Олимпиада СПбГУ по информатике 2020/21 учебного года

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

Task A ()

```
//#pragma GCC optimize("Ofast,no-stack-protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("section-anchors")
//#pragma GCC optimize("profile-values,profile-reorder-functions,tracer")
//#pragma GCC optimize("vpt")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("move-loop-invariants")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("function-sections")
//#pragma GCC optimize("data-sections")
//#pragma GCC optimize("branch-target-load-optimize")
//#pragma GCC optimize("branch-target-load-optimize2")
//#pragma GCC optimize("btr-bb-exclusive")

#include <iostream>
#include <map>
#include <vector>
#include <set>
#include <stack>
#include <bitset>
#include <queue>
#include <algorithm>
#include <iomanip>
#include <unordered_map>
#include <unordered_set>
#include <cmath>
#include <cassert>
#include <random>

using namespace std;

void solve();

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
#if DEBUG
    freopen("input.txt", "r", stdin);
//    freopen("output.txt", "w", stdout);
#endif
    solve();
}

#define int long long

const int N = 501, A = 1e7 + 100, K = 21, Q = 11, INF = 1e18, MOD = 998244353, BASE = 228, MOD1 =
    1e9 + 7, BASE1 = 131;// 998244353
const double EPS = 0.0000001;

int f(int k) {

}
```

```
void solve() {
    int k;
    cin >> k;
    if (k <= 9) {
        cout << k;
    } else {
        k -= 10;
        if (k % 9 == 0) {
            cout << 0;
        } else {
            cout << k % 9 + 1;
        }
    }
//    vector<int> v, e;
//    for (int i = 1; i <= 100; i++) {
//        v.push_back(i);
//    }
//    e.resize(1000);
//    for (int i = 0; i < v.size() - 1; i++) {
//        e[i] = v[i] % 10;
//        if (i) v[i + 1] += v[i] / 10;
//    }
//    for (int i = 0; i < 100; i++) {
//        cout << e[i] << " ";
//    }
//    cout << endl;
//    for (int i = 0; i < 100; i++) {
//        cout << f(i + 1) << " ";
//    }
}
```

Task B ()

```cpp
//#pragma GCC optimize("Ofast,no-stack-protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("section-anchors")
//#pragma GCC optimize("profile-values,profile-reorder-functions,tracer")
//#pragma GCC optimize("vpt")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("move-loop-invariants")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("function-sections")
//#pragma GCC optimize("data-sections")
//#pragma GCC optimize("branch-target-load-optimize")
//#pragma GCC optimize("branch-target-load-optimize2")
//#pragma GCC optimize("btr-bb-exclusive")

#include <iostream>
#include <map>
#include <vector>
#include <set>
#include <stack>
#include <bitset>
#include <queue>
#include <algorithm>
#include <iomanip>
#include <unordered_map>
#include <unordered_set>
#include <cmath>
#include <cassert>
#include <random>

using namespace std;

void solve();

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
#if DEBUG
    freopen("input.txt", "r", stdin);
//    freopen("output.txt", "w", stdout);
#endif
    solve();
}

#define int long long

const int N = 1e5 + 100, A = 1e7 + 100, K = 21, Q = 11, INF = 1e18, MOD = 998244353, BASE = 228,
    MOD1 = 1e9 + 7, BASE1 = 131;// 998244353
const double EPS = 0.0000001;

int n, k;
string s;

void solve() {
    cin >> n >> k >> s;
    set<char> se;
    se.insert(s[0]);
    int ans = 1, cnt = 1;
    for (int i = 1; i < n; i++) {
        if (cnt == k || (se.size() == 3 && !se.count(s[i]))) {
            ans++;
            cnt = 0;
            se.clear();
        }
        cnt++;
        se.insert(s[i]);
    }
    cout << ans;
}
```

