

Олимпиада СПбГУ по информатике 2018/19 учебного года

Шеин Андрей Иванович

A	B	C	D	E	F	Sum
100	100	60	0	100	7	367

Task A (100)

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie();
    cout.tie();

    int n, m;
    cin >> n >> m;

    while (n < m) {
        n *= 2;
    }

    if (n == m) {
        cout << "Yes" << endl;
    }
    else {
        cout << "No" << endl;
    }

    return 0;
}
```

Task B (100)

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

using namespace std;

string s;

bool test(int i, int j) {
    return s[i] == 'o' && s[j] == 'r';
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie();
    cout.tie();

    int n;
    cin >> n;

    cin >> s;

    for (int i = 1; i < n; i++) {
        if (test(i, i - 1) || test(i - 1, i)) {
            cout << "Yes" << endl;
            return 0;
        }
    }

    for (int i = 0; i < n - 2; i++) {
        if (test(i, i + 2)) {
            cout << "Yes" << endl;
            return 0;
        }
    }

    cout << "No" << endl;

    return 0;
}
```

Task C (60)

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

using namespace std;

int n;

vector <vector <int>> simple;
vector <int> tree;
vector <bool> used;
vector <int> cnt;
vector <int> top_sort;

void dfs(int v) {
    used[v] = true;
    for (auto e : simple[v]) {
        if (!used[e]) {
            tree[e] = v;
            used[e] = true;
            dfs(e);
        }
    }
    top_sort.push_back(v);
}

void update() {
    for (int i = 0; i < n; i++) {
        used[i] = false;
        cnt[i] = 1;
    }
    top_sort.clear();
}

void sum() {
    for (auto i : top_sort) {
        cnt[tree[i]] += cnt[i];
    }
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie();
    cout.tie();

    cin >> n;

    if (n > 1000) {
        for (int i = n; i > n / 2; i--) {
            cout << i << '␣';
        }
        for (int i = n / 2 + 1; i <= n; i++) {
            cout << i << '␣';
        }
        cout << endl;
        return 0;
    }

    simple.resize(n);
    tree.resize(n);
    used.resize(n);
    cnt.resize(n);

    for (int i = 0; i < n - 1; i++) {
        int a, b;
        cin >> a >> b;
        simple[a - 1].push_back(b - 1);
    }
}
```

```

        simple[b - 1].push_back(a - 1);
    }

    for (int i = 0; i < n; i++) {
        update();
        tree[i] = -1;
        dfs(i);
        top_sort.pop_back();
        sum();
        int m = 0;
        for (auto e : simple[i]) {
            m = max(m, cnt[e]);
        }
        cout << m + 1 << '␣';
    }
    cout << endl;

    return 0;
}

```

Task D (0)

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

using namespace std;

int mod = 1000 * 1000 * 1000 + 7;
int p = 31;

int num(string s) {
    int res = s[0] - 'a';
    for (int i = 1; i < (int)s.size(); i++) {
        res = (1LL * res * p + (s[i] - 'a')) % mod;
    }
    return res;
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie();
    cout.tie();

    string type;
    cin >> type;

    char c = 'r';
    if (type == "split") {
        int t, n, p;
        cin >> t >> n >> p;
        for (int i = 0; i < t; i++) {
            string pas;
            cin >> pas;
            string one = pas.substr(0, 3);
            string two = pas.substr(3, 3);
            string three = pas.substr(6, 3);
            string first = one + two + c;
            string second = two + c + three;
            string third = c + one + three;
            cout << first << '\n' << second << '\n' << third << endl;
        }
    }
    else {
        int t, n, p;
        cin >> t >> n >> p;
        for (int i = 0; i < t; i++) {
            string first, second;
            cin >> first >> second;
            if (first[6] == c && second[3] == c && first.substr(3, 3) == second.substr(0, 3)) {
                cout << first.substr(0, 6) + second.substr(4, 3) << endl;
            }
            else if (first[3] == c && second[6] == c && second.substr(3, 3) == first.substr(0, 3)) {
                cout << second.substr(0, 6) + first.substr(4, 3) << endl;
            }
            else if (first[6] == c && second[0] == c && first.substr(0, 3) == second.substr(1, 3)) {
                cout << first.substr(0, 6) + second.substr(4, 3) << endl;
            }
            else if (first[0] == c && second[6] == c && second.substr(0, 3) == first.substr(1, 3)) {
                cout << second.substr(0, 6) + first.substr(4, 3) << endl;
            }
            else if (first[3] == c && second[0] == c && first.substr(4, 3) == second.substr(4, 3)) {
                cout << first.substr(0, 3) + second.substr(1, 6) << endl;
            }
        }
    }
}
```

```

        else if (first[0] == c && second[3] == c && first.substr(4, 3) == second.
            substr(4, 3)) {
            cout << second.substr(0, 3) + first.substr(1, 6) << endl;
        }
    }
    return 0;
}

```

Task E (100)

```
def dist(a, b, c, d):  
    return (a - c) ** 2 + (b - d) ** 2  
  
n = int(input())  
x = [0] * n  
y = [0] * n  
for i in range(n):  
    x[i], y[i] = map(int, input().split())  
a, b = map(int, input().split())  
c, d = map(int, input().split())  
t = c + (c - a) * (10 ** 20)  
q = d + (d - b) * (10 ** 20)  
m = 10 ** 100  
j = -1  
cnt = 0  
for i in range(n):  
    cur = dist(x[i], y[i], t, q)  
    if cur < m:  
        m = cur  
        j = i + 1  
        cnt = 1  
    elif cur == m:  
        cnt += 1  
if cnt > 1:  
    print(-1)  
else:  
    print(j)
```

Task F (7)

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

int n, k;
vector<int> r;
vector<int> b;
int ans = 0;

int cnt(vector<bool> a) {
    int res = 0;
    int red = 0;
    int blue = 0;
    for (int i = 0; i < n; i++) {
        if (a[i]) {
            blue += k;
        }
        else {
            red += k;
        }
        res += min(red, r[i]) + min(blue, b[i]);
        red = red - min(red, r[i]);
        blue = blue - min(blue, b[i]);
    }
    return res;
}

void rec(int left, vector<bool> a) {
    if (left == 0) {
        ans = max(ans, cnt(a));
        return;
    }
    a.push_back(false);
    rec(left - 1, a);
    a.pop_back();
    a.push_back(true);
    rec(left - 1, a);
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie();
    cout.tie();

    cin >> n >> k;

    r.resize(n);
    b.resize(n);

    for (int i = 0; i < n; i++) {
        cin >> r[i] >> b[i];
    }

    rec(n, vector<bool>());

    cout << ans << endl;

    return 0;
}
```