

# 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>
#include <vector>
using namespace std;

int main() {
    int k;
    cin >> k;
    if (k <= 10) {
        cout << k % 10;
    }
    else {
        k -= 11;
        vector<int>a = { 2,3,4,5,6,7,8,9,0 };
        cout << a[k % 9];
    }
    return 0;
}
```

## Task B ()

```
#include <iostream>
#include <vector>
#include <set>
#include <string>
using namespace std;

int main(){
    int n, k;
    cin >> n >> k;
    string s;
    cin >> s;
    set<char> used;
    int ans = 0;
    int len = 0;
    for (int i = 0; i < n; i++) {
        if (used.size() == 3 && used.count(s[i]) == 0) {
            ans++;
            used.clear();
            used.insert(s[i]);
            len = 1;
            continue;
        }
        if (len == k) {
            ans++;
            used.clear();
            used.insert(s[i]);
            len = 1;
            continue;
        }
        used.insert(s[i]);
        len++;
    }
    if (used.size()) ans++;
    cout << ans;
    return 0;
}
```

## Task C ()

```
#include <iostream>
#include <vector>
#include <set>
#include <string>
#include <algorithm>
#include <bitset>
#include <memory.h>
using namespace std;

const int W = 250100;
int dp[W];

int VV[1000];
int WW[1000];
bitset<501> pa[W];

int inf;

int main(){
    memset(dp, 63, sizeof(dp));
    inf = dp[0];
    dp[0] = 0;
    int n, x, y;
    cin >> n >> x >> y;
    for (int i = 0; i < n; i++) {
        cin >> VV[i];
    }
    for (int i = 0; i < n; i++) {
        cin >> WW[i];
    }
    for (int i = 0; i < n; i++) {
        int v = VV[i], w = WW[i];
        for (int ww = W - 1; ww >= 0; ww--) {
            dp[ww] = dp[ww] + w;
            pa[ww][i] = 1;
            if (dp[ww - v] != inf && ww - v >= 0 && dp[ww] > dp[ww - v]) {
                dp[ww] = dp[ww - v];
                pa[ww] = pa[ww - v];
                pa[ww][i] = 0;
            }
        }
    }
    for (int ww = 0; ww <= x; ww++) {
        if (dp[ww] <= y) {
            for (int i = 0; i < n; i++) {
                if (pa[ww][i]) {
                    cout << "y";
                }
                else {
                    cout << "x";
                }
            }
            return 0;
        }
    }
    cout << -1;
    return 0;
}
```

## Task D ()

```
#include <iostream>
#include <vector>
#include <set>
#include <string>
#include <algorithm>
#include <bitset>
#include <memory.h>
#include <stack>
#include <deque>
using namespace std;

bool same_type(char a, char b) {
    if (a == '(' && b == ')') {
        return true;
    } if (b == '(' && a == ')') {
        return true;
    } if (a == '[' && b == ']') {
        return true;
    } if (a == '[' && b == '[') {
        return true;
    }
    return false;
}

char same_open(char c) {
    if (c == ']') return '[';
    else return '(';
}

int main(){
    int n;
    cin >> n;
    n *= 2;
    deque<char>st;
    string s;
    cin >> s;
    for (int i = 0; i < n; i++) {
        if (s[i] == ']') s[i] = '[';
        if (s[i] == ')') s[i] = '(';
    }
    int ans = 0;
    for (int i = 0; i < n; i++) {
        if (st.size()) {
            if (s[i] == st.back()) {
                st.pop_back();
            }
            else {
                st.push_back(s[i]);
            }
        }
        else {
            st.push_back(s[i]);
        }
    }
    while (st.size()) {
        if (st.front() != st.back()) {
            ans++;
        }
        st.pop_back();
        st.pop_front();
    }
    cout << ans;
    return 0;
}
```

## Task E ()

```
#include <iostream>
#include <vector>
#include <assert.h>
#include <string>
#include <algorithm>
#include <map>
#include <random>
#include <set>
using namespace std;

typedef long long ll;

const int prime = 1e9 + 97;
const int bb = 37;
const int cc = 123412;

int bin_pow(int x, int y, int mod) {
    if (y == 0) return 1;
    int m = bin_pow(x, y / 2, mod);
    if (y % 2 == 0) {
        return (ll)m * m % mod;
    }
    else {
        return (ll)m * m % mod * (ll)x % mod;
    }
}

const int SEED = 239;

