

Олимпиада СПбГУ по информатике 2018/19 учебного года

Крымов Александр Павлович

A	B	C	D	E	F	Sum
100	100	100	60	35	7	402

Task A (100)

```
#include <bits/stdc++.h>
using namespace std;

#define endl '\n'
void inline fastio(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
}

#define forn(i, n)          for(int i = 0; i < (int)n; i++)
#define ford(i, n)          for(int i = (int)n - 1; i >= 0; i--)
#define fori(i, n)          for(int i = 1; i < (int)n; i++)
#define rsz                  resize
#define sz                   size()
#define all(a)               a.begin(), a.end()
#define upsort(a)            sort(all(a))
#define mysort(a, cmp)       sort(all(a), cmp)
#define pb                   push_back
#define ins                  insert
#define mp                   make_pair

typedef long long ll;
typedef unsigned long long ull;
typedef string str;
typedef long double ld;

typedef vector<int> vint;
typedef vector< vector<int> > mint;

typedef vector<long long> vll;
typedef vector< vector<long long> > mll;

//-----PROBLEM SOLVE-----
int n, m;

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

    cin >> n >> m;

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

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

```
    }  
    return 0;  
}
```

Task B (100)

```
#include <bits/stdc++.h>
using namespace std;

#define endl '\n'
void inline fastio(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
}

#define forn(i, n)          for(int i = 0; i < (int)n; i++)
#define ford(i, n)          for(int i = (int)n - 1; i >= 0; i--)
#define fori(i, n)          for(int i = 1; i < (int)n; i++)
#define rsz                  resize
#define sz                   size()
#define all(a)               a.begin(), a.end()
#define upsort(a)            sort(all(a))
#define mysort(a, cmp)       sort(all(a), cmp)
#define pb                   push_back
#define ins                  insert
#define mp                   make_pair

typedef long long ll;
typedef unsigned long long ull;
typedef string str;
typedef long double ld;

typedef vector<int> vint;
typedef vector< vector<int> > mint;

typedef vector<long long> vll;
typedef vector< vector<long long> > mll;

//-----PROBLEM SOLVE-----
int n;
str s;

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

    cin >> n >> s;

    forn(i, n){
        if(s[i] == 'o' && i + 1 < n){
            if(s[i+1] == 'r' || (i + 2 < n && s[i+2] == 'r')){
                cout << "Yes";
                exit(0);
            }
        }
        if(s[i] == 'r' && i - 1 >= 0){
            if(s[i-1] == 'o' || (i - 2 >= 0 && s[i-2] == 'o')){
                cout << "Yes";
                exit(0);
            }
        }
        if((s[i] == 'o' && (i + 1 < n && s[i+1] == 'r')) || (s[i] == 'r' && (i + 1 < n && s[i+1] == 'o'))){
            cout << "Yes";
            exit(0);
        }
    }

    cout << "No";

    return 0;
}
```

Task C (100)

```
#include <bits/stdc++.h>
using namespace std;

#define endl '\n'
void inline fastio(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
}

#define forn(i, n)          for(int i = 0; i < (int)n; i++)
#define ford(i, n)          for(int i = (int)n - 1; i >= 0; i--)
#define forl(i, n)          for(int i = 1; i < (int)n; i++)
#define rsz                  resize
#define sz                   size()
#define all(a)               a.begin(), a.end()
#define upsort(a)            sort(all(a))
#define mysort(a, cmp)       sort(all(a), cmp)
#define pb                   push_back
#define ins                   insert
#define mp                    make_pair

typedef long long ll;
typedef unsigned long long ull;
typedef string str;
typedef long double ld;

typedef vector<int> vint;
typedef vector< vector<int> > mint;

typedef vector<long long> vll;
typedef vector< vector<long long> > mll;

const int INF = 1e9;

//-----PROBLEM SOLVE-----
int n;
mint g;

vint fdown;
vint p;

int dfs0(int v, int pr = -1){
    int res = 0;

    for(auto to: g[v]){
        if(to == pr) continue;

        p[to] = v;
        res += dfs0(to, v);
    }

    return fdown[v] = res + 1;
}

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

    cin >> n;
    g.rsz(n);

    fdown.rsz(n, 0);

    p.rsz(n, -1);
    forn(i, n-1){
        int u, v;
```

```

        cin >> u >> v;
        u--; v--;
        g[u].pb(v);
        g[v].pb(u);
    }

    dfs0(0);

    forn(i, n){
        int res = 0;
        for(auto to:g[i]){
            if(to == p[i]){
                res = max(res, n - fdown[i]);
            } else {
                res = max(res, fdown[to]);
            }
        }

        cout << res + 1 << "␣";
    }

    return 0;
}

```

Task D (60)

