

# Олимпиада СПбГУ по информатике 2018/19 учебного года

Черковец Денис Олегович

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

## Task A (100)

```
#include <iostream>

using namespace std;

int main()
{
    int n, m;

    cin >> n >> m;
    for (int i = 0; i < 30; i++)
        if (m == (1 << i) * n)
        {
            cout << "Yes";
            return 0;
        }
        else if ((1 << i) * n > m)
            break;
    cout << "No";
}
```

## Task B (100)

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    int n;
    string str;

    cin >> n >> str;
    for (int i = 0; i < n - 1; i++)
        if ((str[i] == 'o' && str[i + 1] == 'r') || (str[i] == 'r' && str[i + 1] == 'o'))
        {
            cout << "Yes";
            return 0;
        }
    for (int i = 0; i < n - 2; i++)
        if (str[i] == 'o' && str[i + 2] == 'r')
        {
            cout << "Yes";
            return 0;
        }
    cout << "No";
}
```

## Task C (100)

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

using namespace std;

const int SIZE = 1e5;
vector<int> graph[SIZE];
int cnt[SIZE], par[SIZE];

void dfs(int v, int curr_par = -1)
{
    cnt[v] = 1;
    par[v] = curr_par;
    for (int u : graph[v])
        if (u != par[v])
        {
            dfs(u, v);
            cnt[v] += cnt[u];
        }
}

int main()
{
    ios::sync_with_stdio(0);
    int n, u, v, ans;

    cin >> n;
    for (int i = 0; i < n - 1; i++)
    {
        cin >> u >> v;
        u--;
        v--;
        graph[u].push_back(v);
        graph[v].push_back(u);
    }
    dfs(0, -1);
    for (int i = 0; i < n; i++)
    {
        ans = cnt[0] - cnt[i] + 1;
        for (int u : graph[i])
            if (u != par[i])
                ans = max(ans, cnt[u] + 1);
        cout << ans << ' ';
    }
}
```

## Task D (60)

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string passw, split[4], req, ans;
    int n, p, t;

    ans.resize(9);
    cin >> req >> t >> n >> p;
    if (req == "split")
    {
        if (n == 3)
            for (int test = 0; test < t; test++)
            {
                cin >> passw;
                cout << 'a';
                for (char i = 0; i < 6; i++)
                    cout << passw[i];
                cout << '\n';
                cout << 'b';
                for (char i = 3; i < 9; i++)
                    cout << passw[i];
                cout << '\n';
                cout << 'c';
                for (char i = 0; i < 3; i++)
                    cout << passw[i];
                for (char i = 6; i < 9; i++)
                    cout << passw[i];
                cout << endl;
            }
        else if (n == 5)
            for (int test = 0; test < t; test++)
            {
                cin >> passw;
                for (char i = 0; i < 2; i++)
                {
                    cout << (char)('a' + i);
                    for (char j = i; j < i + 6; j++)
                        cout << passw[j];
                    cout << '\n';
                }
                cout << 'c';
                for (int i = 3; i < 9; i++)
                    cout << passw[i];
                cout << '\n';
                for (char i = 0; i < 2; i++)
                {
                    cout << (char)('d' + i);
                    for (char i = 0; i < 3; i++)
                        cout << passw[i];
                    for (char i = 6; i < 9; i++)
                        cout << passw[i];
                    cout << '\n';
                }
                cout << endl;
            }
    }
    else
    {
        if (n == 3)
        {
            for (int test = 0; test < t; test++)
            {
                cin >> split[0] >> split[1];
                for (char i = 0; i < 2; i++)
                {
```