Task C ()

```
//#pragma GCC optimize("Ofast,no-stack-protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("section-anchors")
//#pragma GCC optimize("profile-values,profile-reorder-functions,tracer")
//#pragma GCC optimize("vpt")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("move-loop-invariants")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("function-sections")
//#pragma GCC optimize("data-sections")
//#pragma GCC optimize("branch-target-load-optimize")
//#pragma GCC optimize("branch-target-load-optimize2")
//#pragma GCC optimize("btr-bb-exclusive")

#include <iostream>
#include <map>
#include <vector>
#include <set>
#include <stack>
#include <bitset>
#include <queue>
#include <algorithm>
#include <iomanip>
#include <unordered_map>
#include <unordered_set>
#include <cmath>
#include <cassert>
#include <random>

using namespace std;

void solve();

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
#if DEBUG
    freopen("input.txt", "r", stdin);
//    freopen("output.txt", "w", stdout);
#endif
    solve();
}


const int N = 501, A = 250001, K = 21, Q = 11, INF = 1e18, MOD = 998244353, BASE = 228, MOD1 = 1e9
     + 7, BASE1 = 131;// 998244353
const double EPS = 0.0000001;

int n, x, y, dp[2][A], a[N], b[N];
vector<bool> p[N], ans(N);

void solve() {
    for (int i = 0; i < N; i++) {
        p[i].resize(A);
    }
    cin >> n >> x >> y;
    int sum2 = 0;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    for (int i = 0; i < n; i++) {
        cin >> b[i];
        sum2 += b[i];
    }
    for (int i = 1; i <= n; i++) {
        for (int j = 0; j <= x; j++) {
            dp[1][j] = dp[0][j];
            if (j >= a[i - 1] && dp[1][j] < dp[0][j - a[i - 1]] + b[i - 1]) {
                dp[1][j] = max(dp[1][j], dp[0][j - a[i - 1]] + b[i - 1]);
                p[i][j] = true;
            }
        }
        for (int j = 1; j <= x; j++) {
            if (dp[1][j] < dp[1][j - 1]) {
                dp[1][j] = dp[1][j - 1];
                p[i][j] = p[i][j - 1];
            }
        }
        for (int j = 0; j <= x; j++) {
            dp[0][j] = dp[1][j];
            dp[1][j] = 0;
        }
    }
    int mx = -INF, val = 0;
    for (int i = 0; i <= x; i++) {
        if (mx < dp[0][i]) {
            mx = dp[0][i];
            val = i;
        }
    }
    if (sum2 - mx > y) {
        cout << -1;
        return;
    }
    int pos = n;
    while (pos > 0) {
        if (p[pos][val]) {
            ans[pos - 1] = true;
            val -= a[pos - 1];
        }
        pos--;
    }
    for (int i = 0; i < n; i++) {
        if (ans[i]) {
            cout << "x";
        } else {
            cout << "y";
        }
    }
}
```

Task D ()

```cpp
//#pragma GCC optimize("Ofast,no-stack-protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("section-anchors")
//#pragma GCC optimize("profile-values,profile-reorder-functions,tracer")
//#pragma GCC optimize("vpt")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("move-loop-invariants")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("function-sections")
//#pragma GCC optimize("data-sections")
//#pragma GCC optimize("branch-target-load-optimize")
//#pragma GCC optimize("branch-target-load-optimize2")
//#pragma GCC optimize("btr-bb-exclusive")

#include <iostream>
#include <map>
#include <vector>
#include <set>
#include <stack>
#include <bitset>
#include <queue>
#include <algorithm>
#include <iomanip>
#include <unordered_map>
#include <unordered_set>
#include <cmath>
#include <cassert>
#include <random>

using namespace std;

void solve();

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
#if DEBUG
    freopen("input.txt", "r", stdin);
//    freopen("output.txt", "w", stdout);
#endif
    solve();
}

#define int long long

const int N = 501, A = 1e7 + 100, K = 21, Q = 11, INF = 1e18, MOD = 998244353, BASE = 228, MOD1 = 1e9 + 7, BASE1 = 131;// 998244353
const double EPS = 0.0000001;

int n, dp[N][N];
string s;

void solve() {
    cin >> n >> s;
    n *= 2;
    for (int i = 0; i < n; i++) {
        if (s[i] == ')') s[i] = '(';
        if (s[i] == ']') s[i] = '[';
    }
    int ans = 0;
    vector<char> v;
    for (int i = 0; i < n; i++) {
        if (!v.empty() && v.back() == s[i]) {
            v.pop_back();
        } else {
            v.push_back(s[i]);
        }
    }
    cout << v.size() / 2;
    return;

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            dp[i][j] = INF;
        }
    }
    for (int i = 0; i < n - 1; i++) {
        if (s[i] != s[i + 1]) {
            dp[i][i + 1] = 1;
        } else {
            dp[i][i + 1] = 0;
        }
    }
    for (int len = 3; len < n; len++) {
        if (len % 2 == 0) continue;
        for (int i = 0; i < n - len; i++) {
            int j = i + len;
            if (s[i] == s[j]) {
                dp[i][j] = min(dp[i][j], dp[i + 1][j - 1]);
            } else {
                dp[i][j] = min(dp[i][j], dp[i + 1][j - 1] + 1);
            }
            for (int h = i + 1; h < j; h++) {
                dp[i][j] = min(dp[i][j], dp[i][h] + dp[h + 1][j]);
            }
        }
    }
    cout << dp[0][n - 1];
}
```