int nxt(int x) {
    return ((ll)x * bb + cc) % prime;
}

int get_for_del(int n, int k, vector<int>p, int t) {
    vector<int> tr;
    mt19937 rng(239 + t);
    set<int>pr;
    for (int i = 0; i < p.size(); i++) pr.insert(p[i]);
    for (int i = 0; i < 10000; i++) {
        tr.push_back(rng() % n);
    }
    for (int i = 0; i < tr.size(); i++) {
        if (!pr.count(tr[i]) && !pr.count(tr[i + 1])) {
            return tr[i];
        }
    }
}

int get_ans(int n, int k, vector<int>p, int t) {
    mt19937 rng(239 + t);
    set<int>pr;
    vector<int>tr;
    for (int i = 0; i < p.size(); i++) pr.insert(p[i]);
    for (int i = 0; i < 10000; i++) {
        tr.push_back(rng() % n);
    }
    for (int i = 0; i < tr.size(); i++) {
        if (pr.count(tr[i]) && !pr.count(tr[i + 1])) {
            return tr[i];
        }
    }
}

map<vector<int>, int> add;
map<vector<int>, int> kik;

map < vector<int>, vector<vector<int>>> g;

map<vector<int>, vector<int>> mt;
map<vector<int>, vector<int>> umt;
set<vector<int>>used;
```

```

bool dfs(vector<int>v) {
    if (used.count(v)) return false;
    used.insert(v);
    for (auto u : g[v]) {
        if (mt.count(u) == 0 || dfs(mt[u])) {
            mt[u] = v;
            umt[v] = u;
            return true;
        }
    }
    return false;
}

int main() {
    vector<vector<int>>> vs;
    for (int a = 1; a <= 10; a++) {
        for (int b = a + 1; b <= 10; b++) {
            for (int c = b + 1; c <= 10; c++) {
                vector<int>p1 = { a,b,c };
                vs.push_back(p1);
                for (int d = 10; d >= 1; d--) {
                    if (a != d && b != d && c != d) {
                        vector<int>p2 = { a,b,c,d };
                        sort(p2.begin(), p2.end());
                        g[p1].push_back(p2);
                    }
                }
            }
        }
    }

    for (auto v : vs) {
        used.clear();
        dfs(v);
    }

    string s;
    cin >> s;
    if (s == "add") {
        int t;
        cin >> t;
        for (; t; t--) {
            int n, k;
            cin >> n >> k;
            if (n == 10) {
                vector<int>p(k);
                set<int>q;
                for (int i = 0; i < k; i++) {
                    cin >> p[i];
                    q.insert(p[i]);
                }
                sort(p.begin(), p.end());
                for (int x : umt[p]) {
                    if (q.count(x) == 0) {
                        cout << x;
                    }
                }
                cout << "\n";
            }
            else {
                vector<int>p(k);
                for (int i = 0; i < k; i++) cin >> p[i];
                for (int i = 0; i < k; i++) p[i]--;
                sort(p.begin(), p.end());
                cout << get_for_del(n, k, p, t) + 1 << "\n";
            }
        }
    }
    else {
        int t;
        cin >> t;
        for (; t; t--) {

```

```

int n, k;
cin >> n >> k;
if (n == 10) {
    vector<int> p(k + 1);
    for (int i = 0; i < k + 1; i++) cin >> p[i];
    sort(p.begin(), p.end());
    for (auto x : mt[p]) {
        cout << x << "␣";
    }
    cout << '\n';
}
else {
    vector<int> pp(k + 1);
    for (int i = 0; i < k + 1; i++) cin >> pp[i];
    for (int i = 0; i < k + 1; i++) pp[i]--;
    sort(pp.begin(), pp.end());
    int xx = get_ans(n, k, pp, t);

    for (int x : pp) if (x != xx) cout << x + 1 << "␣";
    cout << "\n";
}
}
return 0;
}

```

## Task F ()

```
#include <iostream>
#include <vector>
#include <assert.h>
#include <string>
#include <algorithm>
#include <random>
#include <set>
using namespace std;

typedef long long ll;

int main() {
    int n;
    cin >> n;
    cout << 4 << "\n";
    cout << "0_0\n0_1\n1_1\n1_0\n";
    vector<pair<int, int>> v;
    v.push_back({ 1, 1 });
    v.push_back({ -1, -1 });
    v.push_back({ 1, -1 });
    v.push_back({ -1, 1 });
    v.push_back({ 0, 1 });
    v.push_back({ 1, 0 });
    v.push_back({ -1, 0 });
    v.push_back({ 0, -1 });

    for (int i = 0; i < n; i++) {
        cout << v[i].first << "_" << v[i].second << "\n";
    }
    return 0;
}
```