```

        if (split[i][0] == 'a')
            for (char j = 0; j < 6; j++)
                ans[j] = split[i][j + 1];
        else if (split[i][0] == 'b')
            for (char j = 0; j < 6; j++)
                ans[j + 3] = split[i][j + 1];
        else
        {
            for (char j = 0; j < 3; j++)
                ans[j] = split[i][j + 1];
            for (char j = 0; j < 3; j++)
                ans[j + 6] = split[i][j + 4];
        }
    }
    cout << ans << endl;
}
}
else if (n == 5)
{
    for (int test = 0; test < t; test++)
    {
        cin >> split[0] >> split[1] >> split[2];
        for (char i = 0; i < 3; i++)
        {
            if (split[i][0] <= 'b')
                for (char j = 0; j < 6; j++)
                    ans[j + split[i][0] - 'a'] = split[i][j + 1];
            else if (split[i][0] == 'c')
                for (char j = 0; j < 6; j++)
                    ans[j + 3] = split[i][j + 1];
            else
            {
                for (char j = 0; j < 3; j++)
                    ans[j] = split[i][j + 1];
                for (char j = 0; j < 3; j++)
                    ans[j + 6] = split[i][j + 4];
            }
        }
        cout << ans << endl;
    }
}
}
}
/*
split 1 5 3 abcdefghi
aabbcdefg bbcdefg cdefghi dabcghi eabcghi

bbcdefg dabcghi cdefghi
*/

/*
passwords
apasswo bswords cpasrds
uhaaaaaa
auhaaaa baaaaaa cuhaaaa
aaaaaaa
aaaaaaa baaaaaa caaaaaa
plainword
aplainw binword cplaord

merge
4 3 7
apasswo bswords
baaaaaa cuhaaaa
aaaaaaa caaaaaa
binword cplaord

*/

```

## Task E (12)

```
#include <iostream>
#include <math.h>
#include <set>

using namespace std;

int main()
{
    ios::sync_with_stdio(0);
    const int SIZE = 1e5;
    static long double a, b, arr[SIZE][2], p[2], q[2], search[2], len, angle, dist, x, y; //c
    // - current
    set<pair<float, float>> p_set;
    int n, ans;

    cout.precision(5);
    cin >> n;
    for (int i = 0; i < n; i++)
        cin >> arr[i][0] >> arr[i][1];
    cin >> p[0] >> p[1] >> q[0] >> q[1];
    q[0] -= p[0];
    q[1] -= p[1];
    for (int i = 0; i < n; i++)
    {
        arr[i][0] -= p[0];
        arr[i][1] -= p[1];
    }
    if (q[1] == 0)
    {
        a = 0;
        b = 1;
    }
    else
    {
        a = 1;
        b = -q[0] / q[1];
    }
    for (int i = 0; i < n; i++)
        p_set.insert(make_pair(arr[i][0], arr[i][1]));
    //cout << endl;
    //for (int i = 0; i < n; i++)
    //    cout << fixed << arr[i][0] << ' ' << arr[i][1] << ' ';
    if (a == 0)
    {
        for (int i = 0; i < n; i++)
            if (arr[i][1] != 0 && p_set.find(make_pair(arr[i][0], -arr[i][1])) !=
                p_set.end())
            {
                cout << -1;
                return 0;
            }
    }
    else
    {
        len = a * a + b * b;
        //cout << a << ' ' << b << endl;
        for (int i = 0; i < n; i++)
        {
            dist = (a * arr[i][0] + b * arr[i][1]) * (a * arr[i][0] + b * arr[i][1]) /
                (a * a + b * b);
            angle = asin(sqrtl(dist / (arr[i][0] * arr[i][0] + arr[i][1] * arr[i][1])));
            x = -b / sqrtl(len * dist) / tan(angle);
            y = a / sqrtl(len * dist) / tan(angle);
            search[0] = 2 * x - arr[i][0];
            search[1] = 2 * y - arr[i][1];
            //cout << fixed << arr[i][0] << ' ' << arr[i][1] << ' ' << x << ' ' << y
            //    << ' ' << search[0] << ' ' << search[1] << endl;
            if (arr[i][1] != 0 && p_set.find(make_pair(search[0], search[1])) != p_set
```

```

        .end())
    {
        cout << -1;
        return 0;
    }
}
ans = 0;
for (int i = 1; i < n; i++)
    if (arr[i][0] > arr[ans][0] || (arr[i][0] == arr[ans][0]) && fabs1(arr[i][1]) <
        fabs1(arr[ans][1]))
        ans = i;
cout << ans + 1;
}

```

## Task F (—)