Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	52	10	422

Task A ()

```cpp
#include <bits/stdc++.h>

typedef long long ll;
typedef long double ld;
using namespace std;

void solve() {
    vector<int> d(6);
    for (int& x : d)
        cin >> x;

    vector<int> tasks;
    for (int i = 0; i < 6; i++) {
        vector<int> new_tasks;
        for (int j = 0; j < d[i] - 1; j++) {
            new_tasks.push_back(tasks[j]);
        }

        new_tasks.push_back(i + 1);

        for (int j = d[i] - 1; j < tasks.size(); j++) {
            new_tasks.push_back(tasks[j]);
        }

        tasks = new_tasks;
        // for (int x : tasks)
        //     cout << x << ' ';
        // cout << '\n';
    }

    vector<int> answer(7);
    for (int i = 0; i < 6; i++)
        answer[tasks[i]] = i + 1;

    for (int i = 1; i < 7; i++)
        cout << answer[i] << ' ';
    cout << '\n';
}

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

    solve();
}
```

Task B ()

```cpp
#include <bits/stdc++.h>

typedef long long ll;
typedef long double ld;
using namespace std;

void solve() {
    string run;
    cin >> run;

    if (run == "first") {
        int n;
        cin >> n;

        ll s = 0;
        vector<ll> a(n);
        for (ll& x : a)
            cin >> x, s += x;

        s <<= 20;
        cout << s << '\n';
    } else {
        int n;
        cin >> n;

        ll bs = 0;
        vector<ll> b(n);
        for (ll& x : b)
            cin >> x, bs += (x & ((1 << 20) - 1));

        ll as = b[0] >> 20;
        // cout << as << ' ' << bs << '\n';
        cout << as + bs << '\n';
    }
}

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

    solve();
}
```

Task C ()

```cpp
#include <bits/stdc++.h>

typedef long long ll;
typedef long double ld;
using namespace std;

void solve() {
    vector<int> a(3), b(3);
    for (int i = 0; i < 3; i++)
        cin >> a[i] >> b[i];

    set<pair<int, int>> answers;
    for (int mask = 0; mask < (1 << 3); mask++) {
        for (int i = 0; i < 3; i++) {
            if (mask & (1 << i)) {
                swap(a[i], b[i]);
            }
        }

        for (int i = 0; i < 3; i++) {
            answers.emplace(min(a[i], b[i]), max(a[i], b[i]));

            for (int j = 0; j < 3; j++) {
                if (i == j)
                    continue;

                if (a[j] > a[i] || b[j] > b[i])
                    continue;

                if (b[j] == b[i] && a[j] < a[i])
                    answers.emplace(min(a[i] - a[j], b[i]), max(a[i] - a[j], b[i]));

                for (int k = 0; k < 3; k++) {
                    if (k == j || k == i)
                        continue;

                    // 1
                    if (b[j] == b[k] && b[j] == b[i] && a[j] + a[k] < a[i])
                        answers.emplace(min(a[i] - a[j] - a[k], b[i]), max(a[i] - a[j] - a[k], b[i
                            ]));

                    // 2
                    if (a[k] + a[j] == a[i] && b[j] == b[k] && b[i] - b[j] > 0)
                        answers.emplace(min(a[i], b[i] - b[j]), max(a[i], b[i] - b[j]));

                    // 3
                    if (a[k] + a[j] == a[i] && b[j] + b[k] > b[i] && b[k] <= b[i] && b[j] < b[i])
                        answers.emplace(min(a[i] - a[k], b[i] - b[j]), max(a[i] - a[k], b[i] - b[j
                            ]));

                    // 4
                    if (b[j] == b[i] && b[k] <= b[i] && a[i] - a[j] - a[k] > 0)
                        answers.emplace(min(b[j], a[i] - a[j] - a[k]), max(b[j], a[i] - a[j] - a[k
                            ]));

                    // 5
                    if (b[k] < b[i] && b[j] + b[k] > b[i] && a[j] + a[k] <= a[i])
                        answers.emplace(min(a[k], b[i] - b[k]), max(a[k], b[i] - b[k]));

                    if (a[j] + a[k] == a[i] && b[j] + b[k] == b[i])
                        answers.emplace(min(a[k], b[j]), max(a[k], b[j]));
                }
            }
        }

        for (int i = 0; i < 3; i++) {
            if (mask & (1 << i)) {
                swap(a[i], b[i]);
            }
        }
    }
```

```
    for (auto [x, y] : answers) {
        cout << x << ' ' << y << '\n';
    }
}

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

    solve();
}
```

