Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	20	6	0	326

Task A ()

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <cmath>
#include <deque>
#include <queue>
#include <list>

using namespace std;

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

Task B ()

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <cmath>
#include <deque>
#include <queue>
#include <list>

using namespace std;

struct p
{
        double x, y;
};

bool eq(double a, double b)
{
        return (fabs(a - b) < 0.1);
}

double len(p a)
{
        return sqrt(a.x * a.x + a.y * a.y);
}

double scal(p a, p b)
{
        return a.x * b.x + a.y * b.y;
}

double smech(p a, p b)
{
        return a.x * b.y - b.x * a.y;
}

bool check120(p a, p b, p c)
{
        p tmp1, tmp2;
        tmp1.x = a.x - b.x;
        tmp1.y = a.y - b.y;
        tmp2.x = c.x - b.x;
        tmp2.y = c.y - b.y;
        return eq(scal(tmp1, tmp2) / len(tmp1) / len(tmp2), -0.5);
}

int main()
{
        int n;
        cin >> n;
        vector<p> a(n);
        for (auto &i : a)
        {
                cin >> i.x >> i.y;
        }
        if (n == 6)
        {
                for (int i = 0; i < n; ++i)
                {
                        for (int j = 0; j < n; ++j)
                        {
                                for (int k = 0; k < n; ++k)
                                {
                                        if (i == j || i == k || k == j)
                                        {
                                                continue;
                                        }
                                        if (check120(a[i], a[j], a[k]))
                                        {
                                                cout << a[i].x << '␣' << a[i].y << '\n';
                                                cout << a[j].x << '␣' << a[j].y << '\n';
                                                cout << a[k].x << '␣' << a[k].y;
                                                return 0;
                                        }
                                }
                        }
                }
        }
        else
        {
                vector<p> tmp(6);
                for (int i = 0; i < n; ++i)
                {
                        for (int j = 0; j < n; ++j)
                        {
                                for (int k = 0; k < n; ++k)
                                {
                                        if (i == j || i == k || k == j)
                                        {
                                                continue;
                                        }
                                        if (check120(a[i], a[j], a[k]))
                                        {
                                                tmp[0] = a[i];
                                                tmp[1] = a[j];
                                                tmp[2] = a[k];
                                        }
                                }
                        }
                }
                tmp[3].x = (tmp[2].x - tmp[1].x) * 2 + tmp[0].x;
                tmp[3].y = (tmp[2].y - tmp[1].y) * 2 + tmp[0].y;
                tmp[4].x = (tmp[3].x - tmp[2].x) * 2 + tmp[1].x;
                tmp[4].y = (tmp[3].y - tmp[2].y) * 2 + tmp[1].y;
                tmp[5].x = (tmp[4].x - tmp[3].x) * 2 + tmp[2].x;
                tmp[5].y = (tmp[4].y - tmp[3].y) * 2 + tmp[2].y;
                cout << tmp[0].x << '␣' << tmp[0].y << '\n';
                cout << tmp[1].x << '␣' << tmp[1].y << '\n';
                cout << tmp[2].x << '␣' << tmp[2].y << '\n';
                cout << tmp[3].x << '␣' << tmp[3].y << '\n';
                cout << tmp[4].x << '␣' << tmp[4].y << '\n';
                cout << tmp[5].x << '␣' << tmp[5].y;
        }
        return 0;
}
```

Task C ()

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <cmath>
#include <deque>
#include <queue>
#include <list>
#include <string>

using namespace std;

int levDist(string a, string b, int l, int r)
{
        int j = l, ans = 0, sz = a.size(), lim = sz - (r - l + 1);
        for (int i = 0; i < sz && ans <= lim; ++i)
        {
                if (b[j] != a[i])
                {
                        ans++;
                }
                else
                {
                        j++;
                        if (j > r)
                        {
                                ans += sz - i - 1;
                                break;
                        }
                }
        }
        return (ans <= lim);
}

int main()
{
        string s;
        cin >> s;
        int sz = s.size();
        set<char> l;
        for (int i = 0; i < s.size(); i++)
        {
                l.insert(s[i]);
        }
        int n;
        cin >> n;
        int ans = 0;
        for (int i = 0; i < n; ++i)
        {
                string c;
                cin >> c;
                int dl = 0, dr = sz + 1, cz = c.size();
                while (dl != dr - 1)
                {
                        int d = (dl + dr) / 2;
                        bool yes = false;
                        for (int j = 0; j <= cz - d && !yes; j++)
                        {
                                bool flag = false;
                                for (int t = j; t < j + d; t++)
                                {
                                        if (l.find(c[t]) == l.end())
                                        {
                                                flag = true;
                                                break;
                                        }
                                }
                                if (flag)
                                {
                                        continue;
                                }
                                yes = levDist(s, c, j, j + d - 1);
                        }
                        if (yes)
                        {
                                dl = d;
                        }
                        else
                        {
                                dr = d;
                        }
                }
                ans += sz - dl;
        }
        cout << ans;
        return 0;
}
```

