Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	75	100	52	10	437

Task A ()

```
#include "bits/stdc++.h"

#define ll long long
#define ld long double
#define all(u) u.begin(), u.end()
#define dout cout << setpresicton(10) << fixed
#define endl '\n'

using namespace std;

mt19937 gen(time(nullptr));

const int mod = 1e9 + 7;

vector<int> ans(6, 0);
int k = 0;

struct ATree{
    struct ANode{
        ANode* lD;
        ANode* rD;
        int val;
        int sz;
        uint_fast32_t rand;
    };

    ANode* newNode(int val){
        return new ANode{
                nullptr,
                nullptr,
                val,
                1,
                gen()
        };
    }

    ANode* root;

    int getSz(ANode* v){
        if(!v)
            return 0;
        return v->sz;
    }

    void upd(ANode* v){
        v->sz = getSz(v->lD) + 1 + getSz(v->rD);
    }

    pair<ANode*, ANode*> split(ANode* v, int k){
        if(!v)
            return {nullptr, nullptr};
        if(k <= getSz(v->lD)){
            auto a = split(v->lD, k);
            v->lD = a.second;
            upd(v);
            return {a.first, v};
```

```
        }else{
            auto a = split(v->rD, k - getSz(v->lD) - 1);
            v->rD = a.first;
            upd(v);
            return {v, a.second};
        }
    }

    ANode* merge(ANode* l, ANode* r){
        if(!l)
            return r;
        if(!r)
            return l;
        if(l->rand < r->rand){
            auto a = merge(l->rD, r);
            l->rD = a;
            upd(l);
            return l;
        }else{
            auto a = merge(l, r->lD);
            r->lD = a;
            upd(r);
            return r;
        }
    }

    void print(ANode* v){
        if(!v)
            return;
        print(v->lD);
        ans[v->val - 1] = ++k;
        print(v->rD);
    }

    void insert(int val, int poz){
        auto a1 = split(root, poz);
        root = merge(merge(a1.first, newNode(val)), a1.second);
    }
};

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

    ATree tree;
    tree.root = tree.newNode(1);
    int g;
    cin >> g;
    for (int i = 0; i < 5; ++i) {
        cin >> g;
        tree.insert(i + 2, g - 1);
    }
    tree.print(tree.root);
    for(auto u : ans)
        cout << u << ' ';
    cout << endl;
}
```

Task B ()

```
#include "bits/stdc++.h"

#define ll long long
#define ld long double
#define all(u) u.begin(), u.end()
#define dout cout << setpresicton(10) << fixed
#define endl '\n'

using namespace std;

mt19937 gen(time(nullptr));

const int mod = 1e9 + 7;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

    string mode;
    cin >> mode;
    if(mode == "first"){
        int n;
        cin >> n;
        ll h = 0;
        for (int i = 0; i < n; ++i) {
            int g;
            cin >> g;
            h += g;
        }
        cout << h / n + (ll)1e13 * (h % n) << endl;
    }else{
        int n;
        cin >> n;
        ll h = 0;
        ll f = 0;
        for (int i = 0; i < n; ++i) {
            ll g;
            cin >> g;
            h += g % (ll)1e13;
            f = g / (ll)1e13;
        }
        cout << h + f << endl;
    }
}
```

Task C ()

```
#include "bits/stdc++.h"

#define ll long long
#define ld long double
#define all(u) u.begin(), u.end()
#define dout cout << setpresicton(10) << fixed
#define endl '\n'

using namespace std;

mt19937 gen(time(nullptr));

const int mod = 1e9 + 7;

vector<pair<int, int>> avaliable;
vector<pair<int, int>> sizes_pr(3);
vector<pair<int, int>> sizes(3);


void fnd(){
    if(sizes[0].first < sizes[1].first or sizes[0].second < sizes[1].second)
        return;
    if(sizes[0].first == sizes[1].first){
        avaliable.emplace_back(sizes[0].first, sizes[0].second - sizes[1].second);
    }

    if(sizes[0].first < sizes[1].first or
       sizes[0].first < sizes[2].first or
       sizes[0].second < sizes[1].second or
       sizes[0].second < sizes[2].second)
        return;
    int u = sizes[0].first - sizes[1].first;
    int k = sizes[0].second - sizes[1].second;
    if(u == sizes[2].first){
        if(sizes[1].second == sizes[2].second){
            avaliable.emplace_back(sizes[0].first, k);
        }
        if(k == sizes[2].second){
            avaliable.emplace_back(sizes[1].first, k);
            avaliable.emplace_back(sizes[1].second, u);

        }
    }else if(u < sizes[2].first and sizes[2].second <= k){
        avaliable.emplace_back(u, sizes[1].second);
    }
    if(sizes[1].second == sizes[2].second and sizes[1].second == sizes[0].second){
        if(sizes[0].first - sizes[1].first - sizes[2].first > 0){
            avaliable.emplace_back(sizes[0].second, sizes[0].first - sizes[1].first - sizes[2].