Task E ()

```cpp
//#pragma GCC optimize("Ofast,no-stack-protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("section-anchors")
//#pragma GCC optimize("profile-values,profile-reorder-functions,tracer")
//#pragma GCC optimize("vpt")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("move-loop-invariants")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("function-sections")
//#pragma GCC optimize("data-sections")
//#pragma GCC optimize("branch-target-load-optimize")
//#pragma GCC optimize("branch-target-load-optimize2")
//#pragma GCC optimize("btr-bb-exclusive")

#include <iostream>
#include <map>
#include <vector>
#include <set>
#include <stack>
#include <bitset>
#include <queue>
#include <algorithm>
#include <iomanip>
#include <unordered_map>
#include <unordered_set>
#include <cmath>
#include <cassert>
#include <random>

using namespace std;

void solve();

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
#if DEBUG
    freopen("input.txt", "r", stdin);
//    freopen("output.txt", "w", stdout);
#endif
    solve();
}

#define int long long

const int N = 501, A = 1e5 + 100, K = 21, Q = 11, F = 423907, INF = 1e18, MOD = 998244353, BASE =
    228, MOD1 = 1e9 + 7, BASE1 = 131;// 998244353
const double EPS = 0.0000001;

bool bit(int a, int n) {
    return (a >> n) & 1;
}

int t, n, k;
vector<int> a;
bool b[A];

void solve() {
    srand(228);
    vector<int> d;
    for (int i = 0; i < 100000 - 9; i++) {
        d.push_back(i);
    }
    random_shuffle(d.begin(), d.end());
    vector<int> test;
    for (int i = 0; i < 10; i++) {
        test.push_back(i);
    }
    vector<int> mm;
    for (int i = 0; i < (1 << 10); i++) {
        mm.push_back(i);
    }
    swap(mm[0], mm[100]);
//    reverse(mm.begin(), mm.end());
//    random_shuffle(mm.begin(), mm.end());
    vector<int> u;
    for (int i = 0; i < 10; i++) {
        u.push_back(i);
    }
//    random_shuffle(u.begin(), u.end());
    map<int, int> ma;
    set<int> se;
    for (int h = 0; h < mm.size(); h++) {
        int mask = mm[h];
        int cnt = 0;
        for (int i = 0; i < 10; i++) {
            cnt += bit(mask, i);
        }
        if (cnt != 3) continue;
        int smask;
        for (int j : u) {
            if (!bit(mask, j)) {
                smask = (mask | (1 << j));
                if (!se.count(smask)) {
                    se.insert(smask);
                    ma[mask] = smask;
                    break;
                }
            }
        }
    }
    string type;
    cin >> type >> t;
    while (t--) {
        if (type == "add") {
            cin >> n >> k;
            if (n == 10 && k == 3) {
                a.resize(k);
                for (int i = 0; i < k; i++) {
                    cin >> a[i];
                    a[i]--;
                }
                int mask = 0;
                for (int i = 0; i < k; i++) {
                    mask |= (1 << a[i]);
                }
                int mmask = (mask ^ ma[mask]);
                for (int i = 0; i < n; i++) {
                    if (bit(mmask, i)) {
                        cout << i + 1 << "␣";
                    }
                }
                cout << '\n';
            } else if (n == 1000000) {
                a.resize(k);
                for (int i = 0; i < k; i++) {
                    cin >> a[i];
                }
                cout << F << '\n';
            } else {
                a.resize(k);
                for (int i = 0; i < k; i++) {
                    cin >> a[i];
                    a[i]--;
                }
                fill(b, b + A, false);
                for (int i = 0; i < k; i++) {
                    b[a[i]] = true;
                }
                for (int i : d) {
                    int mask = 0, cnt = 0;
                    for (int j = 0; j < 10; j++) {
```

```
                    if (b[i + j]) {
                        mask |= (1 << j);
                        cnt++;
                    }
                }
                if (cnt == 3) {
                    int mmask = (ma[mask] ^ mask);
                    for (int j : u) {
                        if (bit(mmask, j)) {
                            cout << i + j + 1 << '\n';
                            break;
                        }
                    }
                    break;
                }
            }
        }
    } else {
        cin >> n >> k;
        if (n == 10 && k == 3) {
            k++;
            a.resize(k);
            for (int i = 0; i < k; i++) {
                cin >> a[i];
                a[i]--;
            }
            int mask = 0;
            for (int i = 0; i < k; i++) {
                mask |= (1 << a[i]);
            }
            for (int mmask : mm) {
                if (ma.count(mmask) && ma[mmask] == mask) {
                    for (int i = 0; i < n; i++) {
                        if (bit(mmask, i)) {
                            cout << i + 1 << " ";
                        }
                    }
                    break;
                }
            }
            cout << '\n';
        } else if (n == 1000000) {
            k++;
            a.resize(k);
            for (int i = 0; i < k; i++) {
                cin >> a[i];
            }
            for (int i = 0; i < k; i++) {
                if (a[i] != F) {
                    cout << a[i] << " ";
                }
            }
            cout << '\n';
        } else {
            k++;
            a.resize(k);
            for (int i = 0; i < k; i++) {
                cin >> a[i];
                a[i]--;
            }
            fill(b, b + A, false);
            for (int i = 0; i < k; i++) {
                b[a[i]] = true;
            }
            for (int i : d) {
                int mask = 0, cnt = 0;
                for (int j = 0; j < 10; j++) {
                    if (b[i + j]) {
                        mask |= (1 << j);
                        cnt++;
                    }
                }
                if (cnt == 4) {
                    bool ok = false;
```

```
                        for (int j : u) {
                            if (bit(mask, j)) {
                                int mmask = (mask ^ (1 << j));
                                if (ma.count(mmask) && ma[mmask] == mask) {
                                    b[i + j] = false;
                                    ok = true;
                                    break;
                                }
                            }
                        }
                        if (ok) {
                            break;
                        }
                    }
                }
                for (int i = 0; i < A; i++) {
                    if (b[i]) {
                        cout << i + 1 << " ";
                    }
                }
            }
        }
    }
}
```

