Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	55	25	480

Task A ()

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

using namespace std;

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

    int k;
    cin >> k;
    if (k == 1) {
        cout << 1;
    } else {
        k -= 2;
        string s = "234567890";
        cout << s[k % s.size()];
    }

    //12345690
}
```

Task B ()

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

using namespace std;

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

    int n, k;
    cin >> n >> k;
    int res = 0;
    set<char> cur;
    int ln = 0;
    string s;
    cin >> s;
    for (int i = 0; i < n; ++i) {
        if (ln == k || (cur.size() == 3 && cur.count(s[i]) == 0)) {
            res++;
            ln = 0;
            cur.clear();
        }

        ln++;
        cur.insert(s[i]);
    }
    res++;
    cout << res;
}
```

Task C ()

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

using namespace std;

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

    int n, x, y;
    cin >> n >> x >> y;
    vector<int> v(n);
    int sm_v = 0;
    vector<int> w(n);
    int sm_w = 0;
    for (int i = 0; i < n; ++i) {
        cin >> v[i];
        sm_v += v[i];
    }
    for (int i = 0; i < n; ++i) {
        cin >> w[i];
        sm_w += w[i];
    }
//    sort(v.begin(), v.end());
//    sort(w.begin(), w.end());

    int mn_pos_w = sm_w - y;

    vector<int> mx_w_for_v(x + 1, -1);
    vector<bitset<500>> item(x + 1, 0);
    mx_w_for_v[0] = 0;
    for (int i = 0; i < n; ++i) {
        for (int cur_v = x - v[i]; cur_v >= 0; --cur_v) {
            if (mx_w_for_v[cur_v] != -1) {
                if (mx_w_for_v[cur_v + v[i]] == -1 || mx_w_for_v[cur_v] + w[i] > mx_w_for_v[cur_v
                    + v[i]]) {
                    mx_w_for_v[cur_v + v[i]] = mx_w_for_v[cur_v] + w[i];
                    item[cur_v + v[i]] = item[cur_v];
                    item[cur_v + v[i]][i] = true;
                }
            }
        }
    }

//    int lst = -1;
    int pos = -1;
    for (int i = 0; i <= x; ++i) {
//        cerr << mx_w_for_v[i] << endl;
        if (mx_w_for_v[i] != -1 && mx_w_for_v[i] >= mn_pos_w) {
//            lst = item[i];
            pos = i;
        }
    }
    if (pos == -1) {
        cout << -1;
    } else {
        for (int i = 0; i < n; ++i) {
//            if (res[i]) {
//                x -= v[i];
//            } else {
//                y -= w[i];
//            }
            cout << (item[pos][i] ? 'x' : 'y');
        }
        assert(x >= 0);
        assert(y >= 0);
    }
}
```

Task D ()

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

using namespace std;

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

    int n;
    cin >> n;
    n *= 2;
    string s0;
    cin >> s0;

    string s;
    for (int i = 0; i < n; ++i) {
        s.push_back(s0[i]);

        if (s.size() > 1) {
            bool sq1 = s[s.size() - 1] == '[' || s[s.size() - 1] == ']';
            bool sq2 = s[s.size() - 2] == '[' || s[s.size() - 2] == ']';
            if (sq2 == sq1) {
                s.pop_back();
                s.pop_back();
            }
        }
    }
    n = s.size();

    vector<int> blocks;

    int cur = 0;

    for (int i = 0; i < n; ++i) {
        if (s[i] == '[' || s[i] == ']') {
            blocks.push_back(cur);
            cur = 0;
        } else {
            cur++;
        }
    }
    blocks.push_back(cur);

    int res = 0;

//    bool c1 = blocks.size() % 2 == 1;
//  [[[(
    for (int i = 0; i < blocks.size(); ++i) {
        res += blocks[i] % 2;
    }

    cout << res;
}
```

Task E ()

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

using namespace std;

int rpr(int a, int b, int c) {
    vector<int> cur = {a, b, c};
    sort(cur.begin(), cur.end());
    return cur[0] * 100 + cur[1] * 10 + cur[2];
}

