Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	100	25	525

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <random>
#include <set>
#include <map>
#include <bitset>
#include <array>
#include <stack>
#include <queue>
#include <cassert>

using namespace std;

typedef long long ll;
typedef long double ld;

const int N = 200010;
const int K = 10000010;
const int LOGN = 20;
const ll INF = 1000000007;

mt19937 gen(41);

int c[N];

signed main() {
        ios_base::sync_with_stdio(false);
        cin.tie(0);
#ifdef _DEBUG
        freopen("Test.txt", "r", stdin);
#endif

        int k;

        cin >> k;

        --k;

        cout << (k + 1 + (k - 1) / 9) % 10;

}
```

Task B ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <random>
#include <set>
#include <map>
#include <bitset>
#include <array>
#include <stack>
#include <queue>
#include <cassert>

using namespace std;

typedef long long ll;
typedef long double ld;

const int N = 200010;
const int K = 10000010;
const int LOGN = 20;
const ll INF = 1000000007;

mt19937 gen(41);

int dp[N];

set < char > m;

string s;

signed main() {
        ios_base::sync_with_stdio(false);
        cin.tie(0);
#ifdef _DEBUG
        freopen("Test.txt", "r", stdin);
#endif

        int n, k;

        cin >> n >> k;

        cin >> s;

        int ans = 1;

        int cnt = 0;

        for (int i = 0; i < n; ++i) {
                ++cnt;
                if ((m.size() == 3 && m.find(s[i]) == m.end()) || (cnt == k + 1)) {
                        ++ans;
                        m.clear();
                        m.insert(s[i]);
                        cnt = 1;
                }
                else {
                        m.insert(s[i]);
                }
        }

        cout << ans;

}
```

Task C ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <random>
#include <set>
#include <map>
#include <bitset>
#include <array>
#include <stack>
#include <queue>
#include <cassert>

using namespace std;

typedef long long ll;
typedef long double ld;

const int N = 501;
const int K = 250001;
const int LOGN = 20;
const ll INF = 1000000007;

mt19937 gen(41);

ll dp1[K], dp2[K];
int par[N][K];

ll v[N], w[N];

signed main() {
        ios_base::sync_with_stdio(false);
        cin.tie(0);
#ifdef _DEBUG
        freopen("Test.txt", "r", stdin);
#endif

        int n, x, y;

        cin >> n >> x >> y;

        for (int i = 1; i <= n; ++i) {
                cin >> v[i];
        }

        for (int i = 1; i <= n; ++i) {
                cin >> w[i];
        }

        fill(dp1, dp1 + K, INF);

        dp1[0] = 0;

        for (int i = 1; i <= n; ++i) {
                fill(dp2, dp2 + K, INF);
                for (int j = 0; j <= x; ++j) {
                        if (dp2[j] > dp1[j] + w[i]) {
                                dp2[j] = dp1[j] + w[i];
                                par[i][j] = j;
                        }
                        if (j + v[i] > x)
                                continue;
                        if (dp2[j + v[i]] > dp1[j]) {
                                dp2[j + v[i]] = dp1[j];
                                par[i][j + v[i]] = j;
                        }
                }
                for (int j = 0; j <= x; ++j) {
                        swap(dp1[j], dp2[j]);
                }
        }
        for (int j = 0; j <= x; ++j) {
                if (dp1[j] <= y) {
                        int ind = n;
                        string ans;
                        while (ind != 0) {
                                if (par[ind][j] == j)
                                        ans += 'y';
                                else
                                        ans += 'x';
                                j = par[ind][j];
                                --ind;
                        }
                        reverse(ans.begin(), ans.end());
                        cout << ans;
                        return 0;
                }
        }

        cout << -1;

