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;

using ll = long long;
using pll = pair<ll, ll>;
using pii = pair<int, int>;
using ld = long double;
using vii = vector<int>;
using vll = vector<ll>;

const int N = 5e5 + 5;
const int K = 20;
const int MAX = 1e6 + 5;
const ll INF = 1e18 + 5;
const ll MOD = 1e9 + 7;
const ld eps = 1e-9;

void optimize() {
    cin.tie(0);
    ios::sync_with_stdio(false);
}

int a[] = {0, 2, 3, 4, 5, 6, 7, 8, 9};

int main() {
    optimize();

    ll k;
    cin >> k;
    if (k < 10) {
        cout << k % 10 << endl;
        exit(0);
    }
    k -= 10;
    k %= 9;

    cout << a[k] << endl;

    return 0;
}
```

Task B ()

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
using pll = pair<ll, ll>;
using pii = pair<int, int>;
using ld = long double;
using vii = vector<int>;
using vll = vector<ll>;

const int N = 5e5 + 5;
const int K = 20;
const int MAX = 1e6 + 5;
const ll INF = 1e18 + 5;
const ll MOD = 1e9 + 7;
const ld eps = 1e-9;

void optimize() {
    cin.tie(0);
    ios::sync_with_stdio(false);
}


int main() {
    optimize();

    int n, k;
    cin >> n >> k;
    string s; cin >> s;
    int ans = 1;
    unordered_set<char> q;
    int sz = 0;
    for (int i = 0; i < n; ++i) {
        char c = s[i];
        if (sz == k) {
            sz = 1;
            ++ans;
            q.clear();
            q.insert(c);
            continue;
        }
        q.insert(c);
        if (q.size() > 3) {
            q.clear();
            q.insert(c);
            ++ans;
            sz = 1;
        }
        else {
            ++sz;
        }
    }

    cout << ans << endl;

    return 0;
}
```

Task C ()

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
using pll = pair<ll, ll>;
using pii = pair<int, int>;
using ld = long double;
using vii = vector<int>;
using vll = vector<ll>;

const int N = 501;
const int K = 20;
const int MAX = 1e6 + 5;
const ll INF = 1e18 + 5;
const ll MOD = 1e9 + 7;
const ld eps = 1e-9;

void optimize() {
    cin.tie(0);
    ios::sync_with_stdio(false);
}

int v[N], w[N];

int d[2][250001];
int p[N][250001];
bitset<N> ans;

int main() {
    optimize();

    int n;
    ll x, y;
    cin >> n >> x >> y;
    ll sw = 0;
    for (int i = 0; i < n; ++i) {
        cin >> v[i];
    }
    for (int i = 0; i < n; ++i) {
        cin >> w[i];
        sw += w[i];
    }

    ll w1 = sw - y;

    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= x; ++j) {
            d[1][j] = d[0][j];
            p[i][j] = -1;
            if (v[i - 1] <= j && d[0][j - v[i - 1]] + w[i - 1] > d[1][j]) {
                p[i][j] = i - 1;
                d[1][j] = d[0][j - v[i - 1]] + w[i - 1];
            }
        }
        for (int j = 1; j <= x; ++j) {
            d[0][j] = d[1][j];
        }
    }

    if (d[0][x] < w1) {
        cout << -1 << endl;
        exit(0);
    }

    int i = n, j = x;
    while (i > 0 && j > 0) {
        if (p[i][j] == -1) {
            --i;
            continue;
        }
        int num = p[i][j];
        ans[num] = 1;
        --i;
        j -= v[num];
    }

    for (int i = 0; i < n; ++i) {
        cout << (ans[i] ? "x" : "y");
    }
    cout << endl;

    return 0;
}
```

Task D ()

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
using pll = pair<ll, ll>;
using pii = pair<int, int>;
using ld = long double;
using vii = vector<int>;
using vll = vector<ll>;

const int N = 5e5 + 5;
const int K = 20;
const int MAX = 1e6 + 5;
const ll INF = 1e18 + 5;
const ll MOD = 1e9 + 7;
const ld eps = 1e-9;

void optimize() {
    cin.tie(0);
    ios::sync_with_stdio(false);
}

int main() {
    optimize();

    vector<int> st;

    int n;
    cin >> n;
    string s; cin >> s; n *= 2;
    for (int i = 0; i < n; ++i) {
        int type = (s[i] == ')' || s[i] == '(' ? 1 : 2);
        if (!st.empty() && st.back() == type) {
            st.pop_back();
            continue;
        }
        st.push_back(type);
    }

    cout << st.size() / 2 << endl;

    return 0;
}
```

Task E ()

