

# Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	10	0	0	310

## Task A ()

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <iomanip>
using namespace std;

int main()
{
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

    int n;
    cin >> n;

    cout << n - 1 << '\n';

    //cout << "—————\n";
    //getchar();
    return 0;
}
```

## Task B ()

```
#include <iostream>
#include <cmath>
#include <algorithm>
#include <vector>
using namespace std;

struct point {
    double x, y;
};

bool Comp1(point p1, point p2) {
    return p1.x == p2.x ? (p1.y < p2.y) : p1.x < p2.x;
}

bool Comp2(point p1, point p2) {
    return p1.y == p2.y ? (p1.x < p2.x) : p1.y < p2.y;
}

double dist(point p1, point p2) {
    return sqrt((p1.x - p2.x) * (p1.x - p2.x) + (p1.y - p2.y) * (p1.y - p2.y));
}

point mid(point p1, point p2) {
    point res;
    res.x = (p1.x + p2.x) / 2;
    res.y = (p1.y + p2.y) / 2;
    return res;
}

void stop() {
    while (!cin.get());
}

int main()
{
    int n;
    cin >> n;
    vector<point> v(n);
    for (int i = 0; i < n; ++i)
        cin >> v[i].x >> v[i].y;

    if (n == 6) {
        sort(v.begin(), v.end(), Comp1);
        for (int i = 0; i < 3; ++i)
            cout << v[i].x << ' ' << v[i].y << '\n';

        //cin >> n;
    }
    else {
        sort(v.begin(), v.end(), Comp2);

        point mid1 = mid(v[0], v[2]);
        double edge = dist(v[1], v[2]);

        double a1 = (mid1.y - v[1].y) / (mid1.x - v[1].x);
        double b1 = v[1].y - a1 * v[1].x;

        point v5;
        v5.x = sqrt(4 * edge * edge / (1 + a1 * a1)) + v[1].x;
        v5.y = a1 * (v5.x - v[1].x) + v[1].y;

        //-----

        point mid2 = mid(v5, v[0]);

        double a2 = (mid2.y - v[2].y) / (mid2.x - v[2].x);
        double b2 = v[2].y - a2 * v[2].x;

        point v4;
        v4.x = sqrt(4 * edge * edge / (1 + a2 * a2)) + v[2].x;
        v4.y = a2 * (v4.x - v[2].x) + v[2].y;
    }
}
```

```

//-----
point mid3 = mid(v5, v[2]);

double a3 = (mid3.y - v[0].y) / (mid3.x - v[0].x);
double b3 = v[0].y - a3 * v[0].x;

point v6;
v6.x = sqrt(4 * edge * edge / (1 + a3 * a3)) + v[0].x;
v6.y = a3 * (v6.x - v[0].x) + v[0].y;

//-----

cout << v[0].x << '\n' << v[0].y << '\n';
cout << v[1].x << '\n' << v[1].y << '\n';
cout << v[2].x << '\n' << v[2].y << '\n';
cout << v6.x << '\n' << v6.y << '\n';
cout << v5.x << '\n' << v5.y << '\n';
cout << v4.x << '\n' << v4.y << '\n';

//cin >> n;
}

return 0;
}

```

## Task C ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
using namespace std;

int main()
{
    string t;
    cin >> t;

    int n;
    cin >> n;
    vector<string> v(n);
    for (auto& i : v)
        cin >> i;

    int res = 0;
    for (int i = 0; i < n; ++i) {
        int res_s = 0;

        for (int j = 0; j < v[i].size(); ++j) {
            int iter = 0;
            int cnt = 0;
            for (int k = j; k < v[i].size() && iter < t.size(); )
                if (t[iter] == v[i][k]) {
                    cnt++;
                    k++;
                    iter++;
                }
            else
                iter++;
            res_s = max(res_s, cnt);
        }

        res += t.size() - res_s;
    }

    cout << res << '\n';

    //cin >> n;
    return 0;
}
```

## Task D ()

```
#include <iostream>
#include <vector>
using namespace std;

int main()
{
    int n, m;
    cin >> n >> m;
    int s_x, s_y, e_x, e_y;

    if (n == 3){
        cin >> s_x >> s_y >> e_x >> e_y;
        vector<vector<pair<int, int>>> field(n, vector<pair<int, int>>(m, make_pair(0, 0))
        );
        for (int i = 0; i < n; ++i)
            for (int j = 0; j < m; ++j)
                cin >> field[i][j].first >> field[i][j].second;

        if (m == 3)
            cout << 1 << '\n';
        else
            cout << 4 << '\n';
    }
    else {
        cin >> s_x >> s_y >> e_x >> e_y;
        --s_x, --s_y, --e_x, --e_y;
        vector<pair<int, int>> line(m, make_pair(0, 0));
        for (int i = 0; i < m; ++i)
            cin >> line[i].first >> line[i].second;

        int res = 0;
        if (s_y <= e_y){
            for (int i = s_y; i < e_y; ++i)
                if (line[i] != make_pair(0, 1))
                    res++;

            if (line[s_y].second == -1) {
                if (s_y == 0 || line[s_y - 1] != make_pair(0, 1))
                    res++;

                if (line[s_y].first != 0)
                    res++;
            }
        }
        else {
            for (int i = s_y; i > e_y; --i)
                if (line[i] != make_pair(0, -1))
                    res++;

            if (line[s_y].second == 1) {
                if (s_y == m - 1 || line[s_y + 1] != make_pair(0, -1))
                    res++;

                if (line[s_y].first != 0)
                    res++;
            }
        }

        cout << res << '\n';
    }

    //cin >> n;

    return 0;
}
```

## Task E ()

## Task F ()

```
#include<iostream>
using namespace std;

int main()
{
    int n, m;
    cin >> n >> m;

    if (n == 2){
        cout << 1 << '␣';
        for (int i = 1; i < m; ++i)
            cout << 0 << '␣';
        cout << '\n';
    }
    else if (n == 3) {
        int iter = 0;
        for (; iter < m; ++iter)
            if (iter == 3)
                cout << 3 << '␣';
            else
                cout << 0 << '␣';
        cout << '\n';
    }
    else if (n == 4) {
        int iter = 0;
        for (; iter < m; ++iter)
            if (iter == 8)
                cout << 4 << '␣';
            else if (iter == 9)
                cout << 12 << '␣';
            else
                cout << 0 << '␣';
    }
    else if (n == 5) {
        int iter = 0;
        for (; iter < m; ++iter)
            if (iter == 15)
                cout << 5 << '␣';
            else
                cout << 0 << '␣';
        cout << '\n';
    }
    else{
        for (int i = 0; i < m; ++i)
            cout << 0 << '␣';
        cout << '\n';
    }

    cin >> n;
    return 0;
}
```