Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	55	25	480

Task A ()

```
k = int(input())

n = 10**1000

if (k <= 10):
        print(k % 10)
else:
        if (k >= n):
                q = n * 2 - k
                print(q % 10)
        else:
                k -= 10
                k %= 9
                if (k == 0):
                        print(0)
                else:
                        print(k + 1)
```

Task B ()

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

#define ll long long
#define pb push_back
#define pii pair<int, int>

using namespace std;

int n, k;
string s;

signed main() {
        ios::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        cin >> n >> k;
        cin >> s;

        int ans = 0;
        int len = 0;
        set<char> t;

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

                if (len == k) {
                        ans++;
                        t.clear();
                        len = 0;
                }
        }
        if (len > 0) ans++;
        cout << ans << "\n";
        return 0;
}
```

Task C ()

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

#define ll long long
#define pb push_back
#define pii pair<int, int>

using namespace std;

const int inf = 1e9;

int n, x, y, c_sum = 0;
vector<int> w, c, ans;  // max cnt of coins
vector<vector<int>> dp;
 // p;

void read() {
        cin >> n >> x >> y;
        w.resize(n);
        c.resize(n);
        for (int i = 0; i < n; i++) {
                cin >> w[i];
        }
        for (int i = 0; i < n; i++) {
                cin >> c[i];
                c_sum += c[i];
        }
}

void relax(int &a, int b) {
        a = max(a, b);
}

void solve() {
        // rukzak
        dp.assign(n + 1, vector<int>(x + 1, -inf));
        // p.assign(n + 1, vector<int>(x + 1, -1));
        dp[0][0] = 0;
        for (int i = 0; i < n; i++) {
                dp[i + 1] = dp[i];
                for (int j = x; j >= 0; j--) {
                        if (dp[i][j] == -inf) continue;
                        if (j + w[i] <= x) {
                                if (dp[i][j] + c[i] > dp[i + 1][j + w[i]]) {
                                        dp[i + 1][j + w[i]] = dp[i][j] + c[i];
                                        // p[i + 1][j + w[i]] = i;
                                }
                        }
                }
        }
        ans.assign(n, 0);
        // for (auto e : dp) cout << e << " "; cout << endl;
        for (int i = 0; i <= x; i++) {
                if ((dp[n][i] != -inf) && ((c_sum - dp[n][i]) <= y)) {
                        int t = i;
                        for (int j = n; j >= 1; j--) {
                                if (dp[j - 1][t] == dp[j][t]) {

                                } else {
                                        ans[j - 1] = 1;
                                        t -= w[j - 1];
                                }
                        }
                        for (auto e : ans) {
                                if (e) {
                                        cout << "x";
                                } else {
                                        cout << "y";
                                }
                        }
                        cout << "\n";
                        return;
                //      int t = i;
                //      while (t > 0) {
                //              ans[p[t]] = 1;
                //              t -= w[p[t]];
                //      }
                //      for (auto e : ans) {
                //              if (e) {
                //                      cout << "x";
                //              } else {
                //                      cout << "y";
                //              }
                //      }
                //      cout << "\n";
                //      return;
                }
        }
        cout << "-1\n";
}

void write() {

}

signed main() {
        ios::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        read();
        solve();
        write();
        return 0;
}
```

Task D ()

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

#define ll long long
#define pb push_back
#define pii pair<int, int>

using namespace std;

int n;
string s;
string q;

void solve() {
        cin >> n >> s;
        q.clear();
        for (auto e : s) {
                if (e == ')') e = '(';
                if (e == ']') e = '[';
                if ((q.size() > 0) && (q.back() == e)) {
                        q.pop_back();
                } else {
                        q.pb(e);
                }
        }
        cout << q.size() / 2 << "\n";
}

