

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	6	9	415

Task A ()

```
///#pragma GCC optimize("Ofast,unroll-loops")
///#pragma GCC target("sse,sse2,sse3,popcnt,tune=native")
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef long double ld;

const ld pi = acos(-1);
const ld eps = 1e-10;

#define all(x) x.begin(), x.end()
#define db(x) cout << #x << " = " << x << "\n";

void fast()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
}

int main()
{
    ll n;
    cin >> n;
    cout << n - 1;
}
```

Task B ()

```
///#pragma GCC optimize("Ofast,unroll-loops")
///#pragma GCC target("sse,sse2,sse3,popcnt,tune=native")
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef long double ld;

const ld pi = acos(-1);
const ld eps = 1e-10;

#define all(x) x.begin(), x.end()
#define db(x) cout << #x << " = " << x << "\n";

class Point
{
    friend istream &operator>>(istream &in, Point &p)
    {
        in >> p.x >> p.y;
        return in;
    }
    friend ostream &operator<<(ostream &out, Point p)
    {
        out << p.x << " " << p.y;
        return out;
    }
public:
    ld x, y;
    Point(ld x1 = 0, ld y1 = 0)
    {
        x = x1;
        y = y1;
    }
    Point operator+(Point a)
    {
        return Point(x + a.x, y + a.y);
    }
    Point operator-(Point a)
    {
        return Point(x - a.x, y - a.y);
    }
    Point operator*(ld a)
    {
        return Point(x * a, y * a);
    }
    Point operator/(ld a)
    {
        return Point(x / a, y / a);
    }
    ld operator*(Point a)
    {
        return x * a.y - y * a.x;
    }
    ld operator%(Point a)
    {
        return x * a.x + y * a.y;
    }
    bool operator==(Point a)
    {
        return ((abs(x - a.x) < eps) && (abs(y - a.y) < eps));
    }
    bool operator<(const Point &a) const
    {
        if (x < a.x)
            return true;
        if (x > a.x)
            return false;
        if (y < a.y)
            return true;
        return false;
    }
}
```

```

};

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

int sig(ld x)
{
    if (abs(x) < eps)
        return 0;
    if (x > 0)
        return 1;
    return -1;
}

ld abs(Point a)
{
    return sqrt(sqr(a.x) + sqr(a.y));
}

ld dist(Point a, Point b)
{
    return abs(a - b);
}

void fast()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
}

void print(vector<ll> A)
{
    for (int i = 0; i < A.size(); i++)
        cout << A[i] << " ";
    cout << "\n";
}

int main()
{
    ll n;
    cin >> n;
    vector<Point> A1(n);
    for (int i = 0; i < n; i++)
    {
        cin >> A1[i];
    }
    if (n == 6)
    {
        Point O(0, 0);
        for (int i = 0; i < n; i++)
        {
            O = O + A1[i];
        }
        O = O / 6.0;
        ll id = -1;
        for (int i = 1; i < n; i++)
        {
            if (id == -1 || dist(A1[id], A1[0]) > dist(A1[i], A1[0]))
            {
                id = i;
            }
        }
        cout << fixed << setprecision(10) << O << "\n" << A1[0] << "\n" << A1[id];
        return 0;
    }

    Point O, A, B;
    O = A1[0];
    A = A1[1];
    B = A1[2];
    Point D = O - A + O;

```

```

Point C = O - A + B;
Point E = O - B + O;
Point F = O - C + O;
cout << fixed << setprecision(10) << A << "\n" << B << "\n" << C << "\n" << D << "\n" << E <<
    "\n" << F;
}

```

Task C ()

```
///#pragma GCC optimize("Ofast,unroll-loops")
///#pragma GCC target("sse,sse2,sse3,popcnt,tune=native")
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef long double ld;

const ld pi = acos(-1);
const ld eps = 1e-10;

#define all(x) x.begin(), x.end()
#define db(x) cout << #x << " = " << x << "\n";

void fast()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
}

ll getkol(string s, string t)
{
    ll ykx = 0, ykt = 0;
    ll ans = 0;
    while (true)
    {
        if (ykx >= int(s.size()) || ykt >= int(t.size()))
            break;
        if (s[ykx] == t[ykt])
        {
            ykx++;
            ykt++;
            continue;
        }
        ans++;
        ykt++;
    }
    if (ykt >= t.size())
        return ans;
    return ans + int(t.size()) - ykt;
}

ll gkol(string s, string t)
{
    ll minn = 1e10;
    ll n = s.size();
    for (int i = 0; i < n; i++)
    {
        minn = min(minn, getkol(s.substr(i, n - i), t));
    }
    return minn;
}

int main()
{
    fast();
    string t;
    cin >> t;
    ll n;
    cin >> n;
    ll ans = 0;
    for (int i = 0; i < n; i++)
    {
        string a;
        cin >> a;
        ll g = gkol(a, t);
        ans += g;
    }
    cout << ans;
}
```