}
```

Task D ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <random>
#include <set>
#include <map>
#include <bitset>
#include <array>
#include <stack>
#include <queue>
#include <cassert>

using namespace std;

typedef long long ll;
typedef long double ld;

const int N = 200010;
const int K = 10000010;
const int LOGN = 20;
const ll INF = 1000000007;

mt19937 gen(41);

int t[N];

string s;

signed main() {
        ios_base::sync_with_stdio(false);
        cin.tie(0);
#ifdef _DEBUG
        freopen("Test.txt", "r", stdin);
#endif

        int n;

        cin >> n;

        n *= 2;

        cin >> s;

        n = s.size();

        s = '0' + s;

        for (int i = 1; i <= n; ++i) {
                if (s[i] == '(' || s[i] == ')')
                        t[i] = 1;
                else
                        t[i] = 2;
        }

        vector < int > x;

        int cnt = 1;

        for (int i = 2; i <= n; ++i) {
                if (t[i] == t[i - 1]) {
                        ++cnt;
                }
                else {
                        if (cnt % 2 == 1) {
                                if (!x.empty() && x.back() == t[i - 1])
                                        x.pop_back();
                                else
                                        x.push_back(t[i - 1]);
                        }
                        cnt = 1;
                }
        }

        if (cnt % 2 == 1) {
                if (!x.empty() && x.back() == t[n])
                        x.pop_back();
                else
                        x.push_back(t[n]);
        }
        //cout << dp[1][n];

        //assert(dp[1][n] == max(0, (int)x.size() - 1));

        cout << max(0, (int)x.size() / 2);

}
```

Task E ()

```cpp
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <random>
#include <set>
#include <map>
#include <bitset>
#include <array>
#include <stack>
#include <queue>
#include <cassert>

using namespace std;

typedef long long ll;
typedef long double ld;

const int N = 200010;
const int K = 10000010;
const int LOGN = 20;
const ll INF = 1000000007;
const ll MOD = (1ll << 31ll) - 1ll;

mt19937 gen(41);

ll mod(ll x) {
        ll a = x & MOD;
        ll b = x >> 31ll;
        ll ans = a + b;
        while (ans >= MOD)
                ans -= MOD;
        return ans;
}

ll b = gen() % INF, a;

ll a1 = gen() % INF, b1 = gen() % INF;

ll f1(ll x) {
        return mod(a1 * x + b);
}

ll f(ll x) {
        return mod(a * f1(x) + b);
}

string type;

ll sdvig(ll x, ll d, ll n) {
        --x;
        x += d;
        x %= n;
        if (x < 0)
                x += n;
        return x + 1;
}

ll add[11][11][11];
ll clr[11][11][11][11];

signed main() {
        ios_base::sync_with_stdio(false);
        cin.tie(0);
#ifdef _DEBUG
        freopen("Test.txt", "r", stdin);
#endif

        int ar[4];

        for (ar[0] = 1; ar[0] <= 10; ++ar[0]) {
                for (ar[1] = ar[0] + 1; ar[1] <= 10; ++ar[1]) {
                        for (ar[2] = ar[1] + 1; ar[2] <= 10; ++ar[2]) {
                                for (ar[3] = 1; ar[3] <= 10; ++ar[3]) {
                                        vector < int > sar(4);
                                        set < int > s;
                                        for (int j = 0; j < 4; ++j)
                                                sar[j] = ar[j], s.insert(ar[j]);
                                        if (s.size() != 4)
                                                continue;
                                        sort(sar.begin(), sar.end());
                                        if (add[ar[0]][ar[1]][ar[2]] == 0 && clr[sar[0]][sar[1]][
                                            sar[2]][sar[3]] == 0) {
                                                add[ar[0]][ar[1]][ar[2]] = ar[3];
                                                clr[sar[0]][sar[1]][sar[2]][sar[3]] = ar[3];
                                        }
                                }
                        }
                }
        }

        cin >> type;

        if (type == "add") {

                int tests;

                cin >> tests;

                while (tests--) {

                        int n, k;
                        cin >> n >> k;
                        set < ll > s;
                        if (n == 1000000) {
                                for (int i = 1; i <= k; ++i) {
                                        ll x;
                                        cin >> x;
                                        s.insert(x);
                                }
                                cout << sdvig(1, -b, n) << '\n';
                        }
                        else if (n == 10) {
                                vector < int > ar;
                                for (int i = 1; i <= k; ++i) {
                                        ll x;
                                        cin >> x;
                                        ar.push_back(x);
                                }
                                sort(ar.begin(), ar.end());
                                cout << add[ar[0]][ar[1]][ar[2]] << '\n';
                        }
                        else {
                                ll sum = 0;
                                vector < int > ar;
                                for (int i = 1; i <= k; ++i) {
                                        ll x;
                                        cin >> x;
                                        ar.push_back(x);
                                        sum += x;
                                        sum %= k + 1;
                                }

                                sort(ar.begin(), ar.end());

                                bool ok = false;

                                for (int i = 1; i <= n; ++i) {
                                        int ind = lower_bound(ar.begin(), ar.end(), i) - ar.begin
                                            ();
                                        if (ind == ar.size() || ar[ind] != i) {
                                                if ((sum + i) % (k + 1) == ind) {
                                                        ok = true;
                                                        cout << i << '\n';
                                                        break;
                                                }
                                        }
                                }
                                assert(ok);
                        }
                }
        }
        else {
                int tests;

                cin >> tests;

                while (tests--) {
                        int n, k;
                        cin >> n >> k;
                        set < ll > s;
                        if (n == 1000000) {
                                int res;
                                for (int i = 1; i <= k + 1; ++i) {
                                        ll x;
                                        cin >> x;
                                        s.insert(x);
                                        if (sdvig(x, b, n) == 1) {
                                                res = x;
                                        }
                                }
                                s.erase(res);
                                for (ll x : s)
                                        cout << x << '␣';
                                cout << '\n';
                        }
                        else if (n == 10) {
                                set < int > s;
                                vector < int > ar;
                                for (int i = 1; i <= k + 1; ++i) {
                                        ll x;
                                        cin >> x;
                                        s.insert(x);
                                        ar.push_back(x);
                                }
                                sort(ar.begin(), ar.end());
                                s.erase(clr[ar[0]][ar[1]][ar[2]][ar[3]]);
                                for (ll x : s)
                                        cout << x << '␣';
                                cout << '\n';
                        }
                        else {
                                ll sum = 0;
                                vector < int > ar;
                                for (int i = 1; i <= k + 1; ++i) {
                                        ll x;
                                        cin >> x;
                                        ar.push_back(x);
                                        sum += x;
                                        sum %= k + 1;
                                }

                                sort(ar.begin(), ar.end());

                                ar.erase(ar.begin() + sum);

                                for (ll x : ar)
                                        cout << x << '␣';

                                cout << '\n';

                        }
                }
        }

}
```

