Олимпиада СПбГУ по информатике 2018/19 учебного года

Карнаухов Кирилл Евгеньевич

A	B	C	D	E	F	Sum
100	100	100	60	100	24	484

Task A (100)

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

int main() {
        int n, m; cin >> n >> m;
        while (n < m) {
                n *= 2;
        }
        if (n == m) {
                cout << "Yes\n";
        }
        else {
                cout << "No\n";
        }
}
```

Task B (100)

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

void out() {
        cout << "Yes\n";
        exit(0);
}

int main() {
        ios::sync_with_stdio(0);
        cin.tie(0); cout.tie(0);
        int n; cin >> n;
        string s; cin >> s;
        for (int i = 0; i < n - 1; i++) {
                if (s[i] == 'o' && s[i + 1] == 'r' || s[i] == 'r' && s[i + 1] == 'o')
                        out();
        }
        for (int i = 2; i < n; i++) {
                if (s[i] == 'r' && s[i - 2] == 'o')
                        out();
        }
        for (int i = 0; i < n - 2; i++) {
                if (s[i] == 'o' && s[i + 2] == 'r')
                        out();
        }
        cout << "No\n";
}
```

Task C (100)

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

vector<vector<int>> g;
vector<int> cnt;
int dfs(int v, int p) {
        cnt[v] = 1;
        for (int to : g[v]) {
                if (to == p)
                        continue;
                cnt[v] += dfs(to, v);
        }
        return cnt[v];
}

vector<int> ans;

void get_ans(int v, int p, int up) {
        ans[v] = up;
        for (int to : g[v]) {
                if (to != p) {
                        ans[v] = max(ans[v], cnt[to]);
                        get_ans(to, v, up + cnt[v] - cnt[to]);
                }
        }
}

int main() {
        ios::sync_with_stdio(0);
        cin.tie(0); cout.tie(0);
        int n; cin >> n;
        g.resize(n);
        cnt.resize(n);
        ans.resize(n);
        for (int i = 0; i < n - 1; i++) {
                int a, b; cin >> a >> b;
                a--, b--;
                g[a].push_back(b);
                g[b].push_back(a);
        }
        dfs(0, -1);
        get_ans(0, -1, 0);
        for (auto &x : ans) {
                cout << x + 1 << "␣";
        }
        cout << endl;
        //system("pause");
}
```

Task D (60)

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

int n, p;

void split() {
        string s; cin >> s;
        if (n == 3 && p == 7) {
                cout << "a" + s.substr(0, 6) << "␣";
                cout << "b" + s.substr(3, 6) << "␣";
                cout << "c" + s.substr(0, 3) + s.substr(6, 3) << "\n";
        }
        if (n == 5 && p == 7) {
                cout << "a" + s.substr(0, 6) << "␣";
                cout << "b" + s.substr(0, 4) + s.substr(8, 1) + "a" << "␣";
                cout << "c" + s.substr(0, 2) + s.substr(6, 3) + "a" << "␣";
                cout << "d" + s.substr(2, 6) << "␣";
                cout << "e" + s.substr(4, 5) + "a" << "\n";
        }
}

void merge() {
        vector<string> s(n / 2 + 1);
        for (auto &x : s) {
                cin >> x;
        }
        string res(9, '␣');
        if (n == 3 && p == 7) {
                for (auto &x : s) {
                        if (x[0] == 'a') {
                                res[0] = x[1], res[1] = x[2], res[2] = x[3];
                                res[3] = x[4], res[4] = x[5], res[5] = x[6];
                        }
                        else if (x[0] == 'b') {
                                res[3] = x[1], res[4] = x[2], res[5] = x[3];
                                res[6] = x[4], res[7] = x[5], res[8] = x[6];
                        }
                        else {
                                res[0] = x[1], res[1] = x[2], res[2] = x[3];
                                res[6] = x[4], res[7] = x[5], res[8] = x[6];
                        }
                }
        }
        if (n == 5 && p == 7) {
                for (auto &x : s) {
                        if (x[0] == 'a') {
                                res[0] = x[1], res[1] = x[2], res[2] = x[3];
                                res[3] = x[4], res[4] = x[5], res[5] = x[6];
                        }
                        else if (x[0] == 'b') {
                                res[0] = x[1], res[1] = x[2];
                                res[2] = x[3], res[3] = x[4];
                                res[8] = x[5];
                        }
                        else if (x[0] == 'c') {
                                res[0] = x[1], res[1] = x[2];
                                res[6] = x[3], res[7] = x[4], res[8] = x[5];
                        }
                        else if (x[0] == 'd') {
                                res[2] = x[1], res[3] = x[2], res[4] = x[3];
                                res[5] = x[4], res[6] = x[5], res[7] = x[6];
                        }
                        else {
                                res[4] = x[1], res[5] = x[2], res[6] = x[3];
                                res[7] = x[4], res[8] = x[5];
                        }
                }
        }
        cout << res << endl;
}

int main() {
        string type; cin >> type;
        int t; cin >> t;
        cin >> n >> p;
        if (type == "split") {
                for (int i = 0; i < t; i++) {
                        split();
                }
        }
        else {
                for (int i = 0; i < t; i++) {
                        merge();
                }
        }
        //system("pause");
}
```

