

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 <iostream>
#define int long long
using namespace std;

signed main() {
    int k;
    cin >> k;

    if (k <= 10) cout << k % 10 << '\n';
    else if ((k - 10) % 9 == 0) cout << 0 << '\n';
    else cout << (k - 10) % 9 + 1 << '\n';
}
```

Task B ()

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int n, k;
    cin >> n >> k;
    string s;
    cin >> s;

    int cnt = 0;
    int left = 0, right = 1;
    vector<int> counter(26);

    auto check = [&](int i) -> bool {
        counter[s[i] - 'a']++;
        int c = 0;
        for (auto i : counter)
            if (i > 0)
                c++;
        return c <= 3;
    };

    while (left < n) {
        counter[s[left] - 'a']++;
        while (right - left < k && right < n)
            if (check(right)) right++;
            else break;
        fill(counter.begin(), counter.end(), 0);
        cnt++;
        left = right;
        right++;
    }

    cout << cnt << '\n';
}
```

Task C ()

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

const int inf = 1e9, N = 500, M = 250000;
int dp[N + 1][M + 1];

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

    for (int i = 0; i <= N; ++i)
        for (int j = 0; j <= M; ++j)
            dp[i][j] = inf;

    dp[0][0] = 0;

    int n, x, y;
    cin >> n >> x >> y;
    vector<int> v(n), w(n);

    for (auto& i : v) cin >> i;
    for (auto& i : w) cin >> i;

    for (int i = 1; i <= n; ++i)
        for (int j = 0; j <= x; ++j) {
            dp[i][j] = dp[i - 1][j] + w[i - 1];
            if (j >= v[i - 1])
                dp[i][j] = min(dp[i][j], dp[i - 1][j - v[i - 1]]);
        }

    int res = inf;
    for (int i = 0; i <= x; ++i)
        res = min(res, dp[n][i]);
    if (res > y) cout << -1 << '\n';
    else {
        int s = -1;
        for (int i = 0; i <= x; ++i)
            if (dp[n][i] <= y) {
                s = i;
                break;
            }

        string ans = "";
        for (int i = n; i > 0; --i)
            if (dp[i][s] == dp[i - 1][s] + w[i - 1])
                ans += 'y';
            else {
                s -= v[i - 1];
                ans += 'x';
            }
        reverse(ans.begin(), ans.end());
        cout << ans << '\n';
    }
}
```

Task D ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <stack>
using namespace std;

int check(const vector<int>& v) {
    stack<int> s;
    for (auto i : v)
        if (s.empty() || s.top() != i)
            s.push(i);
        else
            s.pop();
    return s.size();
}

int main() {
    int n;
    cin >> n;
    vector<int> v(2 * n);

    for (int i = 0; i < 2 * n; ++i) {
        char c;
        cin >> c;
        if (c == '[' || c == ']') v[i] = 0;
        else v[i] = 1;
    }

    cout << check(v) / 2 << '\n';
}
```

Task E ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#define int long long
using namespace std;

set<set<int>> tetr;
set<set<int>> triple;

const int h = 44953;
const int p = 999983, s = 31;
vector<int> nums = { 3, 5, 7, 11, 13, 17, 19, 23, 29, 31 };

void new_solve_add(int n, int k) {
    vector<int> a(k);
    for (auto& i : a) cin >> i;

    sort(a.begin(), a.end());
    int res = 0;
    for (int i = 0; i < k; ++i) {
        res = (res * s + a[i]) % p;
    }
    cout << res % n + 1 << '\n';
}

void new_solve_clear(int n, int k) {
    vector<int> a(k + 1);
    for (auto& i : a) cin >> i;

    sort(a.begin(), a.end());
    for (int i = 0; i <= k; ++i) {
        int res = 0;
        for (int j = 0; j <= k; ++j)
            if (j != i)
                res = (res * s + a[j]) % p;
        if (a[i] == res % n + 1) {
            for (int j = 0; j <= k; ++j)
                if (a[i] != a[j])
                    cout << a[j] << ' ';
            cout << '\n';
            return;
        }
    }
}

int power(int a, int n) {
    if (n == 0) return 1;
    if (n & 1) return (a * power(a, n - 1)) % h;
    int r = power(a, n / 2);
    return r * r % h;
}

void solve_add(int n, int k) {
    vector<int> a(k);
    for (auto& i : a) cin >> i;

    int res = 1;
    for (auto i : a)
        res = (res * i) % h;
    cout << res << '\n';
}

void solve_clear(int n, int k) {
    vector<int> a(k + 1);
    for (auto& i : a) cin >> i;

    int mult = 1;
    for (auto i : a)
        mult = (mult * i) % h;

    int x = -1;
```

```

    for (int i = 0; i <= k; ++i) {
        int r = mult * power(a[i], h - 2) % h;
        if (r == a[i]) {
            x = i;
            break;
        }
    }

    for (int i = 0; i <= k; ++i)
        if (i != x)
            cout << a[i] << ' ';

