

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	28	10	398

Task A ()

```
#include "iostream"
#include "vector"
#include "algorithm"

using namespace std;

#define ll long long
#define all(x) (x).begin(), (x).end()

const int n = 6;

signed main() {
    vector<int> a(n);
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
    }
    vector<int> order;
    for (int i = 0; i < n; ++i) {
        vector<int> new_order;
        for (int j = 0; j < a[i] - 1; ++j) {
            new_order.push_back(order[j]);
        }
        new_order.push_back(i);
        for (int j = a[i] - 1; j < order.size(); ++j) {
            new_order.push_back(order[j]);
        }
        order = new_order;
        /*for (auto el: order) {
            cout << el << " ";
        }
        cout << '\n';*/
    }
    vector<int> ans(n);
    for (int i = 0; i < n; ++i) {
        ans[order[i]] = i + 1;
    }
    for (auto el: ans) {
        cout << el << " ";
    }
    cout << '\n';
}
```


Task B ()

```
#include "iostream"
#include "vector"
#include "algorithm"

using namespace std;

#define ll long long
#define all(x) (x).begin(), (x).end()

const ll M = 10001;

signed main() {
    string s; cin >> s;
    if (s == "first") {
        int n; cin >> n;
        vector<ll> a(n);
        ll sum = 0;
        for (int i = 0; i < n; ++i) {
            cin >> a[i];
            sum += a[i];
        }
        cout << sum * M << '\n';
    } else {
        int n; cin >> n;
        vector<ll> b(n);
        ll sum = 0;
        for (int i = 0; i < n; ++i) {
            cin >> b[i];
            sum += b[i];
        }
        cout << sum % M + (sum - sum % M) / (n * M) << '\n';
    }
}
```


Task C ()

```
#include "iostream"
#include "vector"
#include "algorithm"
#include "set"

using namespace std;

#define ll long long
#define all(x) (x).begin(), (x).end()
#define ld long double

pair<int, int> sp(pair<int, int> p) {
    if (p.first <= p.second) return p;
    return {p.second, p.first};
}

struct rect {
    int x, y, w, h;

    bool operator==(const rect &r) const {
        return x == r.x and y == r.y and w == r.w and h == r.h;
    }
    rect intersect(const rect &r) const {
        int x1 = max(x, r.x), y1 = max(y, r.y);
        int x2 = min(x + w, r.x + r.w), y2 = min(y + h, r.y + r.h);
        return {x1, y1, x2 - x1, y2 - y1};
    }
    bool empty() const {
        return w <= 0 or h <= 0;
    }
};

bool ok(rect r1, rect r2, rect r3) {
    auto r4 = r1.intersect(r2), r5 = r1.intersect(r3), r6 = r2.intersect(r3);
    return (r4.empty() or r4 == r1 or r4 == r2) and (r5.empty() or r5 == r1 or r5 == r3) and (r6.
        empty() or r6 == r2 or r6 == r3);
}

//vector<pair<int, int>> stub = {{3, 2}, {2, 3}, {4, 6}};

signed main() {
    vector<pair<int, int>> a1(3);
    vector<int> mx(3), mn(3);
    for (int i = 0; i < 3; ++i) {
        cin >> a1[i].first >> a1[i].second;
        mx[i] = max(a1[i].first, a1[i].second);
        mn[i] = min(a1[i].first, a1[i].second);
    }
    set<pair<int, int>> st;
    for (auto el: a1) {
        st.insert(sp(el));
    }
    vector<vector<pair<int, int>>> vars = {
        {a1[0], a1[1], a1[2]},
        {a1[0], a1[2], a1[1]},
        {a1[1], a1[0], a1[2]},
        {a1[1], a1[2], a1[0]},
        {a1[2], a1[0], a1[1]},
        {a1[2], a1[1], a1[0]}
    };
    for (auto &ap: vars) {
        for (int mask = 0; mask < 8; ++mask) {
            auto a = ap;
            for (int j = 0; j < 3; ++j) {
                if ((mask >> j) % 2) swap(a[j].first, a[j].second);
            }
            if (a[0].second == a[1].second and a[1].first < a[0].first) {/**/ and (a[2].first <= a
                [1].first and a[2].second <= a[1].second or a[2].first >= a[0].first and a[2].
                second >= a[1].second)) {
                st.insert(sp({a[0].second, a[0].first - a[1].first}));
            }
            if (a[1].first < a[0].first and a[1].second < a[0].second and a[2].second <= a[0].
```



```

        second and a[1].first + a[2].first <= a[0].first and (a[1].second > a[0].second -
        a[2].second or a[1].second == a[0].second - a[2].second and a[1].first + a[2].
        first == a[0].first)) {
            st.insert(sp({a[1].first , a[0].second - a[1].second}));
        }
        if (a[1].first + a[2].first == a[0].first and a[1].second == a[2].second and a[1].
        second < a[0].second) {
            st.insert(sp({a[0].second - a[1].second , a[0].first}));
        }
        if (a[1].first + a[2].first < a[0].first and a[1].second == a[2].second and a[1].
        second == a[0].second) {
            st.insert(sp({a[0].second , a[0].first - a[1].first - a[2].first}));
        }
    }
}
for (auto el: st) {
    if (el.first <= 0 or el.second <= 0) exit(1);
    cout << el.first << "␣" << el.second << '\n';
}
}