Task E (100)

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

const long long INF = 1e9;

struct point {
        long long x, y;
        int ind;
        point(long long x = 0, long long y = 0, int ind = 0) :
                x(x), y(y), ind(ind) {}
};

long long dist(point a, point b) {
        return (a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y);
}

int main() {
        ios::sync_with_stdio(0);
        cin.tie(0); cout.tie(0);
        int n; cin >> n;
        if (n == 1) {
                cout << 1 << endl;
                return 0;
        }
        vector<point> all(n);
        for (int i = 0; i < n; i++) {
                cin >> all[i].x >> all[i].y;
                all[i].ind = i;
        }
        point p, q;
        long long step_x = q.x - p.x;
        long long step_y = q.y - p.y;
        cin >> p.x >> p.y;
        cin >> q.x >> q.y;
        sort(all.begin(), all.end(), [&](point &a, point &b) {
                long long fir = dist(p, a) - dist(p, b);
                long long sec = dist(q, a) - dist(q, b);
                if (sec != fir)
                        return sec < fir;
                return sec < 0;
        });
        point a = all[0];
        point b = all[1];
        long long fir = dist(p, a) - dist(p, b);
        long long sec = dist(q, a) - dist(q, b);
        if (fir == sec && sec == 0) {
                cout << -1 << endl;
        }
        else {
                cout << all[0].ind + 1 << endl;
        }
        //system("pause");
}
```

Task F (24)

```cpp
#include <iostream>
#include <algorithm>
#include <vector>
#include <map>
using namespace std;
const long long INF = 1e16;

int main() {
        ios::sync_with_stdio(0);
        cin.tie(0); cout.tie(0);
        int n, k; cin >> n >> k;
        map<pair<int, int>, int> d;
        map<pair<int, int>, int> now;;
        d[{0, 0}] = 0;
        int add = 0;
        for (int i = 0; i < n; i++) {
                int x, y;
                cin >> x >> y;
                int tmp = k;
                if (x >= k) {
                        x -= k;
                        add += k;
                        k = 0;
                }
                else if (y >= k) {
                        y -= k;
                        add += k;
                        k = 0;
                }
                int num = 0;
                int size = d.size();
                for (auto &p : d) {
                        num++;
                        if (num > 10 && num < size / 40)
                                continue;
                        int q = p.first.first;
                        int w = p.first.second;
                        int cost = p.second;
                        now[{max(0, q + k - x), max(0, w - y)}] = max(now[{max(0, q + k - x), max(0, w - y)}], cost + min(x, q + k) + min(w, y));
                        now[{max(0, q - x), max(0, w + k - y)}] = max(now[{max(0, q - x), max(0, w + k - y)}], cost + min(y, w + k) + min(q, x));
                }
                k = tmp;
                d = now;
                now.clear();
        }
        int ans = 0;
        for (auto &p : d) {
                ans = max(ans, p.second);
        }
        cout << ans + add << endl;
        //system("pause");
}
```