                first);
        }
    }
}

vector<bool> used(3, false);
void dfs(int deep = 0){
    if(deep == 3){
        fnd();
        return;
    }
    for (int i = 0; i < 3; ++i) {
        if(!used[i]){
            sizes[deep] = sizes_pr[i];
            used[i] = true;
            dfs(deep + 1);
            swap(sizes[deep].first, sizes[deep].second);
            dfs(deep + 1);
            used[i] = false;
        }
    }
}
```

```
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

    for (int i = 0; i < 3; ++i) {
        cin >> sizes_pr[i].first >> sizes_pr[i].second;
        avaliable.emplace_back(sizes_pr[i].first , sizes_pr[i].second);
    }

    dfs();

    for (auto &u : avaliable) {
        if(u.first > u.second)
            swap(u.first , u.second);
    }
    sort(all(avaliable));
    avaliable.erase(unique(all(avaliable)), avaliable.end());
    for(auto u : avaliable)
        if(u.first > 0)
            cout << u.first << ' ' << u.second << endl;
}
```

Task D ()

```
#include "bits/stdc++.h"

#define ll long long
#define ld long double
#define all(u) u.begin(), u.end()
#define dout cout << setpresicton(10) << fixed
//#define endl '\n'

using namespace std;

mt19937 gen(time(nullptr));

const int mod = 1e9 + 7;

map<tuple<vector<int>, vector<bool>, bool>, int> sd;
map<tuple<vector<int>, vector<bool>, bool>, pair<int, int>> win_hod;

vector<int> st_v;
vector<int> now_v;
vector<bool> upped;


bool have_win(bool ch = true){
    if(sd[{now_v, upped, ch}])
        return sd[{now_v, upped, ch}] - 1;
    if(ch){
        for (int i = 0; i < now_v.size(); ++i) {
            for (int j = 0; j < now_v[i]; ++j) {
                int g = now_v[i];
                now_v[i] = j;
                bool k = have_win(!ch);
                now_v[i] = g;
                if(k){
                    win_hod[{now_v, upped, ch}] = {i + 1, j};
                    sd[{now_v, upped, ch}] = 2;
                    return true;
                }
            }
        }
        for (int i = 0; i < upped.size(); ++i) {
            if(!upped[i]){
                int g = now_v[i];
                now_v[i] = st_v[i];
                upped[i] = true;
                bool k = have_win(!ch);
                now_v[i] = g;
                upped[i] = false;
                if(k){
                    win_hod[{now_v, upped, ch}] = {i + 1, -1};
                    sd[{now_v, upped, ch}] = 2;
                    return true;
                }
            }
        }
        sd[{now_v, upped, ch}] = 1;
        return false;
    }else{
        bool out = true;
        for (int i = 0; i < now_v.size(); ++i) {
            for (int j = 0; j < now_v[i]; ++j) {
                int g = now_v[i];
                now_v[i] = j;
                out &= have_win(!ch);
                now_v[i] = g;
            }
        }
        for (int i = 0; i < upped.size(); ++i) {
            if(!upped[i]){
                int g = now_v[i];
                now_v[i] = st_v[i];
                upped[i] = true;
                out &= have_win(!ch);
```

```
                now_v[i] = g;
                upped[i] = false;
            }
        }
        sd[{now_v, upped, ch}] = (int)out + 1;
        return out;
    }
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

    int n;
    cin >> n;
    st_v.assign(n, 0);
    upped.assign(n, false);
    for (int i = 0; i < n; ++i) {
        cin >> st_v[i];
    }
    now_v = st_v;
    if(n == 3){
        cout << "1 0" << endl;
        upped[0] = true;
        while (true){
            int a, b;
            cin >> a >> b;
            if(a == -1)
                break;
            if(b == 0){
                upped[a - 1] = true;
                now_v[a - 1] = st_v[a - 1];
            }else{
                now_v[a - 1] = now_v[a - 1] - b;
            }
            bool ok = false;
            int sum_up = 0;
            for (int i = 0; i < n; ++i) {
                sum_up += !upped[i];
            }
            if(sum_up % 2 != 0){
                for (int i = 0; !ok and i < n; ++i) {
                    //important
                    if(now_v[i]){
                        cout << i + 1 << ' ' << now_v[i] << endl;