```
#include <bits/stdc++.h>
using namespace std;

#define endl '\n'
void inline fastio(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
}

#define forn(i, n)          for(int i = 0; i < (int)n; i++)
#define ford(i, n)          for(int i = (int)n - 1; i >= 0; i--)
#define fori(i, n)          for(int i = 1; i < (int)n; i++)
#define rsz                  resize
#define sz                   size()
#define all(a)               a.begin(), a.end()
#define upsort(a)            sort(all(a))
#define mysort(a, cmp)       sort(all(a), cmp)
#define pb                   push_back
#define ins                   insert
#define mp                    make_pair

typedef long long ll;
typedef unsigned long long ull;
typedef string str;
typedef long double ld;

typedef vector<int> vint;
typedef vector< vector<int> > mint;

typedef vector<long long> vll;
typedef vector< vector<long long> > mll;

typedef vector<string> vstr;

const int INF = 1e9;

//-----PROBLEM SOLVE-----
str type;
int t, n, p;

map<int, int> tp1_map;
map<int, int> tp2_map;
map<int, int> tp3_map;
map<int, int> tp4_map;
map<int, int> tp5_map;

vstr code_3(str s){
    str tp1(7, 'a');
    str tp2(7, 'b');
    str tp3(7, 'c');

    tp1[1] = s[0];
    tp1[2] = s[1];
    tp1[3] = s[2];
    tp1[4] = s[4];
    tp1[5] = s[6];
    tp1[6] = s[8];

    tp2[1] = s[0];
    tp2[2] = s[1];
    tp2[3] = s[2];
    tp2[4] = s[3];
    tp2[5] = s[5];
    tp2[6] = s[7];

    tp3[1] = s[4];
    tp3[2] = s[6];
    tp3[3] = s[8];
    tp3[4] = s[3];
```

```

    tp3[5] = s[5];
    tp3[6] = s[7];

    vstr res;
    res.pb(tp1);
    res.pb(tp2);
    res.pb(tp3);

    return res;
}

str decode_3(vstr data){
    str tp1 = "", tp2 = "", tp3 = "";
    forn(i, data.sz){
        if(data[i][0] == 'a'){
            tp1 = data[i];
        }
        if(data[i][0] == 'b'){
            tp2 = data[i];
        }
        if(data[i][0] == 'c'){
            tp3 = data[i];
        }
    }

    str res(9, '\0');
    if(tp3 == ""){
        res[0] = tp1[1];
        res[1] = tp1[2];
        res[2] = tp1[3];

        res[3] = tp2[4];
        res[4] = tp1[4];
        res[5] = tp2[5];
        res[6] = tp1[5];
        res[7] = tp2[6];
        res[8] = tp1[6];
    } else if(tp1 == ""){
        res[0] = tp2[1];
        res[1] = tp2[2];
        res[2] = tp2[3];

        res[3] = tp3[4];
        res[4] = tp3[1];
        res[5] = tp3[5];
        res[6] = tp3[2];
        res[7] = tp3[6];
        res[8] = tp3[3];
    } else if(tp2 == ""){
        res[0] = tp1[1];
        res[1] = tp1[2];
        res[2] = tp1[3];

        res[3] = tp3[4];
        res[4] = tp3[1];
        res[5] = tp3[5];
        res[6] = tp3[2];
        res[7] = tp3[6];
        res[8] = tp3[3];
    }

    return res;
}

vstr code_5(str s){
    str tp1(7, 'a');
    str tp2(7, 'b');
    str tp3(7, 'c');
    str tp4(7, 'd');
    str tp5(7, 'e');

    for(auto t:tp1_map){
        tp1[t.first] = s[t.second];
    }
}

```

```

    for (auto t : tp2_map) {
        tp2[t.first] = s[t.second];
    }
    for (auto t : tp3_map) {
        tp3[t.first] = s[t.second];
    }
    for (auto t : tp4_map) {
        tp4[t.first] = s[t.second];
    }
    for (auto t : tp5_map) {
        tp5[t.first] = s[t.second];
    }

