Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	80	58	0	438

Task A ()

```
#pragma GCC optimize("O3")
#pragma GCC optimize("unroll-loops")
//#pragma GCC target("avx2,avx,fma,bmi2")

#include <bits/stdc++.h>
#include <immintrin.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;

#define endl '\n'
#define int long long
#define all(arr) arr.begin(), arr.end()
#define multitest() int _gorilla_silverback; cin >> _gorilla_silverback; while (
    _gorilla_silverback --> 0)
#define ls(id) (id << 1 | 1)
#define rs(id) ((id << 1) + 2)
#define dlg(x) (31 - __builtin_clz(x))
#define ulg(x) (32 - __builtin_clz(x))
#define sqr(x) ((x) * (x))
#define out(x) cout << (x) << endl, exit(0)

typedef pair<int, int> ipair;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update> treap;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());

const int MAXN = 200200;

int n;
int arr[MAXN], brr[MAXN];
map<int, vector<int>> D;

void build() {
        for (int i = 0; i < n; ++i) {
                for (; !(arr[i] % 10); arr[i] /= 10)
                        ++brr[i];
                D[brr[i]].push_back(arr[i]);
        }
}

int solve() {
        for (auto& [key, val]: D) {
                int sum = accumulate(all(val), 0LL);
                int cnt = 0;
                for (; !(sum % 10); sum /= 10)
                        ++cnt;
                if (!cnt) return key;
                D[key + cnt].push_back(sum);
        }
        return -1;
}

int32_t main() {
        ios_base::sync_with_stdio(false), cin.tie(NULL), cout.tie(NULL);
        #ifdef LOCAL
        freopen("input.txt", "r", stdin), freopen("output.txt", "w", stdout);
        #endif
        cin >> n;
        for (int i = 0; i < n; ++i)
                cin >> arr[i] >> brr[i];
        build();
        cout << solve() << endl;
}
```

Task B ()

```
#pragma GCC optimize("O3")
#pragma GCC optimize("unroll-loops")
//#pragma GCC target("avx2,avx,fma,bmi2")

#include <bits/stdc++.h>
#include <immintrin.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;

//#define int long long
#define all(arr) arr.begin(), arr.end()
#define multitest() int _gorilla_silverback; cin >> _gorilla_silverback; while (
    _gorilla_silverback --> 0)
#define ls(id) (id << 1 | 1)
#define rs(id) ((id << 1) + 2)
#define dlg(x) (31 - __builtin_clz(x))
#define ulg(x) (32 - __builtin_clz(x))
#define sqr(x) ((x) * (x))
#define out(x) cout << (x) << endl, exit(0)

typedef pair<int, int> ipair;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update> treap;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());

const int MAXN = 110;

int n;
int arr[MAXN];

inline int calc() {
        string s; cin >> s;
        if (s == "Burn" || s == "Tired" || s == "Fail")
                exit(0);
        return count(all(s), 'e') >> 1;
}

void go(int i, int x) {
        while (x--) {
                cout << "Flip␣and␣wait" << endl;
                for (int cur = calc(); cur <= i; cur += calc())
                        cout << "Wait" << endl;
        }
}

void solve() {
        for (int i = n - 1; i >= 0; --i)
                go(i, arr[i]);
        cout << "Stop" << endl;
}

int32_t main() {
        cin >> n;
        for (int i = 0; i < n; ++i)
                cin >> arr[i];
        solve();
}
```

Task C ()

```
#pragma GCC optimize("O3")
#pragma GCC optimize("unroll-loops")
//#pragma GCC target("avx2,avx,fma,bmi2")

#include <bits/stdc++.h>
#include <immintrin.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;

#define endl '\n'
//#define int long long
#define all(arr) arr.begin(), arr.end()
#define multitest() int _gorilla_silverback; cin >> _gorilla_silverback; while (
    _gorilla_silverback --> 0)
#define ls(id) (id << 1 | 1)
#define rs(id) ((id << 1) + 2)
#define dlg(x) (31 - __builtin_clz(x))
#define ulg(x) (32 - __builtin_clz(x))
#define sqr(x) ((x) * (x))
#define out(x) cout << (x) << endl, exit(0)

typedef pair<int, int> ipair;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update> treap;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());

int n;
string s;

void solve() {
        int c0 = count(all(s), '0');
        int c1 = n - c0;
        if (!c1) fill(s.begin(), s.begin() + (n >> 1), '?');
        else if (c0 >= c1) replace(all(s), '0', '?');
        else replace(all(s), '1', '?');
}

void solve2() {
        char c = *min_element(all(s));
        int ca = count(all(s), '?');
        if (ca == (n >> 1) && c == '0') fill(all(s), '0');
        else if (ca == n) fill(all(s), '1');
        else replace(all(s), '?', (char)(c ^ 1));
}

int32_t main() {
        ios_base::sync_with_stdio(false), cin.tie(NULL), cout.tie(NULL);
        #ifdef LOCAL
        freopen("input.txt", "r", stdin), freopen("output.txt", "w", stdout);
        #endif
        multitest() {
                cin >> s, n = s.size();
        //      assert(!(n & 1));
                if (find(all(s), '?') == s.end()) solve();
                else solve2();
                cout << s << endl;
        }
}
```