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie();
    int c4 = 761232;

    int c = c4;
    string mode;
    cin >> mode;
    map<int, int> precalc;
    map<int, int> precalc2;
    for (int i = 0; i < 10; ++i) {
        for (int j = i + 1; j < 10; ++j) {
            for (int k = j + 1; k < 10; ++k) {
                for (int d = 0; d < 10; ++d) {
                    if (d == i || d == j || d == k) {
                        continue;
                    }

                    bool flag = true;
                    if (precalc.count(rpr(i, j, d)) && precalc[rpr(i, j, d)] == k) {
                        flag = false;
                    }
                    if (precalc.count(rpr(i, k, d)) && precalc[rpr(i, k, d)] == j) {
                        flag = false;
                    }
                    if (precalc.count(rpr(k, j, d)) && precalc[rpr(k, j, d)] == i) {
                        flag = false;
                    }

                    if (flag) {
                        precalc[rpr(i, j, k)] = d;
                        break;
                    }
                }
            }
        }
    }
    // 11 i j --> d
    // 11 i d -/-> j
    // 11 j d -/-> i

    for (int i = 0; i < 10; ++i) {
        for (int j = i + 1; j < 10; ++j) {
            for (int d = 0; d < 10; ++d) {
                if (d == i || d == j) {
                    continue;
                }

                bool flag = true;
                if (precalc2.count(rpr(11, i, d)) && precalc2[rpr(11, i, d)] == j) {
                    flag = false;
                }
                if (precalc2.count(rpr(11, j, d)) && precalc2[rpr(11, j, d)] == i) {
                    flag = false;
                }

                if (flag) {
                    precalc2[rpr(11, i, j)] = d;
                    break;
                }
            }
        }
    }

    vector<int> rndnum = {62464, 65059, 18213, 72807, 69360, 66289, 43543, 80825, 68990, 93119};
    map<int, int> cmpr;
    for (int i = 0; i < rndnum.size(); ++i) {
        cmpr[rndnum[i]] = i;
    }

// a b c --> d :

// a b d -/-> c
// a c d -/-> b
// b c d -/-> a

    int tests;
    cin >>
        tests;
    for (
            int test = 0;
            test < tests;
            ++test) {
        int n, k;
        cin >> n >>
            k;
        if (mode == "add") {
            if (n == 1000000) {
                vector<int> a(k);
                for (
                        int i = 0;
                        i < k;
                        ++i) {
                    cin >> a[i];
                    if (a[i] == c) {
                        return -1;
                    }
                }
                cout << c << '\n';
            } else if (n == 10) {
                vector<int> a;
                for (
                        int i = 0;
                        i < k;
                        ++i) {
                    int x;
                    cin >>
                        x;
                    x--;
                    a.
                            push_back(x);
                }
                cout << precalc[
                                rpr(a[0], a[1], a[2]
                                )] + 1 << '\n';
            } else {
                vector<int> a;
                for (int i = 0; i < k; ++i) {
                    int x;
                    cin >> x;
                    x--;
                    if (cmpr.count(x)) {
                        a.push_back(cmpr[x]);
                    }
                }

                assert(a.size() <= 7);
            }
        } else {
            if (n == 1000000) {
                for (
                        int i = 0;
                        i < k + 1; ++i) {
                    int x;
                    cin >>
                        x;
                    if (x != c) {
                        cout << x << '␣';
                    }
                }
                cout << '\n';
            } else if (n == 10) {
                vector<int> a;
                for (int i = 0; i < k + 1; ++i) {
                    int x;
                    cin >> x;
                    x--;
                    a.push_back(x);
                }
                sort(a.begin(), a.end());
                vector<int> a1 = {a[0], a[1], a[2]};
                int b1 = a[3];

                vector<int> a2 = {a[0], a[1], a[3]};
                int b2 = a[2];

                vector<int> a3 = {a[0], a[2], a[3]};
                int b3 = a[1];

                vector<int> a4 = {a[1], a[2], a[3]};
                int b4 = a[0];

                vector<int> res;
                if (precalc[
                            rpr(a1[0], a1[1], a1[2]
                            )] == b1) {
                    res = a1;
                } else if (precalc[
                                   rpr(a2[0], a2[1], a2[2]
                                   )] == b2) {
                    res = a2;
                } else if (precalc[
                                   rpr(a3[0], a3[1], a3[2]
                                   )] == b3) {
                    res = a3;
                } else {
                    res = a4;
                }
                for (
                    auto e
                        : res) {
                    cout << e + 1 << '␣';
                }

                cout << '\n';
            } else {

            }
        }

    }
}

// 4 6 -> 2
// 5 7 -> -2
// 2 3 7 --> 1
// 1 2 7 --> 3
```

Task F ()

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

using namespace std;

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

    int n;
    cin >> n;

    cout << "4" << '\n';
    cout << "0␣0" << '\n';
    cout << "0␣2" << '\n';
    cout << "2␣2" << '\n';
    cout << "2␣0" << '\n';

    vector<string> v = {
            "0␣2",
            "0␣-2",
            "2␣0",
            "-2␣0",
            "2␣2",
            "2␣-2",
            "-2␣2",
            "-2␣-2",
    };

    for (int i = 0; i < n; ++i) {
        cout << v[i] << '\n';
    }
}
```