

# Олимпиада СПбГУ по информатике 2018/19 учебного года

Альжанов Максим Булатович

A	B	C	D	E	F	Sum
100	100	100	60	100	7	467

## Task A (100)

```
a = int(input())
b = int(input())
for i in range(100):
    if a == b:
        print("YES")
        break
    a*=2
else:
    print("NO")
```

## Task B (100)

```
import re
s = re.compile(r"(o.r|or|ro)")
input()
e = input()
if s.search(e):
    print("YES")
else:
    print("NO")
```

## Task C (100)

```
#include <bits/stdc++.h>
using namespace std;
int solve();

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie();
    cout.tie();
    return solve();
}

vector<vector<int>> > g;
vector<int> s;
vector<int> ans;
int size(int v, int p) {
    int ans = 0;
    for(int u: g[v]) {
        if(u != p)
            ans += size(u, v);
    }
    s[v] = ans+1;
    return ans+1;
}

void dfs(int v, int p) {
    int maxx = 0;
    for(int u: g[v]) {
        maxx = max(maxx, s[u]);
    }
    for(int u: g[v]) {
        if(u!=p) {
            int sv = s[v];
            int su = s[u];
            s[v] = s[v]-s[u];
            s[u] = sv;
            dfs(u, v);
            s[u] = su;
            s[v] = sv;
        }
    }
    ans[v] = maxx+1;
}

int solve() {
    int n;
    cin >> n;
    g.resize(n);
    s.resize(n);
    ans.resize(n);
    for(int i = 0; i<n-1; i++) {
        int f, t;
        cin >> f >> t;
        f--;
        t--;
        g[f].push_back(t);
        g[t].push_back(f);
    }
    size(0, -1);
    dfs(0, -1);
    for(int i: ans) {
        cout << i << " ";
    }
    return 0;
}
```

## Task D (60)

```
def encode_3_7(s) -> list:
    s = s * 2
    ans = []
    for i in range(3):
        ans += [chr(i + ord('a')) + s[i * 6:i * 6 + 6]]
    return ans

def decode_3_7(a, b) -> str:
    s = ['_'] * 9
    o = (ord(a[0]) - ord('a')) * 6
    for i, c in enumerate(a[1:]):
        s[(i + o) % 9] = c
    o = (ord(b[0]) - ord('a')) * 6
    for i, c in enumerate(b[1:]):
        s[(i + o) % 9] = c
    return ''.join(s)

def encode_5_7(s) -> list:
    s = s * 4
    ans = []
    for i in range(5):
        ans += [chr(i + ord('a')) + s[i * 6:i * 6 + 6]]
    return ans

def decode_5_7(a, b, cc) -> str:
    s = ['_'] * 9
    o = (ord(a[0]) - ord('a')) * 6
    for i, c in enumerate(a[1:]):
        s[(i + o) % 9] = c
    o = (ord(b[0]) - ord('a')) * 6
    for i, c in enumerate(b[1:]):
        s[(i + o) % 9] = c
    o = (ord(cc[0]) - ord('a')) * 6
    for i, c in enumerate(cc[1:]):
        s[(i + o) % 9] = c
    return ''.join(s)

if input() == 'merge':
    t, n, p = map(int, input().split())
    func = {(3, 7): decode_3_7,
            (5, 7): decode_5_7}[n, p]
    for _ in range(t):
        print(func(*input().split()))
else:
    t, n, p = map(int, input().split())
    func = {(3, 7): encode_3_7,
            (5, 7): encode_5_7}[n, p]
    for _ in range(t):
        print(*func(input()))
```

## Task E (100)

```
n = int(input())
if(n == 1):
    print(1)
    exit(0)
ps = [tuple(map(int, input().split())) for _ in range(n)]
p = eval(input().replace('_', ','))
m = eval(input().replace('_', ','))
k = 10**19
e = (p[0]+(m[0]-p[0])*k, p[1]+(m[1]-p[1])*k)
def dist(a, b):
    return sum([(i-j)**2 for i,j in zip(a, b)])
d = []
for i, p in enumerate(ps):
    d += [(dist(e, p), i+1)]
d.sort()
if d[0][0] == d[1][0]:
    print(-1)
else:
    print(d[0][1])
```

## Task F (7)

```
#include <bits/stdc++.h>
#define x first
#define y second
using namespace std;
int solve();

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie();
    cout.tie();
    return solve();
}

int solve() {
    int n;
    int k;
    cin >> n >> k;
    vector<pair<int, int>> a(n);
    for(int i = 0; i < n; i++) {
        cin >> a[i].x >> a[i].y;
    }
    int maxx = 0;
    for(int d = 0; d < (1 << n); d++) {
        int r = 0, b = 0;
        int p = 0;
        for(int i = 0; i < n; i++) {
            if(d & (1 << i)) {
                r += k;
            } else {
                b += k;
            }
            int amr = min(r, a[i].x);
            r -= amr;
            p += amr;
            int amb = min(b, a[i].y);
            b -= amb;
            p += amb;
        }
        maxx = max(maxx, p);
    }
    cout << maxx << endl;
    return 0;
}
```