

Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	58	0	458

Task A ()

```
#ifdef AVEVAD
#define _GLIBCXX_DEBUG
#else
#pragma GCC optimize("O3")
#endif

#include <bits/stdc++.h>

using namespace std;
using ll = long long;

int main() {
#ifdef AVEVAD
    cin.tie(0);
    ios_base::sync_with_stdio(0);
#endif
    ll n;
    cin >> n;
    map<ll, vector<ll>> d;
    for (ll i = 0; i < n; i++) {
        ll a, b;
        cin >> a >> b;
        while (a != 0) {
            d[b].push_back(a % 10);
            a /= 10;
            b++;
        }
    }
    map<ll, ll> ans;
    for (const auto&[b, v]: d) {
        for (ll e: v) ans[b] += e;
        ll pos = b;
        while (true) {
            if (ans[pos] < 10) break;
            ans[pos + 1] += ans[pos] / 10;
            ans[pos] = ans[pos] % 10;
            pos++;
        }
    }
    for (auto[pos, x]: ans) {
        if (x != 0) {
            cout << pos << '\n';
            return 0;
        }
    }
}
```

Task B ()

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

using namespace std;
using ll = long long;

ll query() {
    string s;
    cin >> s;
    if (s.back() != '!') exit(0);
    return (ll(s.length()) - 3ll) / 2ll;
}

void cont() { cout << "Wait" << endl; }

void flip() { cout << "Flip_and_wait" << endl; }

int main() {
    ll n;
    cin >> n;
    vector<ll> a(n);
    for (ll &e: a) cin >> e;
    for (ll i = n - 1; i >= 0; i--) {
        ll c = a[i];
        while (c--) {
            flip(); //
            ll cnt = 0;
            while (true) {
                cnt += query();
                if (cnt > i) break;
                cont();
            }
        }
    }
    cout << "Stop" << endl;
}
```

Task C ()

```
#ifdef AVEVAD
#define _GLIBCXX_DEBUG
#else
#pragma GCC optimize("O3")
#endif

#include <bits/stdc++.h>

using namespace std;
using ll = long long;
ll N = 10;

unordered_map<string, string> D, D1;

bool initd = 0;

void precalc(ll n) {
    for (ll s = 0; s < 1 << n; s++) {
        for (ll t = 0; t < 1 << n; t++) {
            if (__builtin_popcount(t) * 2 + 1 < n) continue;
            string ss(n, '0'), st(n, '0');
            for (ll i = 0; i < n; i++) {
                if (s >> i & 1) ss[i] = '1';
                else ss[i] = '0';
                if (t >> i & 1) st[i] = '?';
                else st[i] = ss[i];
            }
            if (D.find(st) == D.end()) {
                D[st] = ss;
                D1[ss] = st;
                break;
            }
        }
    }
}

void init() {
    if (initd) return;
    initd = 1;
    for (ll n = 1; n <= N; n++) precalc(n);
}

string encr(string s) {
    if (s == string(s.length(), s[0])) {
        s[0] = '?';
        for (ll i = 2; i < s.length(); i++) s[i] = '?';
        return s;
    } else {
        ll n = s.length();
        ll c0 = 0, c1 = 0;
        for (ll i = 1; i < n; i++) (s[i] == '0' ? c0 : c1)++;
        char rep = (c0 > c1 ? '0' : '1');
        for (ll i = 1; i < n; i++) if (s[i] == rep) s[i] = '?';
        return s;
    }
}

string decr(string s) {
    ll n = s.length();
    if (s[0] == '?') {
        return string(n, s[1]);
    } else {
        char rep = 'z';
        for (ll i = 1; i < n; i++) if (s[i] != '?') rep = s[i];
        if (rep == '0') rep = '1';
        else rep = '0';
        for (ll i = 1; i < n; i++) if (s[i] == '?') s[i] = rep;
        return s;
    }
}
```

