

# Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	0	45	0	345

## Task A ()

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <set>
#include <string>

using namespace std;
int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    //ifstream cin("input.txt");
    int n;
    cin >> n;
    cout << n - 1;
    return 0;
}
```

## Task B ()

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <set>
#include <string>

using namespace std;
struct p
{
    double x, y;
};
double dist(p v1, p v2)
{
    return sqrt((v1.x - v2.x) * (v1.x - v2.x) + (v1.y - v2.y) * (v1.y - v2.y));
}
p vec(p v1, p v2)
{
    p v3;
    v3.x = v2.x - v1.x;
    v3.y = v2.y - v1.y;
    return v3;
}
int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    //ifstream cin("input.txt");
    int n;
    cin >> n;
    if (n == 6)
    {
        vector<p> col(6);
        for (int i = 0; i != n; i++)
        {
            cin >> col[i].x >> col[i].y;
        }
        vector<pair<double, int>> col1(6);
        for (int i = 0; i != 6; i++)
        {
            col1[i] = {dist(col[0], col[i]), i};
        }
        sort(col1.begin(), col1.end());
        int id1 = col1[0].second, id2 = col1[3].second, id3 = col1[4].second;
        cout.precision(10);
        cout << fixed << col[id1].x << " " << col[id1].y << "\n";
        cout << fixed << col[id2].x << " " << col[id2].y << "\n";
        cout << fixed << col[id3].x << " " << col[id3].y << "\n";
    }
    else
    {
        vector<p> col(3);
        vector<p> col1(3);
        for (int i = 0; i != n; i++)
        {
            cin >> col[i].x >> col[i].y;
        }
        for (int i = 0; i != n; i++)
        {
            p v1 = col[i];
            p v2 = col[(i + 1) % 3];
            p v3 = col[(i + 2) % 3];
            p vec1 = vec(v1, v2);
            vec1.x = ((vec1.x) / sqrt(3.0)) / 2.0;
            vec1.y = ((vec1.y) / sqrt(3.0)) / 2.0;
            p vec2;
            vec2.x = vec1.y;
            vec2.y = vec1.x;
            p vec3 = vec2;
        }
    }
}
```

```

        vec2.x = -vec2.x;
        vec3.y = -vec3.y;
        p md;
        md.x = (v1.x + v2.x) / 2.0;
        md.y = (v1.y + v2.y) / 2.0;
        p p1, p2;
        p1.x = md.x + vec2.x;
        p1.y = md.y + vec2.y;
        p2.x = md.x + vec3.x;
        p2.y = md.y + vec3.y;
        if (dist(v3, p1) < dist(v3, p2))
        {
            swap(p1, p2);
        }
        col1[i] = p1;
    }
    for (int i = 0; i != 3; i++)
    {
        cout.precision(10);
        cout << fixed << col[i].x << "␣" << col[i].y << "\n" << col1[i].x << "␣"
            << col1[i].y << "\n";
    }
}
return 0;
}

```

## Task C ()

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <set>
#include <string>

using namespace std;
int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    //ifstream cin("input.txt");
    string s;
    cin >> s;
    vector<vector<long long>> al(27);
    for (long long i = 0; i != s.size(); i++)
    {
        al[int(s[i] - 'a')].push_back(i + 1);
    }
    for (long long i = 0; i != 27; i++)
    {
        al[i].push_back(1e9);
    }
    long long n;
    cin >> n;
    long long ans = 0;
    for (long long i = 0; i != n; i++)
    {
        string ss;
        cin >> ss;
        ss.push_back('a' + 26);
        long long mx = 0;
        for (long long i = 0; i != ss.size() - 1; i++)
        {
            long long cmx = 0;
            long long pr = -1;
            long long cur = -1;
            for (int j = i; j != ss.size(); j++)
            {
                auto it = lower_bound(al[int(ss[j] - 'a')].begin(), al[int(ss[j] - 'a')].end(), pr + 1);
                if (it == al[int(ss[j] - 'a')].end())
                {
                    break;
                }
                else
                {
                    pr = *it;
                    cur++;
                }
                cmx = max(cmx, cur);
            }
            mx = max(mx, cmx);
        }
        ans += s.size() - mx;
        //cout << s.size() - mx << "\n";
    }
    cout << ans;
    return 0;
}
```

## Task D ()

## Task E ()

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>
#include <cmath>
#include <set>
#include <string>

using namespace std;
struct p
{
    long long x = 0, y = 0;
};
int main()
{
    //ifstream cin("input.txt");
    long long n, m, b;
    cin >> n >> m >> b;
    vector<p> col1(b);
    for (long long i = 0; i != b; i++)
    {
        cin >> col1[i].x >> col1[i].y;
        col1[i].x--;
        col1[i].y--;
    }
    vector<bool> col((1 << b) + (1 << (b - 1)), 1);
    long long last = -1;
    for (long long bb = b - 1; bb > -1; bb--)
    {
        long long i1 = -1, i2 = -1;
        for (long long i = 0; i != col.size(); i++)
        {
            if (col[i])
            {
                i1 = i2;
                i2 = i;
            }
            if (i1 != -1 && i2 != -1)
            {
                cout << "?_" << i1 * n + col1[bb].x << "_" << col1[bb].y << "_" <<
                    i2 * n + col1[bb].x << "_" << col1[bb].y << endl;
                long long x, y;
                cin >> x >> y;
                col[x / n] = 0;
                if (x / n == i1)
                {
                    last = i2;
                }
                else
                {
                    last = i1;
                }
                i1 = -1;
                i2 = -1;
            }
        }
    }
    cout << "!_" << last * n << "_" << 0 << endl;
    return 0;
}
```

## Task F ()