Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	58	31	449

Task A ()

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

using namespace std;

#define int long long

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n; cin >> n;
    map<int, int> mp;
    set<int> st;
    for (int i = 0; i < n; ++i) {
        int a, b; cin >> a >> b;
        while (a % 10 == 0) {
            a /= 10;
            b++;
        }
        mp[b] += a;
        st.insert(b);
    }
    //for (auto it: mp) cout << it.first << " " << it.second << endl;
    int t = 0;
    while (true) {
        auto it = st.lower_bound(t);
        int u = (*it);
        t = u;
        int o = mp[u];
        if (o % 10 != 0) break;
        mp[u + 1] += o / 10;
        st.insert(u + 1);
        t = u + 1;
    }
    cout << t;
}
```

Task B ()

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

using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n; cin >> n;
    vector<int> a(n);
    int pos = 0;
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
        if (a[i] > 0) pos = i;
    }
    vector<int> b(n);
    int o = 0;
    int t = 1;
    cout << "Flip␣and␣wait" << endl;
    while (o <= pos) {
        string s; cin >> s;
        int m = s.size();
        int p = (m - 3) / 2;
        for (int i = o; i < o + p; ++i) b[i] = t;
        t++;
        o = o + p;
        if (o > pos) break;
        cout << "Wait" << endl;
    }
    a[pos]--;
    while (pos >= 0) {
        if (a[pos] == 0) {
            pos--;
            continue;
        }
        while (a[pos] > 0) {
            cout << "Flip␣and␣wait" << endl;
            string s; cin >> s;
            for (int i = 0; i < b[pos] - 1; ++i) {
                cout << "Wait" << endl;
                cin >> s;
            }
            a[pos]--;
        }
    }
    cout << "Stop" << endl;
    return 0;
}

/*
Beeeep!
*/
```

Task C ()

```
#include <bits/stdc++.h>
#pragma GCC optimize("no-stack-protector")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("O3")
#pragma GCC target("avx")
#pragma GCC target("avx2")

using namespace std;

#define int long long
#define double long double
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()


signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int t; cin >> t;
    string ans;
    while (t--) {
        string s; cin >> s;
        int n = s.size();
        bool f = 0;
        for (int i = 0; i < n; ++i) {
            if (s[i] == '?') {
                f = 1;
                break;
            }
        }
        ans = s;
        if (f == 0) {
            for (int i = 0; i < n - 1; i += 2) {
                if (s[i] == '0' && s[i + 1] == '0') {
                    ans[i] = '0';
                    ans[i + 1] = '?';
                }
                if (s[i] == '0' && s[i + 1] == '1') {
                    ans[i] = '?';
                    ans[i + 1] = '1';
                }
                if (s[i] == '1' && s[i + 1] == '0') {
                    ans[i] = '?';
                    ans[i + 1] = '0';
                }
                if (s[i] == '1' && s[i + 1] == '1') {
                    ans[i] = '1';
                    ans[i + 1] = '?';
                }
            }
        }
        else {
            for (int i = 0; i < n - 1; i += 2) {
                if (s[i] == '0' && s[i + 1] == '?') {
                    ans[i] = '0';
                    ans[i + 1] = '0';
                }
                if (s[i] == '?' && s[i + 1] == '1') {
                    ans[i] = '0';
                    ans[i + 1] = '1';
                }
                if (s[i] == '?' && s[i + 1] == '0') {
                    ans[i] = '1';
                    ans[i + 1] = '0';
                }
                if (s[i] == '1' && s[i + 1] == '?') {
                    ans[i] = '1';
                    ans[i + 1] = '1';
                }
            }
        }
        cout << ans << endl;
    }
}

/*
2
?01?
?1?00
*/
```

Task D ()

```
#include <bits/stdc++.h>
#pragma GCC optimize("no-stack-protector")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("O3")
#pragma GCC target("avx")
#pragma GCC target("avx2")

using namespace std;

#define int long long
#define double long double
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()

const int MOD = 998244353;

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n; cin >> n;
    int ans = 1;
    for (int i = 1; i <= n + 1; ++i) ans = (ans * i) % MOD;
    cout << 0 << "␣" << ans;
    /*for (int i = 0; i <= n; ++i) {
        cout << i << " " << dp[i] % MOD << endl;
    }*/
}

/*
2
?01?
?1?00
*/
```

Task E ()

```cpp
#include <bits/stdc++.h>
#pragma GCC optimize("no-stack-protector")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("O3")
#pragma GCC target("avx")
#pragma GCC target("avx2")

using namespace std;

#define int long long
#define double long double
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()

const int MOD = 998244353;

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n; cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; ++i) cin >> a[i];
    int q; cin >> q;
    while (q--) {
        int l, r, d; cin >> l >> r >> d;
        int ans = 0;
        for (int i = l; i < r; ++i) {
            ans += (d + a[i] - 1) / a[i];
        }
        cout << ans << '\n';
    }
}

/*
Beeeep!
*/
```

Task F ()

```
#include <bits/stdc++.h>
/*#pragma GCC optimize("no-stack-protector")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("O3")
#pragma GCC target("avx")
#pragma GCC target("avx2")*/

using namespace std;

#define int long long
#define double long double
#define all(a) a.begin(), a.end()
#define rall(a) a.rbegin(), a.rend()

const int MOD = 998244353;

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    int n; cin >> n;
    vector<pair<int, int>> vec(n);
    for (int i = 0; i < n; ++i) {
        int l, r; cin >> l >> r;
        vec[i] = {l, r};
    }
    vector<int> dp(n + 1);
    dp.back() = 1;
    for (int i = n - 1; i >= 0; --i) {
        int l = 0, r = 0;
        dp[i] = 1;
        for (int j = i + 1; j < n + 1; ++j) {
            l += vec[j - 1].first;
            r += vec[j - 1].second;
            if (l <= 0 && r >= 0) dp[i] = max(dp[i], dp[j] + 1);
        }
    }
    int mx = 1;
    for (auto it: dp) mx = max(mx, it);
    cout << mx;
}

/*
Beeeep!
*/
```