```

int main() {
#ifdef AVEVAD
    cin.tie(0);
    ios_base::sync_with_stdio(0);
#endif
    ll t;
    cin >> t;
    while (t--) {
        string s;
        cin >> s;
        if (s.find('?') == string::npos) {
            if (s.length() > N) cout << encr(s) << '\n';
            else {
                init();
                cout << D1[s] << '\n';
            }
        } else {
            if (s.length() > N) cout << decr(s) << '\n';
            else {
                init();
                cout << D[s] << '\n';
            }
        }
    }
}

```

Task D ()

```
#ifdef AVEVAD
#define _GLIBCXX_DEBUG
#else
#pragma GCC optimize("O3")
#endif

#include <bits/stdc++.h>

using namespace std;
using ll = long long;

ll M = 998244353, B = 9982443;

ll F[] {1, 119655238, 586503208, 862123873, 268056231, 472864204, 463851853, 741885365, 950157291,
36754542, 788178534,
276678524, 149046885, 755964685, 968778509, 515593892, 779936056, 648609843, 426701733,
745631356, 834758213,
990586025, 311077187, 768124395, 312636731, 551181485, 731990699, 121131545, 712039368,
221806280, 306967785,
258525094, 634443420, 40227110, 515367333, 452567650, 467098669, 518462747, 168100623,
736541403, 428618105,
55793664, 162376288, 353994203, 383570056, 962754962, 738507868, 921814077, 418068877,
407943090, 304702937,
122057714, 572364866, 469001745, 474126097, 18321531, 749515079, 374498425, 404770473,
562709707, 643782198,
560558760, 33216062, 590033128, 310588601, 206798786, 823696305, 171869108, 232670682,
655160238, 529492655,
188078349, 964617697, 23929778, 48242566, 597850370, 532571395, 928997729, 37406901,
292928205, 211434912,
18044336, 579676190, 579818427, 733892312, 58502034, 507709574, 371357386, 638275719,
892456728, 295830656,
790739259, 98715157, 453937126, 908279415, 774426289, 24109512, 948242087, 555039356,
258780878, 734632028};

ll FN = 998244352;

ll fact_part(ll n) {
    ll k = n / B;
    ll ans = F[k];
    for (ll i = k * B + 1; i <= n; i++) ans = (ans * i) % M;
    return ans;
}

ll zero_cnt(ll n) {
    ll ans = 0;
    ans += n / M;
    ans += n / (M * M);
    return ans;
}

ll binpow(ll a, ll n) {
    if (n == 0) return 1;
    ll b = binpow(a, n / 2);
    b = (b * b) % M;
    if (n % 2 == 1) b = (b * a) % M;
    return b;
}

ll fact1(ll n) {
    ll ans = binpow(FN, n / M);
    ans = (ans * fact_part(n % M)) % M;
    return ans;
}

pair<ll, ll> fact_m(ll n) {
    return {zero_cnt(n), fact1(n)};
}

int main() {
#ifdef AVEVAD
    cin.tie(0);
    ios_base::sync_with_stdio(0);
#endif
}
```

```
#endif
    ll n;
    cin >> n;
    auto [c, d] = fact_m(n + 1);
    cout << c << ' ' << d << '\n';
}
```

Task E ()

```
#ifndef AVEVAD
#define _GLIBCXX_DEBUG
#else
#pragma GCC optimize("O3")
#endif

#include <bits/stdc++.h>

using namespace std;
using ll = long long;

int main() {
#ifndef AVEVAD
    cin.tie(0);
    ios_base::sync_with_stdio(0);
#endif
    ll n;
    cin >> n;
    vector<ll> v(n);
    for (ll &e: v) cin >> e;
    ll q;
    cin >> q;
    while (q--) {
        ll l, r, c;
        cin >> l >> r >> c;
        ll ans = 0;
        for (ll i = l; i < r; i++) ans += (c + v[i] - 1) / v[i];
        cout << ans << '\n';
    }
}
```

Task F ()