Олимпиада СПбГУ по информатике 2018/19 учебного года

Ястреба Елизавета Сергеевна

A	B	C	D	E	F	Sum
100	100	100	0	44	0	344

Task A (100)

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

int main()
{
#ifdef _DEBUG
	freopen("input.txt", "r", stdin);
	freopen("output.txt", "w", stdout);
#endif // _DEBUG

	int a, b;
	cin >> a >> b;
	if (a > b)
		cout << "No";
	else {
		int p = 1;
		for (int i = 1; i <= 10; i++) {

			//cout << a << " " << b << endl;
			if (a* p == b) {
				cout << "Yes";
				return 0;
			}
			p = p * 2;
		}
		cout << "No";
	}

	return 0;
}
```

Task B (100)

```cpp
#include <iostream>
#include <algorithm>
#include <math.h>
#include <vector>
#include <string>
using namespace std;

int main()
{
#ifdef _DEBUG
	freopen("input.txt", "r", stdin);
	freopen("output.txt", "w", stdout);
#endif // _DEBUG
	int n;
	cin >> n;
	string s;
	cin >> s;
	if (s.size() == 1)
	{
		cout << "No";
		return 0;
	}
	for (int i = 0; i < s.size() - 1; i++) {
		char a = s[i];
		char b = s[i + 1];
		if (b == 'r' && a == 'o') {
			cout << "Yes";
			return 0;
		}

		if (b == 'o' && a == 'r') {
			cout << "Yes";
			return 0;
		}

		if (i != 0) {
			char c = s[i - 1];
			swap(a, b);
			if (c == 'o' && a == 'r') {
				cout << "Yes";
				return 0;
			}
		}
	}
	cout << "No";
	return 0;
}
```

Task C (100)

```cpp
#include <iostream>
#include <algorithm>
#include <math.h>
#include <vector>
#include <string>
#include <queue>
using namespace std;

void dfs(int v, vector<vector<int>>&A, vector<bool>&used, vector<int>& ans) {
	used[v] = 1;
	ans[v] = 0;
	for (int i = 0; i < A[v].size(); i++) {
		int to = A[v][i];
		if (!used[to]) {
			dfs(to, A, used, ans);
		}
	}
	for (int i = 0; i < A[v].size(); i++) {
		ans[v] += ans[A[v][i]];
	}
	ans[v]++;
}


void dfs1(int v, vector<vector<int>>&A, vector<bool>&used1, vector<int>& ans, vector<int>&len) {
	used1[v] = 1;
	int m = 0;

	for (int j = 0; j < A[v].size(); j++) {
		int to = A[v][j];
		m = max(m, ans[to]);
	}

	m = max(m, ans[v]);
	int cnt = 0;
	for (int j = 0; j < A[v].size(); j++) {
		int to = A[v][j];
		if (ans[to] != m)
			cnt += ans[to];
	}
	len[v] = max(m - cnt, cnt);

	for (int i = 0; i < A[v].size(); i++) {
		int to = A[v][i];
		if (!used1[to]) {
			dfs(to, A, used1, ans);
		}
	}

}

int main()
{
#ifdef _DEBUG
	freopen("input.txt", "r", stdin);
	freopen("output.txt", "w", stdout);
#endif // _DEBUG
	int n;
	cin >> n;
	vector<vector<int>>A(n);
	for (int i = 0; i < n - 1; i++) {
		int a, b;
		cin >> a >> b;
		a--;
		b--;
		A[a].push_back(b);
		A[b].push_back(a);
	}
	vector<int>ans(n, 0);
```

```
        vector<int>len(n, 0);
        vector<bool>used(n, 0);
        vector<bool>used1(n, 0);
        dfs(0, A, used, ans);
        //dfs1(0, A, used1, ans, len);
        for (int i = 0; i < n; i++) {
                int m = n;

                int cnt = 0, k = 0;
                for (int j = 0; j < A[i].size(); j++) {
                        int to = A[i][j];
                        if (ans[to] < ans[i])
                        {
                                cnt += ans[to];
                                k = max(k, ans[to]);
                        }
                }
                //cout << cnt << " " << m << " ";
                cout << max(m - cnt, k + 1) << "␣";
        }


        return 0;
}
```

Task D (0)

Task E (44)

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

#define double long double
struct Point {
        double x;
        double y;

        Point() {}

        Point(double _x, double _y) {
                x = _x;
                y = _y;

        }

        void scan() {
                cin >> x >> y;
        }
};


double len(Point a, Point b) {
        return sqrt((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y));
}


vector<pair<double, int>> ans_f(Point m, vector<Point>&A) {
        vector<pair<double, int>>ans;
        for (int i = 0; i < A.size(); i++) {
                ans.push_back({ len(A[i], m), i });
        }

        return ans;
}

bool cmp(pair<double, int> a, pair<double, int> b) {
        return a.first < b.first;
}

int main()
{
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif // _DEBUG
        int n;
        double Max = -1e9;
        bool f = 0;
        cin >> n;
        if (n == 1) {
                cout << 1;
                return 0;
        }
        vector<Point>A(n);
        for (int i = 0; i < n; i++) {

                A[i].scan();
                if (!(abs(A[i].x) <= 1e4 && abs(A[i].y) <= 1e4))
                        f = 1;
        }
        Point p, q;
```

```
        p.scan();
        q.scan();
        double k = q.x - p.x, b = q.y - p.y;
        double s = 0;
        for (int i = 0; i < n; i++) {
                if (len(A[i], p) > s && len(A[i], q) < len(A[i], p)) {
                        s = len(A[i], p);
                        Max = max(abs(A[i].x), abs(A[i].y));

                }
        }
        double pr = 1.0;

        Max = max(pr, Max) * 600;
        if (!f)
                Max = 1e4;
        vector<pair<double, int>> ans = ans_f({ p.x + k * Max , p.y + b * Max }, A);

        sort(ans.begin(), ans.end());
        double eps = 0.000000001;
        if (abs(ans[0].first - ans[1].first) < eps) {
                cout << -1;
        }
        else {
                cout << ans[0].second + 1;
        }

        return 0;
}
```

Task F (—)