    vstr res;
    res.pb(tp1);
    res.pb(tp2);
    res.pb(tp3);
    res.pb(tp4);
    res.pb(tp5);

    return res;
}

str decode_5(vstr data) {
    str tp1 = "", tp2 = "", tp3 = "", tp4 = "", tp5 = "";
    forn(i, data.sz) {
        if (data[i][0] == 'a') {
            tp1 = data[i];
        }
        if (data[i][0] == 'b') {
            tp2 = data[i];
        }
        if (data[i][0] == 'c') {
            tp3 = data[i];
        }
        if (data[i][0] == 'd') {
            tp4 = data[i];
        }
        if (data[i][0] == 'e') {
            tp5 = data[i];
        }
    }

    str res(9, '_');
    if (tp1 != "") {
        for (auto t : tp1_map) {
            res[t.second] = tp1[t.first];
        }
    }
    if (tp2 != "") {
        for (auto t : tp2_map) {
            res[t.second] = tp2[t.first];
        }
    }
    if (tp3 != "") {
        for (auto t : tp3_map) {
            res[t.second] = tp3[t.first];
        }
    }
    if (tp4 != "") {
        for (auto t : tp4_map) {
            res[t.second] = tp4[t.first];
        }
    }
    if (tp5 != "") {
        for (auto t : tp5_map) {
            res[t.second] = tp5[t.first];
        }
    }

    return res;
}

int main() {

```



```

#ifdef _DEBUG_
freopen("input.txt", "r", stdin);
freopen("output.txt", "w", stdout);
#endif
fastio();

tp1_map[1] = 0;
tp1_map[2] = 1;
tp1_map[3] = 2;
tp1_map[4] = 4;
tp1_map[5] = 6;
tp1_map[6] = 8;

tp2_map[1] = 0;
tp2_map[2] = 1;
tp2_map[3] = 2;
tp2_map[4] = 3;
tp2_map[5] = 5;
tp2_map[6] = 7;

tp3_map[1] = 4;
tp3_map[2] = 6;
tp3_map[3] = 8;
tp3_map[4] = 3;
tp3_map[5] = 5;
tp3_map[6] = 7;

tp4_map[1] = 0;
tp4_map[2] = 2;
tp4_map[3] = 4;
tp4_map[4] = 6;
tp4_map[5] = 7;
tp4_map[6] = 8;

tp5_map[1] = 1;
tp5_map[2] = 3;
tp5_map[3] = 5;
tp5_map[4] = 6;
tp5_map[5] = 7;
tp5_map[6] = 8;

cin >> type;
cin >> t >> n >> p;

str s;
if (type == "split"){
    forn(i, t){
        cin >> s;

        vstr res;

        if (n == 3){
            res = code_3(s);
        }
        if (n == 5){
            res = code_5(s);
        }

        for (auto t:res) cout << t << "_";
        cout << endl;
    }
    exit(0);
}

if (type == "merge"){
    forn(i, t){
        vstr data;
        forn(j, n/2 + 1){
            str bf;
            cin >> bf;

            data.pb(bf);
        }
    }
}

```

```
        if (n == 3){
            cout << decode_3(data) << endl;
        }
        if (n == 5){
            cout << decode_5(data) << endl;
        }
    }
    exit(0);
}

return 0;
}
```

Task E (35)

```
#include <bits/stdc++.h>
using namespace std;

#define endl '\n'
void inline fastio(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
}

#define forn(i, n)          for(int i = 0; i < (int)n; i++)
#define ford(i, n)          for(int i = (int)n - 1; i >= 0; i--)
#define fori(i, n)           for(int i = 1; i < (int)n; i++)
#define rsz                  resize
#define sz                    size()
#define all(a)                a.begin(), a.end()
#define upsort(a)             sort(all(a))
#define mysort(a, cmp)         sort(all(a), cmp)
#define pb                    push_back
#define ins                    insert
#define mp                     make_pair

typedef long long ll;
typedef unsigned long long ull;
typedef string str;
typedef long double ld;

typedef vector<int> vint;
typedef vector< vector<int> > mint;

typedef vector<long long> vll;
typedef vector< vector<long long> > mll;

typedef vector<string> vstr;

const int INF = 1e9;

struct pt{
    ld x, y;
    int id = 0;
    pt(ld _x = 0, ld _y = 0){
        x = _x;
        y = _y;
    }
};

bool cmp(pt a, pt b){
    return (a.x < b.x) || (a.x == b.x && a.y < b.y);
}

//-----PROBLEM SOLVE-----
int n;
vector<pt> city;
pt p, q;