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
using pll = pair<ll, ll>;
using pii = pair<int, int>;
using ld = long double;
using vii = vector<int>;
using vll = vector<ll>;

const int N = 1e5 + 5;
const int K = 20;
const int MAX = 1e6 + 5;
const ll INF = 1e18 + 5;
const ll MOD = 1e9 + 7;
const ld eps = 1e-9;

const int T = 874194;
const int EMPTY = 7e4;
const int NN = 1e5;

void optimize() {
    cin.tie(0);
    ios::sync_with_stdio(false);
}

bool get(int mask, int bit) {
    return mask & (1 << bit);
}
unordered_map<int, int> par;
unordered_map<int, int> inv_par;

bitset<N> arr;

void add() {
    int n, k; cin >> n >> k;
    if (n == 10) {
        unordered_set<int> s;
        int msk = 0;
        for (int i = 0; i < k; ++i) {
            int x; cin >> x;
            s.insert(x - 1);
            msk += (1 << (x - 1));
        }
        int msk1 = par[msk];
        for (int i = 0; i < n; ++i) {
            if (!s.count(i) && get(msk1, i)) {
                cout << i + 1 << '\n';
                return;
            }
        }
        exit(-1);
    }
    if (n == NN) {
        arr.reset();
        ll s = 0;
        for (int i = 0; i < k; ++i) {
            int x; cin >> x;
            s += x;
            arr[x - 1] = 1;
        }
        //s += 325252;
        s %= EMPTY;
        int num = 0;
        for (int i = 0; i < n; ++i) {
            if (arr[i])
                continue;
            if (num == s) {
                cout << i + 1 << '\n';
                return;
            }
            ++num;
        }
    }
    cout << T << endl;
}

void clear() {
    int n, k; cin >> n >> k;
    if (n == 10) {
        unordered_set<int> s;
        int msk = 0;
        for (int i = 0; i < k + 1; ++i) {
            int x; cin >> x;
            s.insert(x - 1);
            msk += (1 << (x - 1));
        }
        int msk1 = inv_par[msk];
        for (int i = 0; i < n; ++i) {
            if (get(msk1, i)) {
                cout << i + 1 << "␣";
            }
        }
        cout << "\n";
        return;
    }
    if (n == NN) {
        arr.reset();
        ll s = 0;
        for (int i = 0; i < k + 1; ++i) {
            int x; cin >> x;
            s += x;
            arr[x - 1] = 1;
        }
        //s += 325252;
        //s %= 70000;
        int num = 0;
        for (int i = 0; i < n; ++i) {
            if (arr[i] == 0) {
                ++num;
                continue;
            }
            if ((s - i - 1) % EMPTY == num) {
                for (int j = 0; j < n; ++j) {
                    if (arr[j] == 0 || j == i)
                        continue;
                    cout << j + 1 << "␣";
                }
                cout << '\n';
                return;
            }
        }
        exit(-1);
    }
    for (int i = 0; i < k + 1; ++i) {
        int x; cin >> x;
        if (x != T)
            cout << x << "␣";
    }
    cout << endl;
}

bool used[1024];
int connect[1024];
bool f = false;
void go(int t) {
    if (f)
        return;
    if (t == 1024) {
        f = true;
        for (int i = 0; i < t; ++i) {
//            cout << i << ": " << connect[i] << endl;
            par[i] = connect[i];
            inv_par[connect[i]] = i;
        }
        return;
    }
    set<int> pos;
    for (int i = 0; i < 10; ++i) {
        if (get(t, i))
            pos.insert(i);
    }
    if (pos.size() != 3) {
        go(t + 1);
        return;
    }
    for (int i = 0; i < 10; ++i) {
        if (pos.count(i))
            continue;
        if (used[t + (1 << i)])
            continue;
        used[t + (1 << i)] = true;
        connect[t] = t + (1 << i);
        go(t + 1);
        used[t + (1 << i)] = false;
    }
}

void gen_partition() {
    go(0);
}

int main() {
    optimize();

    gen_partition();

    string s;
    cin >> s;
    int t; cin >> t;
    if (s == "add") {
        for (int i = 0; i < t; ++i)
            add();
    }
    else {
        for (int i = 0; i < t; ++i)
            clear();
    }

    return 0;
}
```

Task F ()

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
using pll = pair<ll, ll>;
using pii = pair<int, int>;
using ld = long double;
using vii = vector<int>;
using vll = vector<ll>;

const int N = 501;
const int K = 20;
const int MAX = 1e6 + 5;
const ll INF = 1e18 + 5;
const ll MOD = 1e9 + 7;
const ld eps = 1e-9;

void optimize() {
    cin.tie(0);
    ios::sync_with_stdio(false);
}

map<ll, set<ll>> xs, ys;

int main() {
    optimize();

    int n;

    cin >> n;

    cout << 4 << endl;
    cout << 0 << "␣" << 0 << endl;
    cout << 0 << "␣" << 1 << endl;
    cout << 1 << "␣" << 1 << endl;
    cout << 1 << "␣" << 0 << endl;

    int dx[] = {0, 1, 1, 1, 0, -1, -1, -1};
    int dy[] = {1, 1, 0, -1, -1, -1, 0, 1};
    for (int i = 0; i < n; ++i) {
        cout << dx[i] << "␣" << dy[i] << endl;
    }

    return 0;
}
```