```


Task D ()

```
#include "iostream"
#include "vector"
#include "algorithm"
#include "set"
#include "map"
#include "unordered_map"

using namespace std;

#define ll long long
#define all(x) (x).begin(), (x).end()
#define ld long double

const int mx = 50;
vector<int> start;
int masknow;
int n;

int get_hash(vector<int> &a, int mask, int move) {
    int p = 1, h = 0;
    for (int i = 0; i < a.size(); ++i) {
        h += a[i] * p;
        p *= (mx + 1);
    }
    return (h * (1 << n) + mask) * 2 + move;
}

unordered_map<int, pair<int, pair<int, int>>> mp;

pair<int, pair<int, int>> pos(vector<int> a, int mask, int move) {
    int h = get_hash(a, mask, move);
    if (mp.find(h) != mp.end()) return mp[h];
    /*if (count(all(a), 0) == a.size() and mask == 0) {
        return mp[p] = {1 - move, {-1, -1}};
    }*/
    bool win = false;
    int i1 = -1, num = -1;
    for (int i = 0; i < a.size() and not win; ++i) {
        for (int j = 1; j <= a[i]; ++j) {
            auto a1 = a;
            a1[i] -= j;
            auto res = pos(a1, mask, 1 - move);
            if (res.first == move) {
                win = true;
                i1 = i;
                num = j;
                break;
            }
        }
    }
    if (win) {
        return mp[h] = {move, {i1, num}};
    }
    for (int i = 0; i < n; ++i) {
        if ((mask >> i) % 2) {
            auto a1 = a;
            a1[i] = start[i];
            int mask1 = mask;
            mask1 -= (1 << i);
            auto res = pos(a1, mask1, 1 - move);
            if (res.first == move) {
                win = true;
                i1 = i;
                num = 0;
            }
        }
    }
    if (win) {
        return mp[h] = {move, {i1, num}};
    }
    return mp[h] = {1 - move, {-1, -1}};
}
```



```

void update(vector<int> &a, pair<int, int> p) {
    if (p.second == 0) {
        a[p.first] = start[p.first];
        masknow -= (1 << p.first);
    } else {
        a[p.first] -= p.second;
    }
}

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr); cout.tie(nullptr);
    cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
    }
    start = a;
    masknow = (1 << n) - 1;
    auto res = pos(a, masknow, 0);
    if (res.first == 1) {
        cout << "-1_1" << endl;
        return 0;
    }
    cout << res.second.first + 1 << "┘" << res.second.second << endl;
    update(a, res.second);
    int x, y; cin >> x >> y;
    while (x != -1) {
        x--;
        update(a, {x, y});
        res = pos(a, masknow, 0);
        cout << res.second.first + 1 << "┘" << res.second.second << endl;
        update(a, res.second);
        cin >> x >> y;
    }
}