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

    cin >> n;
    city.rsz(n);

    forn(i, n){
        cin >> city[i].x >> city[i].y;
        city[i].y = abs(city[i].y);
        city[i].id = i;
    }
}
```

```

cin >> p.x >> p.y >> q.x >> q.y;

mysort(city, cmp);

pt our_city;
if(p.x < q.x){
    our_city = city.back();
}else{
    our_city = city.front();
}

forn(i, n){
    if(city[i].x == our_city.x){
        our_city = city[i];
        break;
    }
}

int cnt = 0;
forn(i, n){
    if(city[i].x == our_city.x && city[i].y == our_city.y) cnt++;
}

if(cnt > 1){
    cout << -1;
}else{
    cout << our_city.id + 1;
}

return 0;
}

```

Task F (7)

```
#include <bits/stdc++.h>
using namespace std;

#define endl '\n'
void inline fastio(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);
}

#define forn(i, n)          for(int i = 0; i < (int)n; i++)
#define ford(i, n)          for(int i = (int)n - 1; i >= 0; i--)
#define fori(i, n)          for(int i = 1; i < (int)n; i++)
#define rsz                  resize
#define sz                   size()
#define all(a)               a.begin(), a.end()
#define upsort(a)            sort(all(a))
#define mysort(a, cmp)       sort(all(a), cmp)
#define pb                   push_back
#define ins                   insert
#define mp                    make_pair

typedef long long ll;
typedef unsigned long long ull;
typedef string str;
typedef long double ld;

typedef vector<int> vint;
typedef vector< vector<int> > mint;

typedef vector<long long> vll;
typedef vector< vector<long long> > mll;

typedef vector<string> vstr;
typedef pair<int, int> pii;

const int INF = 1e9;

//-----PROBLEM SOLVE-----
int n, k;
vector<pii> a;

void stupid_solve(){
    int best_ans = 0;

    for(int mask = 0; mask < (1 << n); mask++){
        int ost_first = 0, ost_second = 0;
        int ans = 0;

        forn(i, n){
            if(mask & (1 << i)){
                ost_first += k;
            } else {
                ost_second += k;
            }
        }

        int res_first = min(a[i].first, ost_first);
        int res_second = min(a[i].second, ost_second);

        ost_first -= res_first;
        ost_second -= res_second;

        ans += res_first + res_second;
    }

    best_ans = max(best_ans, ans);
}

cout << best_ans;
```

```

        exit(0);
    }
    mt19937 rnd;
    uniform_int_distribution<> gen;

    int iterations = 0;
    const int MAXIT = 1 * 1e8;

    int best_ans = 0;
    void solve(int day = 0, int ost_first = 0, int ost_second = 0, int cur_cost = 0){
        int res_first, res_second;
        iterations++;

        if(iterations > MAXIT){
            cout << best_ans << endl;
            exit(0);
        }

        if(day == n){
            best_ans = max(best_ans, cur_cost);
            return;
        }

        int a_ost_first = ost_first + k, a_ost_second = ost_second, a_cur_cost = cur_cost;

        res_first = min(a[day].first, a_ost_first);
        res_second = min(a[day].second, a_ost_second);

        a_ost_first -= res_first;
        a_ost_second -= res_second;
        a_cur_cost += res_first + res_second;

        int b_ost_first = ost_first, b_ost_second = ost_second + k, b_cur_cost = cur_cost;

        res_first = min(a[day].first, b_ost_first);
        res_second = min(a[day].second, b_ost_second);

        b_ost_first -= res_first;
        b_ost_second -= res_second;
        b_cur_cost += res_first + res_second;

        if(gen(rnd)&1){
            solve(day+1, a_ost_first, a_ost_second, a_cur_cost);
            solve(day+1, b_ost_first, b_ost_second, b_cur_cost);
        } else {
            solve(day+1, b_ost_first, b_ost_second, b_cur_cost);
            solve(day+1, a_ost_first, a_ost_second, a_cur_cost);
        }
    }

    int main(){
#ifdef _DEBUG_
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif
        rnd.seed(time(nullptr));
        fastio();

        cin >> n >> k;
        a.rsz(n);

        forn(i, n){
            cin >> a[i].first >> a[i].second;
        }

        solve();
        cout << best_ans << endl;

        return 0;
    }

```