

Олимпиада СПбГУ по информатике 2018/19 учебного года

Шарафутдинов Дан Русланович

A	B	C	D	E	F	Sum
100	100	100	60	12	0	372

Task A (100)

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cstdio>

using namespace std;

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

    if (m % n != 0) {
        cout << "No";
        return 0;
    }

    int d = m / n;

    int a = 1;
    while (true) {
        if (a >= d) {
            if (a == d) {
                cout << "Yes";
                return 0;
            }

            cout << "No";
            return 0;
        }

        a <<= 1;
    }

    return 0;
}
```

Task B (100)

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cstdio>

#include <string>

using namespace std;

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

    string s;
    cin >> s;

    for (int i = 0; i < n - 1; ++i) {
        if (s[i] == 'o' && s[i + 1] == 'r') {
            cout << "Yes";
            return 0;
        }

        if (s[i] == 'r' && s[i + 1] == 'o') {
            cout << "Yes";
            return 0;
        }
    }

    for (int i = 0; i < n - 2; ++i) {
        if (s[i] == 'o' && s[i + 2] == 'r') {
            cout << "Yes";
            return 0;
        }
    }

    cout << "No";
    return 0;
}
```

Task C (100)

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cstdio>

#include <vector>
#include <set>
#include <queue>
#include <algorithm>

using namespace std;

vector <vector <int>> > g;
vector <int> d;

int fr(int v) {
    if (g[v].size() == 0) {
        d[v] = 0;
    }
    else {
        int s = 0;
        for (int i = 0; i < g[v].size(); ++i) {
            int u = g[v][i];
            s += fr(u);
        }
        d[v] = s + g[v].size();
    }
    return d[v];
}

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

    vector <pair <int, int>> > a(n - 1);

    bool fl = true;
    for (int i = 0; i < n - 1; ++i) {
        int u, v;
        cin >> u >> v;

        if (v != u + 1) {
            fl = false;
        }

        a[i] = { u - 1, v - 1 };
    }

    if (fl) {
        for (int i = 0; i < n / 2; ++i) {
            cout << n - i << "┐";
        }

        for (int i = n / 2; i < n; ++i) {
            cout << i + 1 << "┐";
        }

        return 0;
    }

    g.resize(n, vector <int>(0));

    set <int> q;
    q.insert(0);

    while (q.size() != n) {
        for (int i = 0; i < n - 1; ++i) {
```

```

        int u = a[i].first, v = a[i].second;
        if (q.find(u) != q.end() && q.find(v) == q.end()) {
            q.insert(v);
            g[u].push_back(v);
        }

        if (q.find(v) != q.end() && q.find(u) == q.end()) {
            q.insert(u);
            g[v].push_back(u);
        }
    }

    d.resize(n, -1);
    fr(0);

    vector<int> kj(n, -1);
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < g[i].size(); ++j) {
            kj[i] = max(kj[i], d[g[i][j]] + 1);
        }

        kj[i] = max(kj[i], d[0] - d[i]);
    }

    for (int i = 0; i < n; ++i) {
        cout << kj[i] + 1 << '␣';
    }

    return 0;
}

```

Task D (60)

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <cstdio>

#include <string>

using namespace std;

string rng(string x, int a, int b) {
    string y = "";
    for (int i = a; i < b; ++i) {
        y += x[i];
    }

    return y;
}

string decab(string a, string b) {
    if (a[0] == 'a' && b[0] == 'b') {
        return rng(a, 1, 4) + rng(b, 1, 7);
    }

    if (a[0] == 'b' && b[0] == 'a') {
        return rng(b, 1, 4) + rng(a, 1, 7);
    }

    if (a[0] == 'a' && b[0] == 'c') {
        return rng(a, 1, 7) + rng(b, 4, 7);
    }

    if (a[0] == 'c' && b[0] == 'a') {
        return rng(b, 1, 7) + rng(a, 4, 7);
    }

    if (a[0] == 'b' && b[0] == 'c') {
        return rng(b, 1, 4) + rng(a, 1, 7);
    }

    if (a[0] == 'c' && b[0] == 'b') {
        return rng(a, 1, 4) + rng(b, 1, 7);
    }
}

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

    int t, n, p;
    cin >> t >> n >> p;

    if (n == 3 && p == 7) {
        if (x == "split") {
            for (int i = 0; i < t; ++i) {
                string y;
                cin >> y;

                cout << "a" + rng(y, 0, 6) + "␣";
                cout << "b" + rng(y, 3, 9) + "␣";
                cout << "c" + rng(y, 0, 3) + rng(y, 6, 9) + "\n";
            }
        }
        else if (x == "merge") {
            for (int i = 0; i < t; ++i) {
                string a, b;
                cin >> a >> b;

                cout << decab(a, b) << "\n";
            }
        }
    }
}
```

```

    }
}
else if (n == 5 && p == 7) {
    if (x == "split") {
        for (int i = 0; i < t; ++i) {
            string y;
            cin >> y;

            cout << "a" + rng(y, 0, 6) + "␣";
            cout << "a" + rng(y, 0, 6) + "␣";
            cout << "b" + rng(y, 3, 9) + "␣";
            cout << "b" + rng(y, 3, 9) + "␣";
            cout << "c" + rng(y, 0, 3) + rng(y, 6, 9) + "\n";
        }
    }
    else if (x == "merge") {
        for (int i = 0; i < t; ++i) {
            string a, b, c;
            cin >> a >> b >> c;

            if (a != b) {
                cout << decab(a, b) << "\n";
            }
            else if (a != c) {
                cout << decab(a, c) << "\n";
            }
            else if (b != c) {
                cout << decab(b, c) << '\n';
            }
        }
    }
}

return 0;
}

```

Task E (12)

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <stdio>
#include <cmath>

using namespace std;

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

    if (n == 2) {
        int ax, ay, bx, by;
        cin >> ax >> ay;
        cin >> bx >> by;

        int px, py, qx, qy;
        cin >> px >> py >> qx >> qy;

        if (ax > bx) {
            cout << 1;
        }
        else if (bx > ax) {
            cout << 2;
        }
        else {
            double da = sqrt((ax - px) * (ax - px) +
                             (ay - py) * (ay - py));
            double db = sqrt((bx - px) * (bx - px) +
                             (by - py) * (by - py));

            if (da < db) {
                cout << 1;
            }
            else if (db < da) {
                cout << 2;
            }
            else {
                cout << -1;
            }
        }
    }
    else {
        cout << -1;
    }

    return 0;
}
```

Task F (—)