

Олимпиада СПбГУ по информатике 2020/21 учебного года

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

Task A ()

```
#ifndef LOCAL
#pragma GCC optimize("O3")
// #pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target ("avx,avx2,fma")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma,tune=
    native")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma")
// #pragma GCC target("tune=native")
#endif

#include <bits/stdc++.h>

#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
using namespace std;
#define endl "\n"

#define double long double
// #define int long long

#ifdef int
#define IMAX INT64_MAX
#define IMIN INT64_MIN
#else
#define IMAX INT32_MAX
#define IMIN INT32_MIN
#endif

#ifdef LOCAL
#define IN_FILE string("in.txt")
#else
// #define FILES string("f")
#ifdef FILES
#define IN_FILE FILES + ".in"
#define OUT_FILE FILES + ".out"
#endif
#endif

#ifdef IN_FILE
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

#ifdef OUT_FILE
#define cout fout
ofstream fout;
#endif

// #define EPS 1e-7L
// #define MULTITEST

inline void solve() {
    int k;
    cin >> k;
    if (k <= 10) {
        cout << "1234567890"[k - 1] << endl;
    }
}
```

```

        return;
    }

    cout << "234567890"[(k - 11) % 9] << endl;
}

int32_t main() {
#ifdef IN_FILE
    fin.open(IN_FILE);
#endif
#ifdef OUT_FILE
    fout.open(OUT_FILE);
#endif
    ON_START

#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; ++i) {
#ifdef MULTITEST
        solve();
    }
#endif
    return 0;
}

```

Task B ()

```
#ifndef LOCAL
#pragma GCC optimize("O3")
// #pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target ("avx,avx2,fma")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma,tune=
    native")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma")
// #pragma GCC target ("tune=native")
#endif

#include <bits/stdc++.h>

#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
using namespace std;
#define endl "\n"

#define double long double
// #define int long long

#ifdef int
#define IMAX INT64_MAX
#define IMIN INT64_MIN
#else
#define IMAX INT32_MAX
#define IMIN INT32_MIN
#endif

#ifdef LOCAL
#define IN_FILE string("in.txt")
#else
// #define FILES string("f")
#ifdef FILES
#define IN_FILE FILES + ".in"
#define OUT_FILE FILES + ".out"
#endif
#endif

#ifdef IN_FILE
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

#ifdef OUT_FILE
#define cout fout
ofstream fout;
#endif

// #define EPS 1e-7L
// #define MULTITEST

inline void solve() {
    int n, k;
    cin >> n >> k;
    string s;
    cin >> s;
    int psum[28][n + 1];
    for (int i = 0; i < 28; i++) {
        psum[i][0] = 0;
    }
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < 28; j++) {
            psum[j][i + 1] = psum[j][i];
        }
        psum[s[i] - 'a'][i + 1]++;
    }

    int dp[n + 1];
    dp[0] = 0;

    for (int p = 1; p <= n; p++) {
        int l = 1, r = min(k + 1, p + 1);
```

```

        while (r != l + 1) {
            int m = (l + r) / 2;

            int unq = 0;

            for (int c = 0; c < 28; c++) {
                if (psum[c][p] - psum[c][p - m] > 0) {
                    unq++;
                    if (unq > 3) {
                        break;
                    }
                }
            }

            if (unq > 3) {
                r = m;
            } else {
                l = m;
            }
        }
        dp[p] = dp[p - 1] + 1;
    }

    cout << dp[n] << endl;
}

int32_t main() {
#ifdef IN_FILE
    fin.open(IN_FILE);
#endif
#ifdef OUT_FILE
    fout.open(OUT_FILE);
#endif
    ON_START

#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; ++i) {
#endif
        solve();
#ifdef MULTITEST
    }
#endif

    return 0;
}