signed main() {
        ios::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        // int t;
        // cin >> t;
        // while (t--) {
        solve();
        // }
        return 0;
}
```

Task E ()

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

#define int long long
#define pb push_back
#define pii pair<int, int>

using namespace std;

int n;
vector<vector<int>> g;
vector<int> used, mt, to;

int khyn(int v) {
        if (used[v]) return 0;
        used[v] = 1;
        for (auto e : g[v]) {
                if (mt[e] == -1 || khyn(mt[e])) {
                        mt[e] = v;
                        return 1;
                }
        }
        return 0;
}

void solve_2_clear(int n, int k) {
        k++;
        vector<int> a;
        a.resize(k);
        for (auto &e : a) cin >> e;

        int mask = 0;
        for (auto e : a) {
                mask |= (1 << (e - 1));
        }
        // cout << "mask = " << mask << endl;

        int o = mt[mask];
        // cout << "o = " << o << endl;

        int h = o ^ mask;
        int del = -1;
        for (int i = 0; i < 10; i++) {
                if ((1 << i) == h) {
                        del = i + 1;
                        break;
                }
        }
        for (auto e : a) {
                if (e != del) cout << e << "␣";
        }
        cout << "\n";
}

void solve_2_add(int n, int k) {
        vector<int> a;
        a.resize(k);
        for (auto &e : a) cin >> e;

        int mask = 0;
        for (auto e : a) {
                mask |= (1 << (e - 1));
        }
        // cout << "mask = " << mask << endl;

        int o = to[mask];
        // cout << "o = " << o << endl;

        int h = o ^ mask;
        for (int i = 0; i < 10; i++) {
                if ((1 << i) == h) {
                        cout << i + 1 << "\n";
                        break;
                }
        }
}


void solve_1_clear(int n, int k) {
        vector<int> a;
        a.resize(k + 1);
        for (auto &e : a) cin >> e;
        int sum = 0;
        for (auto e : a) sum += e;
        // int del = ((sum - 1) % 1000 + 1) / 2;
        int del = 987;
        for (auto e : a) {
                if (e != del) {
                        cout << e << "␣";
                }
        }
        cout << "\n";
}

void solve_1_add(int n, int k) {
        vector<int> a, b;
        a.resize(k);
        b.assign(1001, 0);
        for (auto &e : a) cin >> e;
        int sum = 0;
        for (auto e : a) sum += e;
        // cout << "sum = " << sum << endl;
        for (auto e : a) {
                b[(e - 1) % 1000 + 1] = 1;
        }
        cout << 987 << "\n";
        return;
        for (int i = 1; i <= 1000; i++) {
                if (((sum + i - 1) % 1000 + 1) == i * 2) {
                        cout << i << "\n";
                        break;
                }
        }
}

void solve_clear() {
        int n, k;
        cin >> n >> k;

        if (n == 1000000) {
                solve_1_clear(n, k);
        } else if (n == 10) {
                solve_2_clear(n, k);
        } else {

        }
}

void solve_add() {
        int n, k;
        cin >> n >> k;

        if (n == 1000000) {
                solve_1_add(n, k);
        } else if (n == 10) {
                solve_2_add(n, k);
        } else {

        }
}

void read() {
        n = (1 << 10);
        g.resize(n);
        for (int m = 0; m < n; m++) {
                int cnt = 0;
                int mm = m;
                while (mm) {
                        cnt += 1 & mm;
                        mm >>= 1;
                }
                if (cnt == 3) {
                        for (int i = 0; i < 10; i++) {
                                mm = m | (1 << i);
                                if (mm != m) {
                                        g[m].pb(mm);
                                        // g[mm].pb(m);
                                }
                        }
                }
        }
        mt.assign(n, -1);
        for (int i = 0; i < n; i++) {
                used.assign(n, 0);
                khyn(i);
                // cout << i << " " << khyn(i) << "\n";
        }

        to.assign(n, -1);
        for (int i = 0; i < n; i++) {
                if (mt[i] == -1) {
                        // cout << "ERROR\n";
                        // return;
                } else {
                        to[mt[i]] = i;
                        // cout << mt[i] << " - " << to[mt[i]] << "\n";
                }
        }

        string s;
        cin >> s;
        int t;
        cin >> t;
        while (t--) {
                if (s == "add") solve_add();
                if (s == "clear") solve_clear();
        }

}

signed main() {
        ios::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        read();
        return 0;
}
```

Task F ()

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

#define ll long long
#define pb push_back
#define pii pair<int, int>

using namespace std;

int n;

void solve8() {
        cout << "4\n";
        cout << "-1␣-1\n";
        cout << "-1␣1\n";
        cout << "1␣1\n";
        cout << "1␣-1\n";
        vector<pii> a = {
                {2, -2},
                {2, 0},
                {2, 2},
                {0, 2},
                {-2, 2},
                {-2, 0},
                {-2, -2},
                {0, -2}
        };
        for (int i = 0; i < n; i++) {
                cout << a[i].first << "␣" << a[i].second << "\n";
        }
}

void read() {
        cin >> n;
        if (n <= 8) {
                solve8();
        }
}

signed main() {
        ios::sync_with_stdio(0);
        cin.tie(0);
        cout.tie(0);
        read();
        return 0;
}
```