

Олимпиада СПбГУ по информатике 2018/19 учебного года

Батыргариев Александр Ильдарович

A	B	C	D	E	F	Sum
100	100	100	60	44	0	404

Task A (100)

```
#include <bits/stdc++.h>

using namespace std;

#define int long long
#define double long double

const int INF = 1e18;
const double Dx = 1e-9;

int32_t main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    int n, m;
    cin >> n >> m;
    while (n < m) {
        n *= 2;
    }
    if (n == m) {
        cout << "Yes";
    } else {
        cout << "No";
    }
    return 0;
}
```

Task B (100)

```
#include <bits/stdc++.h>

using namespace std;

#define int long long
#define double long double

const int INF = 1e18;
const double Dx = 1e-9;

int32_t main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    int n;
    cin >> n;
    string s;
    cin >> s;
    auto check = [&](int l, int r) -> bool {
        for (int i = max((int) 0, l); i < min(n, r); i++) {
            if (s[i] == 'o' && s[i + 1] == 'r') {
                return true;
            }
        }
        return false;
    };
    if (check(0, n)) {
        cout << "Yes";
        return 0;
    }
    for (int i = 0; i < n - 1; i++) {
        swap(s[i], s[i + 1]);
        if (check(i - 3, i + 3)) {
            cout << "Yes";
            return 0;
        }
        swap(s[i], s[i + 1]);
    }
    cout << "No";
    return 0;
}
```

Task C (100)

```
#include <bits/stdc++.h>

using namespace std;

#define int long long
#define double long double

int timer = 0;
const int INF = 1e18;
const double Dx = 1e-9;

vector<int> was, sz, ans, tin, tout;
vector<vector<int>>> edges;

void dfs1(int i) {
    was[i] = true;
    tin[i] = timer++;
    for (int to : edges[i]) {
        if (!was[to]) {
            dfs1(to);
        }
    }
    tout[i] = timer++;
}

void change(int f, int t) {
    sz[f] -= sz[t] + 1;
    sz[t] += sz[f] + 1;
}

int calc(int i) {
    int res = 0;
    for (int to : edges[i]) {
        res = max(res, sz[to] + 1);
    }
    return res + 1;
}

void dfs2(int i) {
    was[i] = true;
    ans[i] = calc(i);
    for (int to : edges[i]) {
        if (!was[to]) {
            change(i, to);
            dfs2(to);
            change(to, i);
        }
    }
}

int32_t main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    int n;
    cin >> n;
    was.resize(n);
    edges.resize(n);
    int x, y;
    for (int i = 0; i < n - 1; i++) {
        cin >> x >> y;
        x--;
        y--;
        edges[x].push_back(y);
        edges[y].push_back(x);
    }
    tin.resize(n);
    tout.resize(n);
}
```

```

    sz.resize(n);
    was.resize(n);
    ans.resize(n);
    dfs1(0);
    for (int i = 0; i < n; i++) {
        sz[i] = (tout[i] - tin[i]) / 2;
    }
    was.assign(n, false);
    dfs2(0);
    for (int x : ans) {
        cout << x << '┐';
    }
    return 0;
}

```

Task D (60)

```
#include <bits/stdc++.h>

using namespace std;

string slc(string &s, int i, int j) {
    string res = "";
    int t = i % 9;
    while (i < j) {
        res += s[t];
        i++;
        t = (t + 1) % 9;
    }
    return res;
}

void gen(string &s, vector<string> &res) {
    int n = 3, p = 7;
    res.resize(n);
    for (int i = 0; i < n; i++) {
        char t = ('a' + i);
        res[i] = slc(s, 3 * i, 3 * i + p - 1) + t;
    }
}

void gen01(string &s, vector<string> &res) {
    int n = 5, p = 7;
    res.resize(n);
    for (int i = 0; i < n; i++) {
        char t = ('a' + i);
        res[i] = slc(s, 3 * i, 3 * i + p - 1) + t;
    }
}

void gen02(string &s, vector<string> &res) {
    int n = 7, p = 4;
    res.resize(n);
    for (int i = 0; i < n; i++) {
        char t = ('a' + i);
        res[i] = slc(s, 2 * i, 2 * i + p - 1) + t;
    }
}

string gen2(vector<string> &res) {
    int n = 3, p = 7;
    int p1 = res[0][6] - 'a';
    string ans = "xxxxxxxxx";
    int i = 0;
    for (int t = 3 * p1; t < 3 * p1 + 6; t++) {
        ans[t % 9] = res[0][i];
        i++;
    }
    int p2 = res[1][6] - 'a';
    int j = 0;
    for (int t = 3 * p2; t < 3 * p2 + 6; t++) {
        ans[t % 9] = res[1][j];
        j++;
    }
    return ans;
}

string gen3(vector<string> res) {
    int n = 5, p = 7;
    int p1 = res[0][6] - 'a';
    string ans = "xxxxxxxxx";
    int i = 0;
    for (int t = 3 * p1; t < 3 * p1 + 6; t++) {
        ans[t % 9] = res[0][i];
    }
}
```

```

        i++;
    }
    int p2 = res[1][6] - 'a';
    int j = 0;
    for (int t = 3 * p2; t < 3 * p2 + 6; t++) {
        ans[t % 9] = res[1][j];
        j++;
    }
    int p3 = res[2][6] - 'a';
    int k = 0;
    for (int t = 3 * p3; t < 3 * p3 + 6; t++) {
        ans[t % 9] = res[2][k];
        k++;
    }
    return ans;
}