```

Task C ()

```
#ifndef LOCAL
#pragma GCC optimize("O3")
// #pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target ("avx,avx2,fma")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma,tune=
    native")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma")
// #pragma GCC target ("tune=native")
#endif

#include <bits/stdc++.h>

#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
using namespace std;
#define endl "\n"

#define double long double
// #define int long long

#ifdef int
#define IMAX INT64_MAX
#define IMIN INT64_MIN
#else
#define IMAX INT32_MAX
#define IMIN INT32_MIN
#endif

#ifdef LOCAL
#define IN_FILE string("in.txt")
#else
// #define FILES string("f")
#ifdef FILES
#define IN_FILE FILES + ".in"
#define OUT_FILE FILES + ".out"
#endif
#endif

#ifdef IN_FILE
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

#ifdef OUT_FILE
#define cout fout
ofstream fout;
#endif

// #define EPS 1e-7L
// #define MULTITEST

// struct itm_t {
//     int i, v, u;
// };

struct ll_t {
    int v;
    ll_t *prev;
    int rfc;
};

vector<ll_t *> fr;
ll_t lls[30000000];
int lls_i = 0;

void dec_ll(ll_t *v) {
    if (v == nullptr) {
        return;
    }
    v->rfc--;
    if (v->rfc == 0) {
        dec_ll(v->prev);
    }
}
```

```

        fr.push_back(v);
    }
}

inline void solve() {
    auto strt = chrono::steady_clock::now();

    int n, x, y;
    cin >> n >> x >> y;

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

    pair<int, ll_t*> can[x + 1];
    for (int i = 0; i <= x; i++) {
        can[i] = {-1, nullptr};
    }
    can[0] = {0, nullptr};

    int is[n];
    for (int i = 0; i < n; i++) {
        is[i] = i;
    }

    sort(is, is + n, [&](int i, int j) {
        return w[i] > w[j];
    });

    for (int i : is) {
        for (int c = x - v[i]; c >= 0; c--) {
            if (can[c].first == -1) {
                continue;
            }

            if (can[c + v[i]].first < can[c].first + w[i]) {
                can[c + v[i]].first = can[c].first + w[i];
                auto old = can[c + v[i]].second;
                if (fr.empty()) {
                    can[c + v[i]].second = lls + lls_i;
                    lls_i++;
                } else {
                    can[c + v[i]].second = fr.back();
                    fr.pop_back();
                }
                can[c + v[i]].second->rfc = 1;
                can[c + v[i]].second->prev = can[c].second;
                if (can[c].second) {
                    can[c].second->rfc++;
                }
                can[c + v[i]].second->v = i;

                dec_ll(old);

                if (w_sum - can[c].first <= y) {
                    goto oki;
                }
            }
        }
    }

    if (chrono::duration_cast<chrono::milliseconds>(chrono::steady_clock::now() - strt).count()
        >= 1900) {
        break;
    }
}

oki:
int mx_can_i = 0;

```

```

    for (int i = 0; i <= x; i++) {
        if (can[i].first > can[mx_can_i].first) {
            mx_can_i = i;
        }
    }

    if (w_sum - can[mx_can_i].first > y) {
        cout << "-1\n";
        return;
    }

    string res(n, 'y');
    ll_t *ll = can[mx_can_i].second;
    while (ll != nullptr) {
        res[ll->v] = 'x';
        ll = ll->prev;
    }
    cout << res << endl;
}

int32_t main() {
#ifdef IN_FILE
    fin.open(IN_FILE);
#endif
#ifdef OUT_FILE
    fout.open(OUT_FILE);
#endif
    ON_START

#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; ++i) {
#endif
        solve();
#ifdef MULTITEST
    }
#endif

    return 0;
}

