

Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	80	58	31	469

Task A ()

```
#include <iostream>
#include <vector>
#include <set>

using namespace std;

typedef long long ll;

ll n;

multiset<pair<ll, ll>> ms;

int main() {
    cin.tie(0) -> sync_with_stdio(0);
    cin >> n;
    for (ll i = 0; i < n; i++) {
        ll a, b;
        cin >> a >> b;
        while (a) {
            if (a % 10) ms.insert({ b, a % 10 });
            a /= 10;
            b++;
        }
        ll last = -1;
        ll sm = 0;
        while (!ms.empty()) {
            auto v = *ms.begin();
            ms.erase(ms.begin());
            if (v.first != last) {
                if (sm % 10) break;
                if (sm != 0) {
                    ll j = 0;
                    while (sm) {
                        if (sm % 10) {
                            ms.insert({ last + j, sm % 10 });
                            if (last + j < v.first) {
                                last += j;
                                cout << last;
                                return 0;
                            }
                        }
                        sm /= 10;
                        j++;
                    }
                }
                last = v.first;
                sm = 0;
            }
            sm += v.second;
        }
        while (sm && sm % 10 == 0) {
            sm /= 10;
            last++;
        }
        cout << last;
    }
```

```
        return 0;
    }

    /*
    7
    999 15
    999 12
    999 9
    999 6
    999 3
    999 0
    1 0

    */
```

Task B ()

```
#include <iostream>
#include <vector>
#include <set>

using namespace std;

typedef long long ll;

ll n;

vector<ll> vc;

ll get() {
    string s;
    cin >> s;
    return ((ll)s.size() - 3) / 2;
}

int main() {
    cin.tie(0)->sync_with_stdio(0);
    cin >> n;
    vc.resize(n);
    for (ll& i : vc) cin >> i;
    while (vc.size() && vc.back() == 0) vc.pop_back();
    while (vc.size()) {
        cout << "Flip_and_wait" << endl;
        ll cnt = 0;
        while (cnt < vc.size()) {
            cnt += get();
            if (cnt >= vc.size()) break;
            cout << "Wait" << endl;
        }
        vc.back()--;
        while (vc.size() && vc.back() == 0) vc.pop_back();
    }
    cout << "Stop";
    return 0;
}

/*
7
999 15
999 12
999 9
999 6
999 3
999 0
1 0

*/
```

Task C ()

```
#include <iostream>
#include <vector>
#include <ctime>
#include <set>
#include <map>

using namespace std;

typedef long long ll;

ll n;

string curr = "";

vector<string> vc;
vector<string> strs;

map<string, string> mp;
map<string, string> ump;

ll ln;

void gen(ll ques) {
    if (curr.size() == ln) {
        if (ques == 0) strs.push_back(curr);
        if (ques < curr.size() / 2) return;
        vc.push_back(curr);
        return;
    }
    curr.push_back('0');
    gen(ques);
    curr.pop_back();
    curr.push_back('1');
    gen(ques);
    curr.pop_back();
    curr.push_back('?');
    gen(ques + 1);
    curr.pop_back();
}

int main() {
    cin.tie(0) -> sync_with_stdio(0);
    for (ln = 2; ln <= 10; ln++) {
        vc.clear();
        strs.clear();
        gen(0);
        for (auto& i : strs) {
            bool was = 0;
            for (auto& j : vc) {
                if (mp[j].size() != 0) continue;
                bool ok = 1;
                for (ll q = 0; q < ln; q++) {
                    if (j[q] == '?') continue;
                    if (i[q] != j[q]) {
                        ok = 0;
                        break;
                    }
                }
                if (ok) {
                    mp[j] = i;
                    ump[i] = j;
                    was = 1;
                    break;
                }
            }
            if (!was) {
                cout << "BAD!!!" << endl;
            }
        }
    }
    cin >> n;
```

```

while (n--) {
    string s;
    cin >> s;
    if (s.find('??') != string::npos) {
        if (s.size() <= 10) {
            s = mp[s];
        }
        else {
            bool has0 = false;
            ll has1 = 0;
            for (char c : s) {
                if (c == '0') has0 = 1;
                if (c == '1') has1++;
            }
            if (has0) {
                for (char& c : s) if (c == '??') c = '1';
            }
            else if (has1) {
                if (has1 == 1 && s[37] == '1') {
                    for (char& c : s) if (c == '??') c = '1';
                }
                else {
                    for (char& c : s) if (c == '??') c = '0';
                }
            }
            else {
                for (char& c : s) if (c == '??') c = '0';
            }
        }
    }
    else {
        if (s.size() <= 10) {
            s = ump[s];
        }
        else {
            ll cnt0 = 0, cnt1 = 0;
            for (char c : s) {
                if (c == '0') cnt0++;
                if (c == '1') cnt1++;
            }
            if (cnt0 > cnt1) {
                for (char& c : s) {
                    if (c == '0') c = '??';
                }
            }
            else {
                if (cnt1 == s.size()) {
                    for (ll i = 0; i < s.size(); i++) {
                        if (i != 37) s[i] = '??';
                    }
                }
                else {
                    for (char& c : s) {
                        if (c == '1') c = '??';
                    }
                }
            }
        }
    }
    cout << s << endl;
}
return 0;
}

