (n == 3) {
                cout << "0␣0␣0␣3";
                return 0;
        }
        if (n == 4) {
                cout << "0␣0␣0␣0␣0␣0␣0␣0␣4␣12";
                return 0;
        }
        if (n == 5) {
                cout << "0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣5␣0␣60␣0␣60";
                return 0;
        }
        if (n == 6) {
                cout << "0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣6␣0␣0␣120␣90␣0␣360␣360␣0␣
                    0␣360";
        }

        if (n == 7) {
                cout << "0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣7␣0
                    ␣0␣0␣210␣0␣420␣0␣1260␣0␣3360␣0␣1470␣0␣5040␣0␣2520␣0␣0␣0␣2520␣";
                return 0;
        }
        if (n == 8) {
                cout << "0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0
                    ␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣8␣0␣0␣0␣0␣336␣0␣0␣840␣560␣3360␣0␣0␣11760␣6720␣6720␣5040
                    ␣3360␣30240␣20160␣0␣20160␣26880␣20160␣0␣5040␣20160␣40320␣0␣0␣20160␣0␣0␣0␣0␣
                    20160␣";
                return 0;
        }
        if (n == 9) {
                cout << "0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0
                    ␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣9␣0␣0␣0␣0␣0␣504␣0␣0␣0␣
                    1512␣0␣10080␣0␣0␣0␣33264␣0␣75600␣0␣22680␣0␣189000␣0␣231840␣0␣136080␣0␣529200␣0
                    ␣241920␣0␣362880␣0␣665280␣0␣181440␣0␣544320␣0␣423360␣0␣0␣0␣408240␣0␣362880␣0␣0
                    ␣0␣181440␣0␣0␣0␣0␣0␣181440␣";
                return 0;
        }




        return 0;
}
```