Олимпиада СПбГУ по информатике 2018/19 учебного года

Купчик Антон Михайлович

A	B	C	D	E	F	Sum
100	100	100	0	35	72	407

Task A (100)

```cpp
#define _CRT_SECURE_NO_WARNINGS
#include <algorithm>
#include <iostream>
#include <stdio.h>
#include <iomanip>
#include <cstdio>
#include <string>
#include <vector>
#include <random>
#include <stack>
#include <queue>
#include <deque>
#include <cmath>
#include <ctime>
#include <map>
#include <set>

using namespace std;

const long double PI = 3.14159265358979323;
const long long mx64 = 9223372036854775807;
const int mx32 = 2147483647;
const short int mx16 = 32767;
const long long mod_1 = 1000000007;
const long long mod_2 = 998244353;

int main()
{
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        cout.precision(15);

        int n, m;
        cin >> n >> m;
        while (true)
        {
                if (n == m) {
                        cout << "Yes";
                        return 0;
                }
                else if (n > m) {
                        cout << "No";
                        return 0;
                }
                else {
                        n *= 2;
                }
        }

        //cout << "\n"; system("pause");
        return 0;
}
```

Task B (100)

```
#define _CRT_SECURE_NO_WARNINGS
#include <algorithm>
#include <iostream>
#include <stdio.h>
#include <iomanip>
#include <cstdio>
#include <string>
#include <vector>
#include <random>
#include <stack>
#include <queue>
#include <deque>
#include <cmath>
#include <ctime>
#include <map>
#include <set>

using namespace std;

const long double PI = 3.14159265358979323;
const long long mx64 = 9223372036854775807;
const int mx32 = 2147483647;
const short int mx16 = 32767;
const long long mod_1 = 1000000007;
const long long mod_2 = 998244353;

int main()
{
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        cout.precision(15);

        int n; cin >> n;
        string s;
        cin >> s;

        s = '-' + s + '-';

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

        for (int i = 1; i < s.size() - 2; i++)
        {
                if ((s[i - 1] == 'o' && s[i + 1] == 'r') || (s[i] == 'o' && s[i+2] == 'r') || (s[i
                    ] == 'r' && s[i+1] == 'o')) {
                        cout << "Yes";
                        return 0;
                }
        }

        cout << "No";
        //cout << "\n"; system("pause");
        return 0;
}
/*

*/
```

Task C (100)

```
#define _CRT_SECURE_NO_WARNINGS
#include <algorithm>
#include <iostream>
#include <stdio.h>
#include <iomanip>
#include <cstdio>
#include <string>
#include <vector>
#include <random>
#include <stack>
#include <queue>
#include <deque>
#include <cmath>
#include <ctime>
#include <map>
#include <set>

using namespace std;

const long double PI = 3.14159265358979323;
const long long mx64 = 9223372036854775807;
const int mx32 = 2147483647;
const short int mx16 = 32767;
const long long mod_1 = 1000000007;
const long long mod_2 = 998244353;

int n;
vector<vector<int>> g;
vector<vector<int>> gr;
vector<int> sz;

void dfs(int v)
{
        for (int to : g[v]) {
                dfs(to);
        }
        sz[v]++;
        for (int to : g[v]) {
                sz[v] += sz[to];
        }
}

int CoutMax(int v)
{
        int ret = 0;
        for (int to : g[v]) {
                ret = max(ret, sz[to]);
        }
        return ret;
}

int main()
{
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        cout.precision(15);

        cin >> n;
        g.resize(n);
        gr.resize(n);
        sz.resize(n, 0);
        for (int i = 0; i < n - 1; i++) {
                int u, v;
                cin >> u >> v;
                u--; v--;
                gr[u].push_back(v);
                gr[v].push_back(u);
        }
        vector<bool> usd(n, false);
        queue<int> bfs; bfs.push(0);
        usd[0] = true;
        while (bfs.size())
        {
                int now = bfs.front();
                bfs.pop();

                for (int to : gr[now]) {
                        if (!usd[to]) {
                                g[now].push_back(to);
                                bfs.push(to);
                                usd[to] = true;
                        }
                }
        }

        dfs(0);

        for (int i = 0; i < n; i++) {
                int ret = CoutMax(i);
                int ret2 = n - sz[i];
                cout << (max(ret, ret2) + 1) << "␣";
        }

        //cout << "\n"; system("pause");
        return 0;
}
/*

*/
```