                        now_v[i] = 0;
                        ok = true;
                    }
                }
                for (int i = 0; !ok and i < n; ++i) {
                    //important
                    if (!upped[i]) {
                        cout << i + 1 << ' ' << 0 << endl;
                        upped[i] = true;
                        ok = true;
                    }
                }
            }else {
                for (int i = 0; !ok and i < n; ++i) {
                    //important
                    if (!upped[i]) {
                        cout << i + 1 << ' ' << 0 << endl;
                        upped[i] = true;
                        ok = true;
                    }
                }
                for (int i = 0; !ok and i < n; ++i) {
                    //important
                    if(now_v[i]){
                        cout << i + 1 << ' ' << now_v[i] << endl;
                        now_v[i] = 0;
                        ok = true;
                    }
```

```
                }
            }
        }

        return 0;
    }
    if(!have_win()){
        cout << "-1 -1" << endl;
    }else{
        while (true){
            have_win();
            if(!have_win()){
                cout << "-1 -1" << endl;
            }
            auto h = win_hod[{now_v, upped, true}];
            if(h.second == -1) {
                cout << h.first << ' ' << 0 << endl;
                now_v[h.first - 1] = st_v[h.first - 1];
                upped[h.first - 1] = true;
            }else {
                cout << h.first << ' ' << now_v[h.first - 1] - h.second << endl;
                now_v[h.first - 1] = h.second;
            }
            int a, b;
            cin >> a >> b;
            if(a == -1)
                ::exit(0);
            if(b == 0){
                upped[a - 1] = false;
                now_v[a - 1] = st_v[a - 1];
            }else{
                now_v[a - 1] -= b;
            }
        }
    }
    cout << have_win() << endl;
}
```

Task E ()

```
#include "bits/stdc++.h"

#define ll long long
#define ld long double
#define all(u) u.begin(), u.end()
#define dout cout << setpresicton(10) << fixed
//#define endl '\n'

using namespace std;

mt19937 gen(time(nullptr));

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

    int t;
    cin >> t;
    string mode;
    cin >> mode;
    while (t--){
        if(mode == "transmit"){
            int n;
            cin >> n;
            vector<vector<bool>> m(10, vector<bool>(10, false));
            for (int i = 0; i < 10; ++i) {
                for (int j = i + 1; j < 10; ++j) {
                    m[i][j] = true;
                }
            }
            for (int i = 0; i < 10; ++i) {
                if(n & (1<<i))
                    m[9][i] = true;
            }
            for (int i = 0; i < 10; ++i) {
                for (int j = 0; j < 10; ++j) {
                    cout << m[i][j];
                }
                cout << endl;
            }
            cout << endl;
        }else{
            vector<vector<int>> m(10, vector<int>(10, 0));
            for (int i = 0; i < 10; ++i) {
                for (int j = 0; j < 10; ++j) {
                    char e;
                    cin >> e;
                    m[i][j] = e == '1';
                }
            }
            vector<int> can_bee;
            for (int i = 0; i < 10; ++i) {
                set<int> u;
                vector<int> lk(10, -1);
                for (int j = 0; j < 10; ++j) {
                    int h = 0;
                    for (int k = 0; k < 10; ++k) {
                        if(k == i)
                            continue;
                        h += m[k][j];
                    }
                    lk[h] = j;
                    u.emplace(h);
                }
                bool ok = true;
                for (int j = 0; j < 10; ++j) {
                    if(!u.count(j))
                        ok = false;
                }
                if(ok){
                    int ans = 0;
```

```
                    for (int j = 0; j < 10; ++j) {
                        if(m[i][lk[j]])
                            ans += 1<<j;
                    }