Task D ()

```
#pragma GCC optimize("O3")
#pragma GCC optimize("unroll-loops")
//#pragma GCC target("avx2,avx,fma,bmi2")

#include <bits/stdc++.h>
#include <immintrin.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;

#define endl '\n'
#define int long long
#define all(arr) arr.begin(), arr.end()
#define multitest() int _gorilla_silverback; cin >> _gorilla_silverback; while (
    _gorilla_silverback --> 0)
#define ls(id) (id << 1 | 1)
#define rs(id) ((id << 1) + 2)
#define dlg(x) (31 - __builtin_clz(x))
#define ulg(x) (32 - __builtin_clz(x))
#define sqr(x) ((x) * (x))
#define out(x) cout << (x) << endl, exit(0)

typedef pair<int, int> ipair;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update> treap;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());

const int MAXN = 1001000;
const int C = 998244353;
const int C8 = 100000000;

inline int mul(int a, int b) {
        return a * b % C;
}

inline int add(int a, int b) {
        return (a + b >= C ? a + b - C : a + b);
}

int n;
int fact[] = {1, 808258749, 117153405, 761699708, 573994984, 62402409, 511621808, 242726978,
    887890124, 875880304};

int solve() {
        int start = n / C8;
        int sugoma = fact[start];
        for (int i = start * C8 + 1; i <= n; ++i)
                sugoma = mul(sugoma, i);
        return sugoma;
/*      int sum = 1, prv = 1;
        for (int i = 2; i <= n + 1; ++i) {
                int cur = mul(sum, mul(i, i));
                sum = add(sum, prv);
                prv = cur;
        }
        return sum;*/
}

int32_t main() {
        ios_base::sync_with_stdio(false), cin.tie(NULL), cout.tie(NULL);
        #ifdef LOCAL
        freopen("input.txt", "r", stdin), freopen("output.txt", "w", stdout);
        #endif
        cin >> n, ++n;
        cout << 0 << '␣' << solve() << endl;
}
```

Task E ()

```
#pragma GCC optimize("O3")
#pragma GCC optimize("unroll-loops")
//#pragma GCC target("avx2,avx,fma,bmi2")

#include <bits/stdc++.h>
#include <immintrin.h>
#include <ext/pb_ds/assoc_container.hpp>
using namespace std;
using namespace __gnu_pbds;

#define endl '\n'
//#define int long long
#define all(arr) arr.begin(), arr.end()
#define multitest() int _gorilla_silverback; cin >> _gorilla_silverback; while (
    _gorilla_silverback --> 0)
#define ls(id) (id << 1 | 1)
#define rs(id) ((id << 1) + 2)
#define dlg(x) (31 - __builtin_clz(x))
#define ulg(x) (32 - __builtin_clz(x))
#define sqr(x) ((x) * (x))
#define out(x) cout << (x) << endl, exit(0)

typedef pair<int, int> ipair;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update> treap;
mt19937 rng(chrono::steady_clock::now().time_since_epoch().count());

const int MAXN = 300300;

int n, m;
int arr[MAXN];

long long go(int l, int r, int d) {
        long long ans = 0;
        for (int i = l; i < r; ++i)
                ans += (d + arr[i] - 1) / arr[i];
        return ans;
}

int32_t main() {
        ios_base::sync_with_stdio(false), cin.tie(NULL), cout.tie(NULL);
        #ifdef LOCAL
        freopen("input.txt", "r", stdin), freopen("output.txt", "w", stdout);
        #endif
        cin >> n;
        for (int i = 0; i < n; ++i)
                cin >> arr[i];
        cin >> m;
        for (int i = 0; i < m; ++i) {
                int l, r, d; cin >> l >> r >> d;
                cout << go(l, r, d) << endl;
        }
}
```

Task F ()