Task D (—)

Task E (35)

```python
n = int(input())
psX = []
psY = []
for i in range(n):
    sins = list(map(int, input().split()))
    psX.append(sins[0])
    psY.append(sins[1])

_ = input()
_ = input()

mx = 2000000000000000000
mx = mx*mx

s = []

for i in range(n):
    s.append(((mx - psX[i]) * (mx - psX[i]) + psY[i] * psY[i], i))

s = sorted(s, key=lambda x: x[0])

if (n == 1):
    print('1')
elif (s[0][0] == s[1][0]):
    print('-1')
else:
    print(s[0][1]+1)
```

Task F (72)

```
#define _CRT_SECURE_NO_WARNINGS
#include <algorithm>
#include <iostream>
#include <stdio.h>
#include <iomanip>
#include <cstdio>
#include <string>
#include <vector>
#include <random>
#include <stack>
#include <queue>
#include <deque>
#include <cmath>
#include <ctime>
#include <map>
#include <set>

using namespace std;

const long double PI = 3.14159265358979323;
const long long mx64 = 9223372036854775807;
const int mx32 = 2147483647;
const short int mx16 = 32767;
const long long mod_1 = 1000000007;
const long long mod_2 = 998244353;

int main()
{
        ios_base::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        cout.precision(15);

        long long n, k, ff, ss;
        cin >> n >> k;
        vector<vector<long long>> rr(2, vector<long long>(n + 1, 0)), rb(2, vector<long long>(n +
            1, 0)), dp(2, vector<long long>(n + 1, 0));
        vector<long long> R(n + 1), B(n + 1), cnt(4, 0);

        for (int i = 1; i <= n; i++) {
                cin >> R[i] >> B[i];
        }

        dp[0][1] = min(k, R[1]);
        dp[1][1] = min(k, B[1]);

        rr[0][1] = max((long long)0, k - R[1]);
        rb[1][1] = max((long long)0, k - B[1]);

        for (int i = 2; i <= n; i++)
        {
                cnt[0] = rr[0][i - 1];
                cnt[1] = rb[0][i - 1];
                cnt[0] += k;
                ff = min(cnt[0], R[i]) + min(cnt[1], B[i]);

                cnt[2] = rr[1][i - 1];
                cnt[3] = rb[1][i - 1];
                cnt[2] += k;
                ss = min(cnt[2], R[i]) + min(cnt[3], B[i]);

                if (ff + dp[0][i - 1] >= ss + dp[1][i - 1])
                {
                        dp[0][i] = dp[0][i - 1];
                        rr[0][i] = max((long long)0, cnt[0] - R[i]);
                        rb[0][i] = max((long long)0, cnt[1] - B[i]);
                        dp[0][i] += ff;
                }
                else
                {
                        dp[0][i] = dp[1][i - 1];
                        rr[0][i] = max((long long)0, cnt[2] - R[i]);
                        rb[0][i] = max((long long)0, cnt[3] - B[i]);
                        dp[0][i] += ss;
                }

                cnt[0] = rr[0][i - 1];
                cnt[1] = rb[0][i - 1];
                cnt[1] += k;
                ff = min(cnt[0], R[i]) + min(cnt[1], B[i]);

                cnt[2] = rr[1][i - 1];
                cnt[3] = rb[1][i - 1];
                cnt[3] += k;
                ss = min(cnt[2], R[i]) + min(cnt[3], B[i]);

                if (ff + dp[0][i - 1] >= ss + dp[1][i - 1])
                {
                        dp[1][i] = dp[0][i - 1];
                        rr[1][i] = max((long long)0, cnt[0] - R[i]);
                        rb[1][i] = max((long long)0, cnt[1] - B[i]);
                        dp[1][i] += ff;
                }
                else
                {
                        dp[1][i] = dp[1][i - 1];
                        rr[1][i] = max((long long)0, cnt[2] - R[i]);
                        rb[1][i] = max((long long)0, cnt[3] - B[i]);
                        dp[1][i] += ss;
                }

        }

        cout << max(max((long long)0, dp[0][n]), dp[1][n]);
        //system("pause");
        return 0;
}
/*

*/
```