Task F ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
#include <random>
#include <set>
#include <map>
#include <bitset>
#include <array>
#include <stack>
#include <queue>
#include <cassert>

using namespace std;

typedef long long ll;
typedef long double ld;

const int N = 200010;
const int K = 10000010;
const int LOGN = 20;
const ll INF = 1000000007;
const ll MOD = (1ll << 31ll) - 1ll;

mt19937 gen(41);

signed main() {
        ios_base::sync_with_stdio(false);
        cin.tie(0);
#ifdef _DEBUG
        freopen("Test.txt", "r", stdin);
#endif

        int n;

        cin >> n;

        /*if (n == 1) {
                cout << "4\n";
                cout << "1 1\n1 -1\n-1 -1\n-1 1\n";
                cout << "2 2";
                return 0;
        }
        if (n == 2) {
                cout << "4\n";
                cout << "1 1\n1 -1\n-1 -1\n-1 1\n";
                cout << "2 2\n-2 2";
                return 0;
        }
        if (n == 3) {
                cout << "4\n";
                cout << "1 1\n1 -1\n-1 -1\n-1 1\n";
                cout << "2 2\n-2 2\n-2 -2";
                return 0;
        }
        if (n == 4) {
                cout << "4\n";
                cout << "1 1\n1 -1\n-1 -1\n-1 1\n";
                cout << "2 0\n-2 0\n0 -2\n0 2";
                return 0;
        }*/

        vector < pair <int, int > > res;

        for (int i = -1; i <= 1; ++i) {
                for (int j = -1; j <= 1; ++j) {
                        if (i == 0 && j == 0)
                                continue;
                        res.push_back({ 2 * i, 2 * j });
                }
        }
        cout << "4\n";
        cout << "1␣1\n1␣-1\n-1␣-1\n-1␣1\n";

        for (int i = 1; i <= n; ++i)
                cout << res[i - 1].first << '␣' << res[i - 1].second << '\n';

}
```