                    can_bee.emplace_back(ans);
                }
            }
            int ans = can_bee[0];
            for (int i : can_bee) {
                ans = min(ans, i);
            }
            cout << ans << endl;
        }
    }
}
```

Task F ()

```
#include "bits/stdc++.h"

#define ll long long
#define ld long double
#define all(u) u.begin(), u.end()
#define dout cout << setpresicton(10) << fixed
#define endl '\n'
#define int ll

using namespace std;

mt19937 gen(time(nullptr));

const int mod = 1e9 + 7;

vector<pair<int, int>> parse(string &h1){
    vector<pair<int, int>> hg1;
    for (int i = h1.size() - 1; i >= 0; --i) {
        if (h1[i] == ')') {
            --i;
            string cnt;
            while (h1[i] != '|') {
                cnt = h1[i] + cnt;
                --i;
            }
            --i;
            int g = h1[i] - '0';
            --i;
            hg1.emplace_back(stoi(cnt), g);
        }else
            hg1.emplace_back(1, h1[i] - '0');
    }
    return hg1;
}

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

    string h1, h2;
    cin >> h1 >> h2;
    vector<pair<int, int>> hg1 = parse(h1);
    vector<pair<int, int>> hg2 = parse(h2);
    int p1 = 0, p2 = 0;
    vector<pair<int, int>> out;
    bool plus_one = false;
    while (p1 < hg1.size() or p2 < hg2.size()){
        if(p1 == hg1.size()){
            if(plus_one){
                if(hg2[p2].second != 9){
                    out.emplace_back(1, hg2[p2].second + 1);
                    out.emplace_back(hg2[p2].first - 1, hg2[p2].second);
                    plus_one = false;
                }else{
                    out.emplace_back(1, 0);
                    out.emplace_back(hg2[p2].first - 1, 0);
                }
            }else
                out.emplace_back(hg2[p2]);
            ++p2;
        }else if(p2 == hg2.size()){
            if(plus_one){
                if(hg1[p1].second != 9){
                    out.emplace_back(1, hg1[p1].second + 1);
                    out.emplace_back(hg1[p1].first - 1, hg1[p1].second);
                    plus_one = false;
                }else{
                    out.emplace_back(1, 0);
                    out.emplace_back(hg1[p1].first - 1, 0);
                }
            }else
```

```
                out.emplace_back(hg1[p1]);
            ++p1;
        }else{
            if(hg1[p1].first < hg2[p2].first){
                out.emplace_back(1, (hg1[p1].second + hg2[p2].second + plus_one) % 10);
                if((hg1[p1].second + hg2[p2].second + plus_one) < 10){
                    if(hg1[p1].first != 1)
                        out.emplace_back(hg1[p1].first - 1, hg1[p1].second + hg2[p2].second);
                    plus_one = false;
                }else{
                    if(hg1[p1].first != 1)
                        out.emplace_back(hg1[p1].first - 1, (hg1[p1].second + hg2[p2].second + 1)
                            % 10);
                    plus_one = true;
                }
                hg2[p2].first -= hg1[p1].first;
                ++p1;
            }else{
                out.emplace_back(1, (hg1[p1].second + hg2[p2].second + plus_one) % 10);
                if((hg1[p1].second + hg2[p2].second + plus_one) < 10){
                    if(hg1[p1].first != 1)
                        out.emplace_back(hg2[p2].first - 1, hg1[p1].second + hg2[p2].second);
                    plus_one = false;
                }else{
                    if(hg1[p1].first != 1)
                        out.emplace_back(hg2[p2].first - 1, (hg1[p1].second + hg2[p2].second + 1)
                            % 10);
                    plus_one = true;
                }
                hg1[p1].first -= hg2[p2].first;
                ++p2;
            }
        }
        while (p1 < hg1.size() and hg1[p1].first == 0)
            ++p1;
        while (p2 < hg2.size() and hg2[p2].first == 0)
            ++p2;
    }
    if(plus_one)
        out.emplace_back(1, 1);
    reverse(all(out));
    for (auto h : out) {
        if(h.first == 0)
            continue;
        if(h.first == 1)
            cout << h.second;
        else
            cout << "(" << h.second << "|" << h.first << ")";
    }
    cout << endl;
}
```