    cout << '\n';
}

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

    for (int i = 1; i <= 7; ++i)
        for (int j = i + 1; j <= 8; ++j)
            for (int k = j + 1; k <= 9; ++k)
                for (int q = k + 1; q <= 10; ++q)
                    tetr.insert(set<int>({ i, j, k, q }));

    for (int i = 1; i <= 8; ++i)
        for (int j = i + 1; j <= 9; ++j)
            for (int k = j + 1; k <= 10; ++k)
                triple.insert(set<int>({ i, j, k }));

    vector<int> used(210, 0);
    vector<set<int>> tr;
    for (auto i : tetr)
        tr.push_back(i);

    for (auto t : triple) {
        for (int i = 0; i < 210; ++i)
            if (!used[i]) {
                int flag = 1;
                for (auto j : t)
                    if (tr[i].find(j) == tr[i].end())
                        flag = 0;

                if (flag) {
                    for (auto j : tr[i])
                        if (t.find(j) == t.end()) {
                            used[i] = j;
                            break;
                        }
                    break;
                }
            }
    }

    }

    /*for (int i = 0; i < 210; ++i) {
        cout << "{ ";
        for (auto j : tr[i])
            cout << j << ' ';
        cout << "} : ";
        cout << used[i] << '\n';
    }*/

    string move;
    cin >> move;
    if (move == "add") {
        int q;
        cin >> q;
        while (q--) {
            int n, k;
            cin >> n >> k;
            if (n == 10) {
                int a, b, c;
                cin >> a >> b >> c;

                for (int i = 0; i < 210; ++i)
                    if (used[i] && used[i] != a && used[i] != b && used[i] !=

```

```

        c) {
            int flag = (tr[i].find(a) != tr[i].end())
                        && (tr[i].find(b) != tr[i].end())
                        && (tr[i].find(c) != tr[i].end());
            if (flag) {
                cout << used[i] << '\n';
                break;
            }
        }
    }
    else if (n == 1000000) {
        new_solve_add(n, k);
    }
    else {
        solve_add(n, k);
    }
}
else {
    int q;
    cin >> q;
    while (q--) {
        int n, k;
        cin >> n >> k;
        if (n == 10) {
            int a, b, c, d;
            cin >> a >> b >> c >> d;
            set<int> t({ a, b, c, d });
            for (int i = 0; i < 210; ++i)
                if (used[i] && t == tr[i]) {
                    for (auto j : t)
                        if (j != used[i])
                            cout << j << ' ';
                    cout << '\n';
                    break;
                }
        }
        else if (n == 1000000) {
            new_solve_clear(n, k);
        }
        else {
            solve_clear(n, k);
        }
    }
}
}

```

Task F ()

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int main() {
    int n;
    cin >> n;

    cout << 4 << '\n';
    cout << -1 << '\n' << -1 << '\n';
    cout << -1 << '\n' << 1 << '\n';
    cout << 1 << '\n' << 1 << '\n';
    cout << 1 << '\n' << -1 << '\n';

    if (n == 1) cout << 0 << '\n' << 2 << '\n';
    else if (n == 2) {
        cout << 0 << '\n' << 2 << '\n';
        cout << 0 << '\n' << -2 << '\n';
    }
    else if (n == 3) {
        cout << 0 << '\n' << 2 << '\n';
        cout << 0 << '\n' << -2 << '\n';
        cout << 2 << '\n' << 0 << '\n';
    }
    else if (n == 4) {
        cout << 0 << '\n' << 2 << '\n';
        cout << 0 << '\n' << -2 << '\n';
        cout << 2 << '\n' << 0 << '\n';
        cout << -2 << '\n' << 0 << '\n';
    }
    else if (n == 5) {
        cout << 1 << '\n' << 2 << '\n';
        cout << -1 << '\n' << 2 << '\n';
        cout << 0 << '\n' << -2 << '\n';
        cout << 2 << '\n' << 0 << '\n';
        cout << -2 << '\n' << 0 << '\n';
    }
    else if (n == 6) {
        cout << 1 << '\n' << 2 << '\n';
        cout << -1 << '\n' << 2 << '\n';
        cout << 1 << '\n' << -2 << '\n';
        cout << -1 << '\n' << -2 << '\n';
        cout << 2 << '\n' << 0 << '\n';
        cout << -2 << '\n' << 0 << '\n';
    }
    else if (n == 7) {
        cout << 2 << '\n' << 2 << '\n';
        cout << -2 << '\n' << 2 << '\n';
        cout << 0 << '\n' << 2 << '\n';
        cout << 1 << '\n' << -2 << '\n';
        cout << -1 << '\n' << -2 << '\n';
        cout << 2 << '\n' << 0 << '\n';
        cout << -2 << '\n' << 0 << '\n';
    }
    else if (n == 8) {
        cout << 2 << '\n' << 2 << '\n';
        cout << -2 << '\n' << 2 << '\n';
        cout << 0 << '\n' << 2 << '\n';
        cout << 2 << '\n' << -2 << '\n';
        cout << -2 << '\n' << -2 << '\n';
        cout << 0 << '\n' << -2 << '\n';
        cout << 2 << '\n' << 0 << '\n';
        cout << -2 << '\n' << 0 << '\n';
    }
}
```