```

```

string gen4(vector<string> res) {
    int n = 7, p = 4;
    int p1 = res[0][6] - 'a';
    string ans = "xxxxxxxx";
    int i = 0;
    for (int t = 2 * p1; t < 2 * p1 + 4; t++) {
        ans[t % 9] = res[0][i];
        i++;
    }
    int p2 = res[1][6] - 'a';
    int j = 0;
    for (int t = 2 * p2; t < 2 * p2 + 4; t++) {
        ans[t % 9] = res[1][j];
        j++;
    }
    int p3 = res[2][6] - 'a';
    int k = 0;
    for (int t = 2 * p3; t < 2 * p3 + 4; t++) {
        ans[t % 9] = res[2][k];
        k++;
    }
    int p4 = res[3][6] - 'a';
    int l = 0;
    for (int t = 2 * p4; t < 2 * p4 + 4; t++) {
        ans[t % 9] = res[3][l];
        l++;
    }
    return ans;
}

```

```

int32_t main() {
    string type;
    cin >> type;
    int t, n, p;
    cin >> t >> n >> p;
    string s;
    if (type == "split") {
        for (int i = 0; i < t; i++) {
            cin >> s;
            vector<string> res;
            if (n == 3) {
                gen(s, res);
            }
            if (n == 5) {
                gen01(s, res);
            }
            if (n == 7) {
                gen02(s, res);
            }
            for (auto st : res) {
                cout << st << '␣';
            }
            cout << '\n';
        }
    }
    else {

```

```

vector<string> res;
int n2 = n / 2 + n % 2;
for (int i = 0; i < t; i++) {
    vector<string> res(n2);
    for (string &st : res) {
        cin >> st;
    }
    if (n == 3) {
        cout << gen2(res) << '\n';
    }
    if (n == 5) {
        cout << gen3(res) << '\n';
    }
    if (n == 7) {
        cout << gen4(res) << '\n';
    }
}
return 0;
}

```

Task E (44)

```
#include <bits/stdc++.h>

using namespace std;

#define int long long
#define double long double

struct dot {
    int x, y;
};

int distance(const dot &a, const dot &b) {
    return (a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y);
}

int32_t main() {
    int n;
    cin >> n;
    vector<dot> dots(n);
    for (dot &d : dots) {
        cin >> d.x >> d.y;
    }
    int xp, yp;
    int xq, yq;
    cin >> xp >> yp >> xq >> yq;

    int dx = xq - xp;
    int dy = yq - yp;

    if (dy == 0 && n != 2) {
        int b = 0;
        for (int i = 1; i < n; i++) {
            if (dots[i].x > dots[b].x || (dots[i].x == dots[b].x && abs(dots[i].y) < abs(dots[b].y)))
                b = i;
        }
        cout << b + 1;
    }
    else {
        int m = min(((int) 1e8 - xp) / (abs(dx) + 1), ((int) 1e8 - yp) / (abs(dy) + 1)) + 1;
        int x = xp + m * dx;
        int y = yp + m * dy;
        int d = 1e18;
        int t = -1;
        bool ok = true;
        for (int i = 0; i < n; i++) {
            if (distance(dots[i], {x, y}) < d) {
                d = distance(dots[i], {x, y});
                t = i;
            }
        }
        int calc = 0;
        for (int i = 0; i < n; i++) {
            if (distance(dots[i], {x, y}) == d) {
                calc++;
            }
        }
        if (calc == 1) {
            cout << t + 1;
        }
        else {
            cout << -1;
        }
    }
    return 0;
}
```


Task F (—)