```

```

/*
2
1??1
011??
*/

```

Task D ()

```
#include <iostream>
#include <vector>
#include <ctime>
#include <set>
#include <map>

#pragma GCC optimize("Ofast,unroll-loops,fast-math")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,avx")

using namespace std;

typedef long long ll;

const ll mod = 998244353;

ll n;

ll dp[3];

vector<ll> st;

map<ll, ll> mp;

int main() {
    cin.tie(0) -> sync_with_stdio(0);
    srand(time(0));
    st.push_back(0);
    st.push_back(1);
    for (ll i = 1; i <= 9; i++) {
        st.push_back((ll)(1e8) * i - 1);
        st.push_back((ll)(1e8) * i);
    }
    mp[0] = 1;
    mp[1] = 2;
    mp[999999999] = 808258749;
    mp[1000000000] = 980201571;
    mp[1999999999] = 117153405;
    mp[2000000000] = 468660588;
    mp[2999999999] = 761699708;
    mp[3000000000] = 478389955;
    mp[3999999999] = 573994984;
    mp[4000000000] = 137136996;
    mp[4999999999] = 62402409;
    mp[5000000000] = 203730522;
    mp[5999999999] = 511621808;
    mp[6000000000] = 533107751;
    mp[6999999999] = 242726978;
    mp[7000000000] = 496409701;
    mp[7999999999] = 887890124;
    mp[8000000000] = 637333574;
    mp[8999999999] = 875880304;
    mp[9000000000] = 465585441;
    cin >> n;
    if (mp[n] != 0) {
        cout << "0_" << mp[n];
        return 0;
    }
    ll it = lower_bound(st.begin(), st.end(), n) - st.begin();
    dp[st[it - 2] % 3] = mp[st[it - 2]];
    dp[st[it - 1] % 3] = mp[st[it - 1]];
    for (ll i = st[it - 1] + 1; i <= n; i++) {
        dp[i % 3] = (((i * i) % mod) * dp[(i - 2 + 3) % 3]) + dp[(i - 1 + 3) % 3];
        dp[i % 3] %= mod;
    }
    cout << "0_" << dp[n % 3];
    return 0;
}

/*

dp[i] * i^2 =
dp[i] * i
```

```

dp[i]

dp[i - 1] * i^2
dp[i - 1] * i
dp[i - 1]

dp[i - 2] * i^2
dp[i - 2] * i
dp[i - 2]

*/

```

Task E ()

```
#include <iostream>
#include <vector>
#include <ctime>
#include <set>
#include <map>

#pragma GCC optimize("Ofast")

using namespace std;

typedef long long ll;

const ll cst = 3e5 + 5;

const ll k = 2600;

void operator delete(void*);

ll n;

ll arr[cst];

int blans[cst / k + 1][(ll)1e6 + 1];

ll cn[k + 7];

int main() {
    cin.tie(0) -> sync_with_stdio(0);
    cin >> n;
    for (ll i = 0; i < n; i++) cin >> arr[i];
    for (ll i = 0; i <= (n - 1) / k; i++) {
        for (ll j = 0; j <= k; j++) cn[j] = 0;
        for (ll j = i * k; j < min(n, i * k + k); j++) {
            if (arr[j] <= k) cn[arr[j]]++;
            else {
                for (ll q = 0; q * arr[j] + 1 <= 1e6; q++) blans[i][q * arr[j] + 1] += 1;
            }
        }
        for (ll j = 1; j <= k; j++) {
            if (!cn[j]) continue;
            for (ll q = 0; q * j + 1 <= 1e6; q++) blans[i][q * j + 1] += cn[j];
        }
        for (ll d = 1; d <= 1e6; d++) {
            blans[i][d] += blans[i][d - 1];
        }
    }
    ll q;
    cin >> q;
    while (q--) {
        ll a, b, d;
        cin >> a >> b >> d;
        b--;
        ll abl = a / k;
        ll bbl = b / k;
        ll answ = 0;
        if (abl == bbl) {
            for (ll i = a; i <= b; i++) answ += (d + arr[i] - 1) / arr[i];
        }
        else {
            for (ll i = a; i < (abl + 1) * k; i++) answ += (d + arr[i] - 1) / arr[i];
            for (ll i = b; i >= bbl * k; i--) answ += (d + arr[i] - 1) / arr[i];
            for (ll i = abl + 1; i < bbl; i++) answ += (ll)blans[i][d];
        }
        cout << answ << "\n";
    }
    return 0;
}
```


Task F ()

```
#include <iostream>
#include <vector>
#include <ctime>
#include <set>
#include <map>

#pragma GCC optimize("Ofast,unroll-loops,fast-math")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,avx")

using namespace std;

typedef long long ll;

const ll mod = 998244353;
const ll cst = 1e6 + 6;

ll n;

ll sml[cst];
ll smr[cst];

ll answ = 1;

ll dp[cst];

int main() {
    cin.tie(0) -> sync_with_stdio(0);
    cin >> n;
    dp[0] = 1;
    for (ll i = 1; i <= n; i++) {
        cin >> sml[i] >> smr[i];
        sml[i] += sml[i - 1];
        smr[i] += smr[i - 1];
        dp[i] = 1;
        for (ll j = 0; j < i; j++) {
            if (smr[i] - smr[j] >= 0 && sml[i] - sml[j] <= 0) dp[i] = max(dp[i], dp[j]
                + 1);
        }
        answ = max(answ, dp[i]);
    }
    cout << answ;
    return 0;
}

/*
*/
```