

# Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	12	0	352

## Task A ()

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

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

## Task B ()

```
#include <bits/stdc++.h>
#include <iostream>
using namespace std;

const double PI = 3.141592;

int main()
{
    int n;
    cin >> n;
    if (n == 6)
    {
        vector<pair<double, double>> p(n);
        for (int i = 0; i < n; i++)
            cin >> p[i].first >> p[i].second;
        double x0 = 0.0, y0 = 0.0;
        for (int i = 0; i < 6; i++)
            x0 += p[i].first, y0 += p[i].second;
        x0 /= 6.0, y0 /= 6.0;
        cout << x0 << '\n' << y0 << '\n';
        cout << p[0].first << '\n' << p[0].second << '\n';
        cout << p[1].first << '\n' << p[1].second << '\n';
    }
    else
    {
        vector<pair<double, double>> p(n);
        vector<pair<double, double>> p_res(2 * n);
        for (int i = 0; i < n; i++)
            cin >> p[i].first >> p[i].second;
        double x0 = p[0].first, y0 = p[0].second;
        double x1 = p[1].first, y1 = p[1].second;
        double x2 = p[2].first, y2 = p[2].second;
        double alpha = atan2(y1 - y0, x1 - x0);
        double r = (sqrt((x2 - x0) * (x2 - x0) + (y2 - y0) * (y2 - y0)) + sqrt((x1 - x0) * (x1 - x0) + (y1 - y0) * (y1 - y0))) / 2.0;
        for (int i = 0; i < 2 * n; i++)
        {
            double a = alpha + i * PI / 3.0;
            p_res[i].first = x0 + r * cos(a);
            p_res[i].second = y0 + r * sin(a);
        }
        for (int i = 0; i < 2 * n; i++)
            cout << p_res[i].first << '\n' << p_res[i].second << '\n';
    }
    return 0;
}
```

## Task C ()

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

int main()
{
    string t;
    cin >> t;
    int N;
    cin >> N;
    int res = 0;
    for (int it = 0; it < N; it++)
    {
        string c;
        cin >> c;
        int cres = t.size();
        for (int i = 0; i < c.size(); i++)
        {
            int cnt = 0;
            int l = i;
            for (int j = 0; j < t.size(); j++)
            {
                if (l == c.size())
                {
                    cnt += 1;
                    continue;
                }
                if (t[j] == c[l])
                    l++;
                else
                    cnt++;
            }
            cres = min(cres, cnt);
        }
        //cerr << cres << endl;
        res += cres;
    }
    cout << res << endl;
    return 0;
}
```

## Task D ()

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

int main()
{
    //freopen("D.in", "r", stdin);
    int n, m;
    cin >> n >> m;
    int xs, ys, xf, yf;
    cin >> xs >> ys >> xf >> yf;
    xs--, ys--, xf--, yf--;
    vector<vector<int>> dx(n, vector<int>(m));
    vector<vector<int>> dy(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];
    if (n == 1)
    {
        vector<int> dist(m, -1);
        for (int i = 0; i < m; i++)
            dist[i] = abs(ys + dy[0][ys] - i) + abs(dx[0][ys]);
        dist[ys] = 0;
        vector<bool> used(m, false);
        for (int i = 0; i < m; i++)
        {
            int ind = 0;
            while(used[ind])
                ind++;
            for (int j = ind; j < m; j++)
                if (!used[j] && dist[j] < dist[ind])
                    ind = j;
            //cerr << ind << endl;
            used[ind] = true;
            for (int j = 0; j < m; j++)
                dist[j] = min(dist[j], dist[ind] + abs(ind + dy[0][ind] - j) + abs(dx[0][ind]));
        }
        /*for (int i = 0; i < m; i++)
            cerr << dist[i] << ' ';
        cerr << endl;*/
        cout << dist[yf] << endl;
        return 0;
    }
    vector<vector<int>> dist(n, vector<int>(m));
    vector<vector<bool>> used(n, vector<bool>(m, false));
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            dist[i][j] = abs(xs + dx[xs][ys] - i) + abs(ys + dy[xs][ys] - j);
    dist[xs][ys] = 0;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
        {
            int mix = -1, miy = 0;
            for (int x = 0; x < n; x++)
                for (int y = 0; y < m; y++)
                    if (!used[x][y])
                        if (mix == -1 || dist[x][y] < dist[mix][miy])
                            mix = x, miy = y;
            used[mix][miy] = true;
            for (int x = 0; x < n; x++)
                for (int y = 0; y < m; y++)
                    dist[x][y] = min(dist[x][y], dist[mix][miy] + abs(mix + dx[mix][miy] - x) + abs(miy + dy[mix][miy] - y));
        }
    /*for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
            cerr << dist[i][j] << ' ';
        cerr << endl;
    }*/
    cout << dist[xf][yf] << endl;
    return 0;
}
```