Task D ()

```
#pragma GCC optimize("Ofast,unroll-loops")
///#pragma GCC target("sse,sse2,sse3,popcnt,tune=native")
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef long double ld;

#define all(x) x.begin(), x.end()
#define db(x) cout << #x << "==" << x << "\n";

void fast()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
}

int n, m;
vector<vector<int>> > dx;
vector<vector<int>> > dy;

int R;

/**
1 5
1 1 1 5
0 0 0 0 0 0 0 0 0 0
*/

int main()
{
    fast();
    cin >> n >> m;
    if (n * m >= 10000)
    {
        R = 5;
    }
    else
    {
        R = 15;
    }
    int x0, y0, x1, y1;
    cin >> x0 >> y0 >> x1 >> y1;
    x0--;
    y0--;
    x1--;
    y1--;
    dx.resize(n, vector<int> (m));
    dy.resize(n, vector<int> (m));
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            cin >> dx[i][j] >> dy[i][j];
        }
    }
    vector<vector<int>> > d(n, vector<int> (m, 1e9));
    d[x0][y0] = 0;
    set<pair<int, int>> > S;
    S.insert({0, x0 * m + y0});
    while (!S.empty())
    {
        int v = (*S.begin()).second;
        S.erase(S.begin());
        int x = v / m;
        int y = v % m;
        if (x == x1 && y == y1)
        {
            cout << d[x1][y1];
            return 0;
        }
    }
}
```

```

    }
    for (int i = max(0, x - R); i < min(n, x + R); i++)
    {
        for (int j = max(0, y - R); j < min(m, y + R); j++)
        {
            int dist = abs(x + dx[x][y] - i) + abs(y + dy[x][y] - j);
            if (d[i][j] > d[x][y] + dist)
            {
                pair<int, int> r;
                r.first = d[i][j];
                r.second = i * m + j;
                S.erase(r);
                r.first = d[x][y] + dist;
                S.insert(r);
                d[i][j] = d[x][y] + dist;
            }
        }
    }
}
cout << d[x1][y1];
}

```

Task E ()

```
///#pragma GCC optimize("Ofast,unroll-loops")
///#pragma GCC target("sse,sse2,sse3,popcnt,tune=native")
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef long double ld;

const ld pi = acos(-1);
const ld eps = 1e-10;

#define all(x) x.begin(), x.end()
#define db(x) cout << #x << " = " << x << "\n";

void fast()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
}

set<pair<ll, ll>> black;
map<pair<ll, ll>, bool> dir;

pair<ll, ll> zap(ll x1, ll y1, ll x2, ll y2)
{
    cout << "? " << x1 << " " << y1 << " " << x2 << " " << y2 << endl;
    ll x, y;
    cin >> x >> y;
    black.insert({x1, y1});
    black.insert({x2, y2});
    black.erase({x, y});
    dir[{x, y}] = true;
    return {x, y};
}

const ll M = 1e14;
vector<pair<ll, ll>> ny;
ll n, m, b;
bool che(ll x1, ll y1, ll x2, ll y2)
{
    for (ll i = x1 - ny[0].first + 1; i <= x1 - ny[0].first + n; i++)
    {
        for (ll j = y1 - ny[0].second + 1; j <= y1 - ny[0].second + m; j++)
        {
            if (dir[{i, j}])
                return false;
        }
    }
    return true;
}

ll dx, dy;

void check()
{
    vector<pair<ll, ll>> V;
    for (auto i : black)
        V.push_back(i);
    for (int i = 0; i < V.size(); i++)
    {
        for (int j = 0; j < V.size(); j++)
        {
            if (dir[V[i]] || dir[V[j]])
                continue;
            if (i == j)
                continue;
            if (V[i].first + dx == V[j].first && V[i].second + dy == V[j].second && che(V[i].first, V[i].second, V[j].first, V[j].second))
            {
                cout << "! ";
            }
        }
    }
}
```



```

        cout << V[i].first - ny[0].first + 1 << "┘" << V[i].second - ny[0].second + 1 <<
            endl;
        exit(0);
    }
}
}

int main()
{
    srand(time(0));

    cin >> n >> m >> b;
    ny.resize(b);
    ll minx = 1e10;
    ll miny = 1e10;
    for (int i = 0; i < b; i++)
    {
        cin >> ny[i].first >> ny[i].second;
        minx = min(minx, ny[i].first);
        miny = min(miny, ny[i].second);
    }
    if (b == 1)
    {
        zap(-M, -M, M, M);
        for (auto i : black)
        {
            ll x = i.first;
            ll y = i.second;
            cout << "!┘";
            cout << x - ny[0].first + 1 << "┘" << y - ny[0].second + 1 << endl;
            return 0;
        }
    }
    if (b == 2)
    {
        dx = (ny[1].first - ny[0].first);
        dy = ny[1].second - ny[0].second;
        pair<ll, ll> a = zap(-M, -M, -M + dx, -M + dy);
        check();
        pair<ll, ll> a1 = zap(M, M, M + dx, M + dy);
        check();
        pair<ll, ll> sh1, sh2;
        if (dir[ {-M, -M}])
        {
            sh1.first = -M + 2 * dx;
            sh1.second = -M + 2 * dy;
        }
        else
        {
            sh1.first = -M - dx;
            sh1.second = -M - dy;
        }

        if (dir[ {M, M}])
        {
            sh2.first = M + 2 * dx;
            sh2.second = M + 2 * dy;
        }
        else
        {
            sh2.first = M - dx;
            sh2.second = M - dy;
        }
        a = zap(sh1.first, sh1.second, sh2.first, sh2.second);
        check();
        while (true)
        {
            ll X = rand() % (M * 2) - M;
            ll Y = rand() % (M * 2) - M;

            ll X1 = rand() % (M * 2) - M;
            ll Y1 = rand() % (M * 2) - M;