```

Task D ()

```
#ifndef LOCAL
#pragma GCC optimize("O3")
// #pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target ("avx,avx2,fma")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma,tune=
    native")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma")
// #pragma GCC target ("tune=native")
#endif

#include <bits/stdc++.h>

#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
using namespace std;
#define endl "\n"

#define double long double
#define int long long

#ifdef int
#define IMAX INT64_MAX
#define IMIN INT64_MIN
#else
#define IMAX INT32_MAX
#define IMIN INT32_MIN
#endif

#ifdef LOCAL
#define IN_FILE string("in.txt")
#else
// #define FILES string("f")
#ifdef FILES
#define IN_FILE FILES + ".in"
#define OUT_FILE FILES + ".out"
#endif
#endif

#ifdef IN_FILE
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

#ifdef OUT_FILE
#define cout fout
ofstream fout;
#endif

// #define EPS 1e-7L
// #define MULTITEST

inline void solve() {
    int n;
    string s;
    cin >> n >> s;
    vector<char> st;
    for (char ch : s) {
        if (ch == '(' || ch == ')') {
            if (!st.empty() && st.back() == '(') {
                st.pop_back();
            } else {
                st.push_back('(');
            }
        } else {
            if (!st.empty() && st.back() == '[') {
                st.pop_back();
            } else {
                st.push_back('[');
            }
        }
    }
}
```



```

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

    int32_t main() {
#ifdef IN_FILE
        fin.open(IN_FILE);
#endif
#ifdef OUT_FILE
        fout.open(OUT_FILE);
#endif
        ON_START

#ifdef MULTITEST
        int t;
        cin >> t;
        for (int i = 0; i < t; ++i) {
#endif
            solve();
#ifdef MULTITEST
        }
#endif

        return 0;
    }

```

Task E ()

```
#ifndef LOCAL
#pragma GCC optimize("O3")
// #pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target ("avx,avx2,fma")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma,tune=
    native")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma")
// #pragma GCC target ("tune=native")
#endif

#include <bits/stdc++.h>

#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
using namespace std;
#define endl "\n"

#define double long double
#define int long long

#ifdef int
#define IMAX INT64_MAX
#define IMIN INT64_MIN
#else
#define IMAX INT32_MAX
#define IMIN INT32_MIN
#endif

#ifdef LOCAL
#define IN_FILE string("in.txt")
#else
// #define FILES string("f")
#ifdef FILES
#define IN_FILE FILES + ".in"
#define OUT_FILE FILES + ".out"
#endif
#endif

#ifdef IN_FILE
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

#ifdef OUT_FILE
#define cout fout
ofstream fout;
#endif

// #define EPS 1e-7L
// #define MULTITEST

inline void solve() {
    string tt;
    int t;
    cin >> tt >> t;

    map<vector<int>, int> rrr;
    map<vector<int>, int> obr;
    for (int i = 1; i <= 10; i++) {
        for (int j = i + 1; j <= 10; j++) {
            for (int k = j + 1; k <= 10; k++) {
                bool ok = false;
                for (int l = 1; l <= 10; l++) {
                    if (l == i || l == j || l == k) {
                        continue;
                    }
                }
                vector<int> rr;
                rr.push_back(l);
                rr.push_back(i);
                rr.push_back(j);
                rr.push_back(k);
                sort(rr.begin(), rr.end());
            }
        }
    }
}
```

```

        if (rrr.find(rr) == rrr.end()) {
            rrr[rr] = 1;

            rr.clear();
            rr.push_back(i);
            rr.push_back(j);
            rr.push_back(k);
            obr[rr] = 1;

            ok = true;
            break;
        }
    }
    assert(ok);
}

}

if (tt == "add") {
    for (int asd = 0; asd < t; asd++) {
        int n, k;
        cin >> n >> k;

        if (n == 1000000) {
            int mult = 1;
            int v;
            for (int i = 0; i < k; i++) {
                cin >> v;
                mult *= v;
                mult %= 1000000007;
            }
            mult *= 383337;
            cout << mult % n + 1 << endl;
        } else if (n == 100000) {
            const int mgv = 78231;

            set<int> mmm;

            for (int i = 0; i < k; i++) {
                int v;
                cin >> v;
                mmm.insert(v);
            }

            for (int i = mgv;; i *= 313) {
                while (i > n) {
                    i = i % n + 1;
                }

                if (mmm.find(i) == mmm.end()) {
                    cout << i << endl;
                    break;
                }
            }

        } else {
            vector<int> a(k);
            for (int i = 0; i < k; i++) {
                cin >> a[i];
            }
            sort(a.begin(), a.end());
            cout << obr[a] << endl;
        }
    }
} else {
    for (int asd = 0; asd < t; asd++) {
        int n, k;
        cin >> n >> k;
        if (n == 1000000) {
            k++;
            int a[k];
            for (int i = 0; i < k; i++) {
                cin >> a[i];
            }
        }
    }
}