Task D ()

```cpp
#include <iostream>
#include <algorithm>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <cmath>
#include <deque>
#include <queue>
#include <list>

using namespace std;

const int INF = 1e9;

int dist(vector<vector<pair<int, int>>> &c, int ix, int iy, int jx, int jy)
{
        return abs(jx - ix - c[ix][iy].first) + abs(jy - iy - c[ix][iy].second);
}

int main()
{
        int n, m;
        cin >> n >> m;
        pair<int, int> s, f;
        cin >> s.first >> s.second >> f.first >> f.second;
        s.first--;
        s.second--;
        f.first--;
        f.second--;
        vector<vector<pair<int, int>>> c(n, vector<pair<int, int>>(m));
        for (int i = 0; i < n; ++i)
        {
                for (int j = 0; j < m; ++j)
                {
                        cin >> c[i][j].first >> c[i][j].second;
                }
        }
        vector<vector<int>> d(n * m, vector<int>(n * m));
        for (int ix = 0; ix < n; ++ix)
        {
                for (int iy = 0; iy < m; ++iy)
                {
                        for (int jx = 0; jx < n; ++jx)
                        {
                                for (int jy = 0; jy < m; ++jy)
                                {
                                        if (ix == jx && iy == jy)
                                        {
                                                continue;
                                        }
                                        d[ix * m + iy][jx * m + jy] = dist(c, ix, iy, jx, jy);
                                }
                        }
                }
        }
        for (int ix = 0; ix < n; ++ix)
        {
                for (int iy = 0; iy < m; ++iy)
                {
                        for (int jx = 0; jx < n; ++jx)
                        {
                                for (int jy = 0; jy < m; ++jy)
                                {
                                        for (int kx = 0; kx < n; ++kx)
                                        {
                                                for (int ky = 0; ky < m; ++ky)
                                                {
                                                        d[jx * m + jy][kx * m + ky] = min(d[jx * m
                                                            + jy][kx * m + ky], d[jx * m + jy][ix
                                                            * m + iy] + d[ix * m + iy][kx * m +
                                                                    ky]);
                                                        }
                                                }
                                        }
                                }
                        }
                }
                cout << d[s.first * m + s.second][f.first * m + f.second];
                return 0;
}
```

Task E ()

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <set>
#include <map>
#include <unordered_set>
#include <unordered_map>
#include <cmath>
#include <deque>
#include <queue>
#include <list>

using namespace std;

const long long s = 0;

int main()
{
        long long n, m, b;
        cin >> n >> m >> b;
        long long p, q;
        cin >> p >> q;
        cout << "?␣" << s + p - 1 << '␣' << s + q - 1 << '␣' << s + 2 * n + p - 1 << '␣' << s + q
            - 1 << '\n';
        fflush(stdout);
        long long k, r;
        cin >> k >> r;
        if (k == s + p - 1 && r == s + q - 1)
        {
                cout << "!␣" << s + 2 * n << '␣' << s << '\n';
                fflush(stdout);
        }
        else
        {
                cout << "!␣" << s << '␣' << s << '\n';
                fflush(stdout);
        }
        return 0;
}
```

Task F ()