```

```
        zap(X, Y, X + dx, Y + dy);
        check();
        zap(X1, Y1, X1 + dx, Y1 + dy);
        check();
    }
    assert(false);
    return 0;
}
```

Task F ()

```
///#pragma GCC optimize("Ofast,unroll-loops")
///#pragma GCC target("sse,sse2,sse3,popcnt,tune=native")
#include <bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef long double ld;

const ld pi = acos(-1);
const ld eps = 1e-10;

#define all(x) x.begin(), x.end()
#define db(x) cout << #x << "==" << x << "\n";

void fast()
{
    ios::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
}

vector<vector<ll>>> D;

bool rav(vector<ll> &a, vector<ll> &b)
{
    map<pair<ll, ll>, bool> f;
    for (int i = 0; i < a.size(); i++)
    {
        ll v = i;
        ll to = a[i];
        if (to == -1)
            continue;
        f[{v, to}] = 1;
        f[{to, v}] = 1;
    }
    for (int i = 0; i < b.size(); i++)
    {
        ll v = i;
        ll to = b[i];
        if (to == -1)
            continue;
        if (f[{v, to}] == 0)
            return false;
    }
    return true;
}

ll dfs(ll v, vector<vector<ll>>> &g, ll pr = -1, ll h = 0)
{
    ll ans = h;
    for (int i = 0; i < g[v].size(); i++)
    {
        ll to = g[v][i];
        if (to == pr)
            continue;
        ans += dfs(to, g, v, h + 1);
    }
    return ans;
}

ll n, m;
ll sd(vector<ll> a)
{
    vector<vector<ll>>> g(n);
    for (int i = 0; i < a.size(); i++)
    {
        ll v = i;
        ll to = a[i];
        if (to == -1)
            continue;
        g[v].push_back(to);
    }
}
```

```

        g[to].push_back(v);
    }
    ll ans = 0;
    for (int i = 0; i < n; i++)
    {
        ans += dfs(i, g);
    }
    if (ans % 2 != 0)
        assert(false);
    return ans / 2;
}

vector<ll> A;

vector<ll> pr;
vector<bool> used;

bool add(vector<ll> &y)
{
    for (int i = 0; i < D.size(); i++)
    {
        if (rav(y, D[i]))
            return false;
    }
    return true;
}

bool cic(vector<ll> &pr)
{
    /*cout << "          incic\n";
    cout << "p: ";
    for (int i = 0; i < pr.size(); i++)
        cout << pr[i] << " ";
    cout << "\n";*/
    vector<bool> used(n, false);
    for (int i = 0; i < n; i++)
    {
        if (used[i])
            continue;
        used[i] = true;
        ll i1 = i;
        map<ll, ll> is;
        is[i] = 1;

        while (true)
        {
            i1 = pr[i1];
            if (i1 == -1)
                break;
            if (is[i1])
                return true;
            is[i1] = true;
            used[i1] = true;
        }
    }
    //cout << "          endcic\n";
    return false;
}

void rec(ll v)
{
    if (v == n)
    {
        vector<ll> per(n);
        for (int i = 0; i < n; i++)
            per[i] = i;
        vector<ll> pr1 = pr;
        do
        {
            bool is = false;
            bool is0 = false;
            for (int i = 1; i < n; i++)
            {

```

```

        pr1[i] = per[pr[i]];
        if (pr1[i] == 0)
        {
            is0 = true;
        }
        if (pr1[i] == i)
        {
            is = true;
            break;
        }
    }
    if (is || !is0 || cic(pr1))
    {
        continue;
    }
    if (add(pr1))
    {
        cout << "+1\n";
        D.push_back(pr1);
    }
} while (next_permutation(per.begin(), per.end()));
cout << "endperm\n";
return;
}
for (int p = 0; p <= v - 1; p++)
{
    pr[v] = p;
    rec(v + 1);
}
}

int main()
{
    cin >> n >> m;
    A.resize(m + 1, 0);
    if (n == 2)
    {
        cout << "1";
        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_5_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_6_0_0_120_90_0_360_360_0_0_360";
        return 0;
    }
}
}

```