Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	60	100	55	25	440

Task A ()

```
k = int(input())
n = k % 9
if k <= 9:
        print(k)
elif n == 1:
        print(0)
elif n == 0:
        print(9)
else:
        print(n)
```

Task B ()

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

using namespace std;

#define int long long

signed main() {
    int n, k; cin >> n >> k;
    string s; cin >> s;
    int ans = 0;
    int i = 0;
    int sz = 0;
    int kol = 0;
    vector<int> cnt(26, 0);
    while (i < n) {
        if (cnt[s[i] - 'a'] == 0) kol++;
        if (kol == 4 || sz == k) {
            for (int i = 0; i < 26; ++i) cnt[i] = 0;
            ans++;
            sz = 0;
            kol = 1;
        }
        sz++;
        cnt[s[i] - 'a']++;
        i++;
    }
    cout << ans + 1;
}
```

Task C ()

```
#include <bits/stdc++.h>
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("Ofast")
//#pragma GCC optimize("O3")

using namespace std;

#define int long long

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    int n, x, y; cin >> n >> x >> y;
    vector<int> v(n), w(n);
    int s = 0;
    for (int i = 0; i < n; ++i) {
        cin >> v[i];
        s += v[i];
    }
    for (int i = 0; i < n; ++i) cin >> w[i];
    int dp[n + 1][y + 1];
    int pr[n + 1][y + 1];
    for (int i = 0; i <= n; ++i) {
        for (int j = 0; j <= y; ++j) {
            dp[i][j] = -1;
            pr[i][j] = 0;
        }
    }
    dp[0][0] = 0;
    for (int i = 1; i <= n; ++i) {
        for (int j = 0; j <= y; ++j) {
            dp[i][j] = dp[i - 1][j];
            if (j >= w[i - 1] && dp[i - 1][j - w[i - 1]] != -1) {
                if (dp[i - 1][j - w[i - 1]] + v[i - 1] > dp[i][j]) {
                    pr[i][j] = 1;
                    dp[i][j] = dp[i - 1][j - w[i - 1]] + v[i - 1];
                }
            }
        }
    }
    vector<int> ans;
    int o = 0;
    int mxx = 0;
    for (int i = 0; i <= y; ++i) {
        if (dp[n][i] > mxx) {
            mxx = dp[n][i];
            o = i;
        }
    }
    int p = n;
    while (o != 0) {
        if (pr[p][o] != 0) {
            ans.push_back(p - 1);
            o -= w[p - 1];
        }
        p--;
    }
    int u = 0;
    for (int i = 0; i < ans.size(); ++i) {
        u += v[ans[i]];
    }
    if (s - u > x) {
        cout << -1;
        return 0;
    }
    string r;
    for (int i = 0; i < n; ++i) r += 'x';
    for (auto it: ans) r[it] = 'y';
    cout << r;
}
```

Task D ()

```
#include <bits/stdc++.h>
#pragma GCC optimize("no-stack-protector")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("O3")

using namespace std;

#define int long long

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    int n; cin >> n;
    string s; cin >> s;
    for (int i = 0; i < 2 * n; ++i) {
        if (s[i] == ')') s[i] = '(';
        if (s[i] == ']') s[i] = '[';
    }
    s += '#';
    int k = 1;
    set<pair<int, pair<int, int>>> st2;
    set<pair<pair<int, int>, int>> st1;
    for (int i = 1; i < s.size(); ++i) {
        if (s[i] == s[i - 1]) k++;
        else {
            int t = 0;
            if (s[i - 1] == '[') t = 1;
            st1.insert({{i, t}, -k});
            st2.insert({-k, {i, t}});
            k = 1;
        }
    }
    //for (auto it: st1) cout << it.first.first << " " << it.first.second << "   " << it.second <<
        endl;
    while (true) {
        //for (auto it: st1) cout << it.first.first << " " << it.first.second << "   " << it.
            second << endl;
        if (st1.size() == 0) break;
        auto it = st2.begin();
        if ((*it).first == -1) break;
        auto p = (*it);
        auto it1 = st1.lower_bound({p.second, p.first});
        auto pp = (*it1);
        if ((-p.first) % 2 == 1) {
            st1.erase(it1);
            st2.erase(it);
            p.first %= 2;
            st2.insert(p);
            st1.insert({p.second, p.first});
            continue;
        }
        if (it1 == st1.begin()) {
            st1.erase(it1);
            st2.erase(it);
            continue;
        }
        it1--;
        auto itt1 = it1;
        it1++;
        it1++;
        auto itt2 = it1;
        it1--;
        if (itt2 == st1.end()) {
            st1.erase(it1);
            st2.erase(it);
            continue;
        }
        if ((*itt1).first.second != (*itt2).first.second) {
            st1.erase(it1);
            st2.erase(it);
            continue;
```