## Task E ()

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n, m, b;
    cin >> n >> m >> b;
    vector<pair<int, int>> p(b);
    for (int i = 0; i < b; i++)
        cin >> p[i].first >> p[i].second, —p[i].first, —p[i].second;
    /*if (b == 1)
    {
        cout << "? " << p[0].first << ' ' << p[0].second << ' ' << n + p[0].first << ' ' << p[0].
            second << endl;
        fflush(stdout);
        int x, y;
        cin >> x >> y;
        if (x == p[0].first)
            cout << "! " << n << ' ' << 0 << endl;
        else
            cout << "! 0 0\n";
        return 0;
    }*/
    vector<bool> is_alive;
    vector<int> i_p;
    int alive_cnt = 0;
    while (true)
    {
        for (int i = 0; i < i_p.size(); i++)
            if (is_alive[i] && i_p[i] == b)
            {
                cout << "!_" << n * (i + 1) << ' ' << 0 << endl;
                fflush(stdin);
                return 0;
            }
        int mix = -1;
        int c = 0;
        for (int i = 0; i < i_p.size(); i++)
            if (is_alive[i])
                if (mix == -1 || i_p[mix] >= i_p[i])
                {
                    if (i_p[mix] == i_p[i])
                        c++;
                    else c = 0;
                    mix = i;
                }
        if (c >= 2)
        {
            int t = i_p[mix];
            int ma = -1;
            for (int i = 0; i < i_p.size(); i++)
                if (is_alive[i])
                    if (ma == -1 || i_p[i] == t)
                        ma = i;
            cout << "?_" << (ma + 1) * n + p[i_p[ma]].first << ' ' << p[i_p[ma]].second;
            i_p[ma]++;

            ma = -1;
            for (int i = 0; i < i_p.size(); i++)
                if (is_alive[i])
                    if (ma == -1 || i_p[i] == t)
                        ma = i;
            cout << ' ' << (ma + 1) * n + p[i_p[ma]].first << ' ' << p[i_p[ma]].second << endl;
            i_p[ma]++;
        }
        else if (alive_cnt < 4)
        {
            is_alive.emplace_back(true);
            i_p.emplace_back(1);
            cout << "?_" << p[0].first + i_p.size() * n << ' ' << p[0].second;
        }
    }
}

```

```

        is_alive.emplace_back(true);
        i_p.emplace_back(1);
        cout << '└' << p[0].first + i_p.size() * n << '└' << p[0].second << endl;

        fflush(stdout);
        alive_cnt += 2;
    }
    /*else if (alive_cnt == 10)
    {
        is_alive.emplace_back(true);
        i_p.emplace_back(1);
        cout << "? " << p[0].first + i_p.size() * n << ' ' << p[0].second;
        alive_cnt++;
        int mi = -1;
        for (int i = 0; i < i_p.size(); i++)
            if (is_alive[i])
                if (mi == -1 || i_p[i] < i_p[mi])
                    mi = i;
        cout << ' ' << (mi + 1) * n + p[i_p[mi]].first << ' ' << p[i_p[mi]].second << endl;
        i_p[mi]++;
        fflush(stdin);
    }*/
    else
    {
        int mi = -1;
        for (int i = 0; i < i_p.size(); i++)
            if (is_alive[i])
                if (mi == -1 || i_p[i] < i_p[mi])
                    mi = i;
        cout << "?└" << (mi + 1) * n + p[i_p[mi]].first << '└' << p[i_p[mi]].second;
        i_p[mi]++;

        mi = -1;
        for (int i = 0; i < i_p.size(); i++)
            if (is_alive[i])
                if (mi == -1 || i_p[i] < i_p[mi])
                    mi = i;

        cout << '└' << (mi + 1) * n + p[i_p[mi]].first << '└' << p[i_p[mi]].second << endl;
        i_p[mi]++;

        fflush(stdin);
    }
    int x, y;
    cin >> x >> y;
    if (is_alive[x / n - 1])
        alive_cnt--;
    is_alive[x / n - 1] = false;
    for (int i = 0; i < i_p.size(); i++)
        if (is_alive[i])
            cerr << '└' << i_p[i] << '└';
        else cerr << -i_p[i] << '└';
    cerr << endl;
}
return 0;
}

```

## Task F ()

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

const int DIV = 1e9 + 7;

int main()
{
    int n, m;
    cin >> n >> m;
    vector<long long> fact(n);
    fact[1] = 1;
    for (int i = 2; i < n; i++)
        fact[i] = fact[i - 1] * i % DIV;
    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_4_12";
        return 0;
    }
    if (n == 5)
    {
        for (int i = 0; i < 15; i++)
            cout << "0_";
        cout << "5_0_120_0_60";
        return 0;
    }
    return 0;
}
```