Task D ()

```cpp
#include <bits/stdc++.h>

typedef long long ll;
typedef long double ld;
using namespace std;

struct Result {
    int wins = 0, loses = 0;
    pair<int, int> move = {-1, -1};
    bool guaranteed = false;
};

Result run(vector<int>& a, vector<int>& b, vector<bool>& used, int k) {
    int n = b.size();

    bool moved = false;
    auto result = Result();
    for (int i = 0; i < n; i++) {
        if (!used[i]) {
            int t = b[i];
            b[i] = a[i];
            used[i] = true;
            moved = true;

            auto next = run(a, b, used, k + 1);

            result.wins += next.loses;
            result.loses += next.wins;

            used[i] = false;
            b[i] = t;

            if (next.move.first == -1 || (k % 2 == 1 && !next.guaranteed)) {
                result.move = {i, 0};
                result.guaranteed = true;
            }

            if (k % 2 == 0 && !result.guaranteed && next.loses > 0) {
                result.move = {i, 0};
            }
        }

        for (int j = 0; j < 2; j++) {
            if (b[i] > j) {
                int t = b[i];
                b[i] = j;
                moved = true;

                auto next = run(a, b, used, k + 1);
                // if (k == 0) {
                //     cout << next.loses << ' ' << next.wins << '\n';
                //     cout << "(" << next.move.first << ' ' << next.move.second << ")" << '\n';
                // }

                result.wins += next.loses;
                result.loses += next.wins;

                b[i] = t;

                if (next.move.first == -1 || (k % 2 == 1 && !next.guaranteed)) {
                    result.move = {i, t - j};
                    result.guaranteed = true;
                }

                if (k % 2 == 0 && !result.guaranteed && next.loses > 0) {
                    result.move = {i, t - j};
                }
            }
        }
    }

    result.loses += !moved;
```

```
    return result;
}

void solve() {
    int n;
    cin >> n;

    vector<bool> used(n);
    vector<int> a(n);
    for (int& x : a)
        cin >> x;

    vector<int> b = a;
    auto move = [&](int i, int cnt) {
        cout << i + 1 << ' ' << cnt << endl;

        if (cnt == 0) {
            b[i] = a[i];
            used[i] = true;
        } else {
            b[i] -= cnt;
        }
    };

    auto listen = [&]() {
        int i, cnt;
        cin >> i >> cnt;

        if (i == -1 && cnt == -1) {
            exit(0);
        }

        i--;

        if (cnt == 0) {
            b[i] = a[i];
            used[i] = true;
        } else {
            b[i] -= cnt;
        }
    };

    if (n == 1 && false) {
        move(0, a[0]);
        listen();
        move(0, a[0]);
    } else {
        while (true) {
            auto result = run(a, b, used, 0);

            if (result.wins == 0) {
                cout << "-1 -1" << endl;
                exit(0);
            }

            auto [i, x] = result.move;
            move(i, x);
            listen();
        }
    }
}

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

    solve();
}
```

Task E ()

```
#include <bits/stdc++.h>

typedef long long ll;
typedef long double ld;
using namespace std;

int N = 10;

void run(vector<vector<int>>& nuggets, vector<int>& current, int k, int last) {
    if (k == N) {
        nuggets.push_back(current);
        return;
    }

    for (int i = last; i <= 10; i++) {
        current[k] = i;
        run(nuggets, current, k + 1, i);
    }
}

void solve() {
    int t;
    cin >> t;

    vector<vector<int>> nuggets;
    vector<int> current(10);
    run(nuggets, current, 0, 0);

    int m = nuggets.size();

    map<vector<int>, int> compressed;
    for (int i = 0; i < m; i++)
        compressed[nuggets[i]] = i;

    string type;
    cin >> type;

    while (t--) {
        if (type == "transmit") {
            int x;
            cin >> x;

            vector<int> ass = nuggets[x];
            vector<vector<char>> field(10, vector<char>(10, '0'));
            for (int i = 0; i < 10; i++) {
                for (int j = 0; j < ass[i]; j++) {
                    field[j][i] = '1';
                }
            }

            for (int i = 0; i < 10; i++) {
                cout << string(field[i].begin(), field[i].end()) << '\n';
            }
        } else {
            int result = 0;

            vector<vector<char>> field(10, vector<char>(10));
            for (int i = 0; i < 10; i++) {
                for (char& c : field[i]) {
                    cin >> c;
                }
            }

            vector<int> ass(10);
            for (int i = 0; i < 10; i++) {
                for (int j = 0; j < 10; j++) {
                    ass[i] += field[j][i] == '1';
                }
            }

            sort(ass.begin(), ass.end());
            cout << compressed[ass] << '\n';
```

```
            }
        }
    }

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

    solve();
}
```

Task F ()

```cpp
#include <bits/stdc++.h>

typedef long long ll;
typedef long double ld;
using namespace std;

void solve() {
    string a, b;
    cin >> a >> b;

    reverse(a.begin(), a.end());
    reverse(b.begin(), b.end());

    while (a.size() > b.size())
        b += '0';

    while (b.size() > a.size())
        a += '0';

    int additional = 0;
    string c;
    for (int i = 0; i < a.size(); i++) {
        additional += a[i] - '0';
        additional += b[i] - '0';

        c += '0' + (additional % 10);
        additional /= 10;
    }

    if (additional > 0)
        c += '0' + (additional % 10);

    reverse(c.begin(), c.end());
    cout << c << '\n';
}

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

    solve();
}
```