```


Task E ()

```
#include "iostream"
#include "vector"
#include "algorithm"
#include "set"

using namespace std;

#define ll long long
#define all(x) (x).begin(), (x).end()
#define ld long double

const int n = 10;

signed main() {
    int num = 0;
    vector<int> a(n, 0);
    vector<int> b(n, n);
    vector<int> sumn(n * n + 1, 0);
    vector<vector<vector<int>>> vars(n * n + 1);
    while (a != b) {
        num++;
        int sum = 0;
        for (auto el: a) {
            sum += el;
        }
        sumn[sum]++;
        vars[sum].push_back(a);
        for (int i = n - 1; i >= 0; --i) {
            if (a[i] < n and (i == n - 1 or a[i] < a[i + 1])) {
                a[i]++;
                for (int j = i + 1; j < n; ++j) {
                    a[j] = a[i];
                }
                break;
            }
        }
        /*for (auto el: a) {
            cout << el << " ";
        }
        cout << '\n';*/
    }
    num++;
    int sum = 0;
    for (auto el: a) {
        sum += el;
    }
    sumn[sum]++;
    /*cout << num << '\n';
    int sumsq = 0;
    for (auto el: sumn) {
        cout << el << " ";
        sumsq += el * el;
    }
    cout << '\n';
    cout << sumsq << '\n';*/
    int test; cin >> test;
    string s; cin >> s;
    while (test--) {
        if (s == "transmit") {
            int x; cin >> x;
            int i = 0;
            for (; i <= n * n; ++i) {
                if (x >= sumn[i]) {
                    x -= sumn[i];
                } else {
                    break;
                }
            }
            vector<int> a1 = vars[i][x];
            vector<string> tab(n, "0000000000");
            for (int j = 0; j < n; ++j) {
                for (int k = 0; k < a1[j]; ++k) {
```



```

        tab[j][k] = '1';
    }
}
for (auto el: tab) {
    cout << el << '\n';
}
cout << '\n';
} else {
    vector<string> tab(n);
    for (int i = 0; i < n; ++i) {
        cin >> tab[i];
    }
    vector<int> a1(n, 0);
    int sum1 = 0;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            a1[i] += tab[i][j] - '0';
            sum1 += tab[i][j] - '0';
        }
    }
    sort(all(a1));
    int res = 0;
    for (int i = 0; i < sum1; ++i) {
        res += sumn[i];
    }
    for (int i = 0; i < vars[sum1].size(); ++i) {
        if (a1 == vars[sum1][i]) {
            res += i;
        }
    }
    cout << res << '\n';
}
}
}
}

```


Task F ()

```
#include "iostream"
#include "vector"
#include "algorithm"
#include "set"
#include "map"
#include "unordered_map"
#include "deque"

using namespace std;

#define ll long long
#define all(x) (x).begin(), (x).end()
#define ld long double

signed main() {
    //ios_base::sync_with_stdio(false);
    //cin.tie(nullptr); cout.tie(nullptr);
    string as, bs; cin >> as >> bs;
    vector<pair<int, int>> a, b;
    bool lp = false, rp = false;
    int l = 0, r = 0;
    for (auto c: as) {
        if (c == '(') {
            lp = true;
        } else if (c == '|') {
            lp = false;
            rp = true;
        } else if (c == ')') {
            a.emplace_back(l, r);
            rp = false;
            l = 0;
            r = 0;
        } else {
            if (lp) {
                l = c - '0';
            } else if (rp) {
                r = r * 10 + c - '0';
            } else {
                a.emplace_back(c - '0', 1);
            }
        }
    }
    lp = false; rp = false;
    l = 0; r = 0;
    for (auto c: bs) {
        if (c == '(') {
            lp = true;
        } else if (c == '|') {
            lp = false;
            rp = true;
        } else if (c == ')') {
            b.emplace_back(l, r);
            rp = false;
            l = 0;
            r = 0;
        } else {
            if (lp) {
                l = c - '0';
            } else if (rp) {
                r = r * 10 + c - '0';
            } else {
                b.emplace_back(c - '0', 1);
            }
        }
    }
    int add = 0;
    deque<pair<int, int>> res;
    while (not a.empty() and not b.empty()) {
        int mn = min(a.back().second, b.back().second);
        if (add == 0) {
            if (a.back().first + b.back().first < 10) {
                res.emplace_front(a.back().first + b.back().first, mn);
            }
        }
    }
}
```



```

    } else {
        res.emplace_front(a.back().first + b.back().first - 10, 1);
        if (mn > 1) res.emplace_front(a.back().first + b.back().first - 9, mn - 1);
        add = 1;
    }
} else {
    if (a.back().first + b.back().first + 1 < 10) {
        res.emplace_front(a.back().first + b.back().first + 1, 1);
        res.emplace_front(a.back().first + b.back().first, mn - 1);
        add = 0;
    } else {
        res.emplace_front(a.back().first + b.back().first - 9, mn);
    }
}
if (a.back().second == mn) {
    a.pop_back();
} else {
    a.back().second -= mn;
}
if (b.back().second == mn) {
    b.pop_back();
} else {
    b.back().second -= mn;
}
}
if (a.empty()) swap(a, b);
if (add) {
    while (not a.empty() and a.back().first == 9) {
        res.emplace_front(0, a.back().second);
        a.pop_back();
    }
    if (a.empty()) {
        res.emplace_front(1, 1);
    } else if (a.back().second == 1) {
        res.emplace_front(a.back().first + 1, 1);
        a.pop_back();
    } else {
        res.emplace_front(a.back().first + 1, 1);
        a.back().second--;
    }
}
for (int i = a.size() - 1; i >= 0; --i) {
    res.emplace_front(a[i]);
}
for (auto el: res) {
    if (el.second == 0) continue;
    if (el.second == 1) {
        cout << el.first;
    } else {
        cout << "(" << el.first << "|" << el.second << ")";
    }
}
cout << '\n';
}

```