```

```

    for (auto res : a) {
        int mult = 1;
        for (int v : a) {
            if (v == res) {
                continue;
            }
            mult *= v;
            mult %= 10000000007;
        }

        if ((mult * 383337) % n + 1 == res) {
            for (int v : a) {
                if (v == res) {
                    continue;
                }
                cout << v << " ";
            }
            cout << endl;
        }
    }
} else if (n == 100000) {
    k++;
    const int mgv = 78231;

    set<int> mmm;

    for (int i = 0; i < k; i++) {
        int v;
        cin >> v;
        mmm.insert(v);
    }

    int res;
    for (int i = mgv;; i *= 313) {
        while (i > n) {
            i = i % n + 1;
        }

        if (mmm.find(i) == mmm.end()) {
            break;
        } else {
            res = i;
        }
    }

    cout << mmm.size() << endl;
    for (int v : mmm) {
        if (v == res) {
            continue;
        }
        cout << v << " ";
    }
    cout << endl;

} else {
    k++;
    vector<int> a(k);
    for (int i = 0; i < k; i++) {
        cin >> a[i];
    }
    sort(a.begin(), a.end());
    int res = rrr[a];
    for (int v : a) {
        if (v == res) {
            continue;
        }
        cout << v << " ";
    }
    cout << endl;
}

}

}

}

int32_t main() {

```

```

#ifdef IN_FILE
    fin.open(IN_FILE);
#endif
#ifdef OUT_FILE
    fout.open(OUT_FILE);
#endif
ON_START

#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; ++i) {
#endif
        solve();
#ifdef MULTITEST
    }
#endif

    return 0;
}

```

Task F ()

```
#ifndef LOCAL
#pragma GCC optimize("O3")
// #pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target ("avx,avx2,fma")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma,tune=
    native")
// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,sse4.1,sse4.2,abm,mmx,avx,avx2,popcnt,fma")
// #pragma GCC target ("tune=native")
#endif

#include <bits/stdc++.h>

#define ON_START ios_base::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
using namespace std;
#define endl "\n"

#define double long double
#define int long long

#ifdef int
#define IMAX INT64_MAX
#define IMIN INT64_MIN
#else
#define IMAX INT32_MAX
#define IMIN INT32_MIN
#endif

#ifdef LOCAL
#define IN_FILE string("in.txt")
#else
// #define FILES string("f")
#ifdef FILES
#define IN_FILE FILES + ".in"
#define OUT_FILE FILES + ".out"
#endif
#endif

#ifdef IN_FILE
#define cin fin
ifstream fin; // NOLINT(cert-err58-cpp)
#endif

#ifdef OUT_FILE
#define cout fout
ofstream fout;
#endif

// #define EPS 1e-7L
// #define MULTITEST

inline void solve() {
    int n;
    cin >> n;
    cout << "4\n0_0\n1_0\n1_1\n0_1\n";
    pair<int, int> rrr[] = {
        {0, 1}, {1, 1}, {1, 0}, {1, -1}, {0, -1}, {-1, -1}, {-1, 0}, {-1, 1}
    };
    assert(n <= 8);
    for (int i = 0; i < n; i++) {
        cout << rrr[i].first << "_" << rrr[i].second << endl;
    }
}

int32_t main() {
#ifdef IN_FILE
    fin.open(IN_FILE);
#endif
#ifdef OUT_FILE
    fout.open(OUT_FILE);
#endif
    ON_START
```

```

#ifdef MULTITEST
    int t;
    cin >> t;
    for (int i = 0; i < t; ++i) {
#ifdef MULTITEST
        solve();
    }
#endif
    return 0;
}

```