Task F ()

```
//#pragma GCC optimize("Ofast,no-stack-protector")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("section-anchors")
//#pragma GCC optimize("profile-values,profile-reorder-functions,tracer")
//#pragma GCC optimize("vpt")
//#pragma GCC optimize("rename-registers")
//#pragma GCC optimize("move-loop-invariants")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("function-sections")
//#pragma GCC optimize("data-sections")
//#pragma GCC optimize("branch-target-load-optimize")
//#pragma GCC optimize("branch-target-load-optimize2")
//#pragma GCC optimize("btr-bb-exclusive")

#include <iostream>
#include <map>
#include <vector>
#include <set>
#include <stack>
#include <bitset>
#include <queue>
#include <algorithm>
#include <iomanip>
#include <unordered_map>
#include <unordered_set>
#include <cmath>
#include <cassert>
#include <random>

using namespace std;

void solve();

int main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
#if DEBUG
    freopen("input.txt", "r", stdin);
//    freopen("output.txt", "w", stdout);
#endif
    solve();
}

#define int long long

const int N = 100000, A = 1e5 + 100, K = 30000, Q = 11, F = 423907, INF = 1e18, MOD = 998244353,
    BASE = 228, MOD1 = 1e9 + 7, BASE1 = 131;// 998244353
const double EPS = 0.0000001;

int n;

void f(int a, int b) {
    cout << a << "␣" << b << '\n';
}

void solve() {
    cin >> n;
    cout << 4 << '\n';
    f(0, 0);
    f(1, 0);
    f(1, 1);
    f(0, 1);
    vector<int> x = {1, 1, 0, -1, -1, -1, 0, 1};
    vector<int> y = {0, 1, 1, 1, 0, -1, -1, -1};
    for (int i = 0; i < n; i++) {
        f(x[i], y[i]);
    }
}
```