```
        }
        auto p1 = (*itt1);
        auto p2 = (*itt2);
        st1.erase(it1);
        st1.erase(itt1);
        st1.erase(itt2);
        st2.erase(it);
        st2.erase({p1.second, p1.first});
        st2.erase({p2.second, p2.first});
        pair<pair<int, int>, int> ppp = {p2.first, p1.second + p2.second};
        st1.insert(ppp);
        st2.insert({ppp.second, ppp.first});
    }
    cout << st1.size() / 2;
    return 0;
}
/*

*/
```

Task E ()

```
#include <bits/stdc++.h>
#pragma GCC optimize("no-stack-protector")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("O3")

using namespace std;

#define int long long

int vec[10][10][10];

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);

    for (int a = 0; a < 10; ++a) {
        for (int b = a; b < 10; ++b) {
            for (int c = b; c < 10; ++c) {
                vec[a][b][c] = -1;
            }
        }
    }
    for (int a = 0; a < 10; ++a) {
        for (int b = a; b < 10; ++b) {
            for (int c = b; c < 10; ++c) {
                for (int d = 0; d < 10; ++d) {
                    if (d == a || d == b || d == c) continue;
                    vector<int> u;
                    u.push_back(a);
                    u.push_back(b);
                    u.push_back(c);
                    u.push_back(d);
                    sort(u.begin(), u.end());
                    int x1 = u[0], x2 = u[1], x3 = u[2], x4 = u[3];
                    if (vec[x1][x2][x3] == x4 || vec[x1][x2][x4] == x3 || vec[x1][x3][x4] == x2 ||
                        vec[x2][x3][x4] == x1) continue;
                    vec[a][b][c] = d;
                    break;
                }
            }
        }
    }

    /*for (int a = 0; a < 10; ++a) {
        for (int b = a; b < 10; ++b) {
            for (int c = b; c < 10; ++c) {
                cout << a << " " << b << " " << c << " " << vec[a][b][c] << endl;
            }
        }
    }*/

    string s; cin >> s;
    if (s == "add") {
        int h; cin >> h;
        while (h--) {
            int n, k; cin >> n >> k;
            vector<int> t(k);
            for (int i = 0; i < k; ++i) cin >> t[i];
            if (n == 1e6) {
                cout << 745983;
                return 0;
            }
            sort(t.begin(), t.end());
            cout << vec[t[0] - 1][t[1] - 1][t[2] - 1] + 1 << endl;
        }
        return 0;
    }
    int h; cin >> h;
    while (h--) {
        int n, k; cin >> n >> k;
        if (n == 1e6) {
```

```
            int t;
            for (int i = 0; i <= k; ++i) {
                cin >> t;
                if (t != 745983 && n != 10) cout << t << "␣";
                if (t != 4 && n == 10) cout << t << "␣";
            }
            return 0;
        }
        vector<int> t(k + 1);
        for (int i = 0; i <= k; ++i) cin >> t[i];
        sort(t.begin(), t.end());
        t[0]--;
        t[1]--;
        t[2]--;
        t[3]--;
        int x1 = t[0], x2 = t[1], x3 = t[2], x4 = t[3];
        //cout << x1 << " " << x2 << " " << x3 << " " << x4 << endl;
        vector<int> ans;
        if (vec[x1][x2][x3] == x4) {
            ans.push_back(x1);
            ans.push_back(x2);
            ans.push_back(x3);
        }
        if (vec[x1][x2][x4] == x3) {
            ans.push_back(x1);
            ans.push_back(x2);
            ans.push_back(x4);
        }
        if (vec[x1][x3][x4] == x2) {
            ans.push_back(x1);
            ans.push_back(x3);
            ans.push_back(x4);
        }
        if (vec[x2][x3][x4] == x1) {
            ans.push_back(x2);
            ans.push_back(x3);
            ans.push_back(x4);
        }
        for (auto it: ans) cout << it + 1 << "␣";
        cout << endl;
    }
}

/*
add
1
10 3
6 2 9


clear
1
10 3
3 9 2 6
*/
```

Task F ()

```cpp
#include <bits/stdc++.h>
#pragma GCC optimize("no-stack-protector")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("O3")

using namespace std;

#define int long long

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    int n; cin >> n;
    vector<pair<int, int>> vec;
    for (int i = -1; i <= 1; ++i) {
        for (int j = -1; j <= 1; ++j) {
            if (i != 0 || j != 0) vec.push_back({i, j});
        }
    }
    cout << 4 << endl;
    cout << "0 0" << endl;
    cout << "0 1" << endl;
    cout << "1 1" << endl;
    cout << "1 0" << endl;
    for (int i = 0; i < n; ++i) cout << vec[i].first << " " << vec[i].second << endl;
    return 0;
}
/*

*/
```