Олимпиада СПбГУ по информатике 2021/22 учебного года

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

Task A ()

```
#define _USE_MATH_DEFINES

//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("Ofast")
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime: " << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;


mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());


void solve() {
        int n;
        cin >> n;
        map<int, int> cnt;
        for (int i = 0; i < n; i++) {
                int a, b;
                cin >> a >> b;
                while (a % 10 == 0)
                        a /= 10, b++;
                cnt[b] += a;
        }
        for (auto[i, j] : cnt) {
                int b = i, a = j;
                while (a % 10 == 0)
                        a /= 10, b++;
                cnt[i] = 0;
                cnt[b] += a;
        }
        auto it = cnt.begin();
        while (it != cnt.end() && it->second == 0)
                it++;
        if (it == cnt.end())
                cout << 0 << '\n';
        else
                cout << it->first << '\n';
}

int32_t main() {
        start;

        solve();

        finish;
}
```

Task B ()

```
#define _USE_MATH_DEFINES

//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("Ofast")
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:␣" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;


mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

int a[105];

int flip() {
        cout << "Flip␣and␣wait" << endl;
        string s;
        cin >> s;
        if (s[0] != 'B' || s == "Burn")
                exit(0);
        return count(s.begin(), s.end(), 'e') / 2;
}

int wait() {
        cout << "Wait" << endl;
        string s;
        cin >> s;
        if (s[0] != 'B' || s == "Burn")
                exit(0);
        return count(s.begin(), s.end(), 'e') / 2;
}

void stop() {
        cout << "Stop" << endl;
        return;
}

void solve() {
        int n;
        cin >> n;
        for (int i = 1; i <= n; i++)
                cin >> a[i];
        int r = n + 1;
        int k = 0;
        for (int i = 1; i <= n; i++)
                k += a[i];
        while (r--) {
                while (a[r]--) {
                        int l = flip();
                        while (l < r)
                                l += wait();
                }
        }
        stop();
}

int32_t main() {
        start;

        solve();

        finish;
}
```

Task C ()

```
#define _USE_MATH_DEFINES

//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("Ofast")
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:␣" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;


mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());


void solve() {
        /*
         *  00
         *  11
         *  01
         *  10
         */
        int n;
        cin >> n;
        vector<string> str(n);
        for (auto &i : str)
                cin >> i;
        bool ok = false;
        for (auto i : str)
                ok |= count(i.begin(), i.end(), '?');
        if (!ok) {
                for (auto s : str) {
                        int m = s.size();
                        if (m & 1) {
                                vector<int> cnt(2);
                                for (char c : s)
                                        cnt[c - '0']++;
                                if (!cnt[0] || !cnt[1]) {
                                        m /= 2;
                                        for (int i = 0; i < m; i++)
                                                s[i] = '?';
                                } else if (cnt[0] > cnt[1]) {
                                        replace(s.begin(), s.end(), '0', '?');
                                } else {
                                        replace(s.begin(), s.end(), '1', '?');
                                }
                        } else {
                                vector<int> cnt(2);
                                for (char c : s)
                                        cnt[c - '0']++;
                                m /= 2;
                                if (!cnt[1]) {
                                        replace(s.begin(), s.end(), '0', '?');
                                } else if (!cnt[0]) {
                                        for (int i = 0; i < m; i++)
                                                s[i] = '?';
                                } else if (cnt[0] > cnt[1]) {
                                        replace(s.begin(), s.end(), '0', '?');
                                } else
                                        replace(s.begin(), s.end(), '1', '?');
                        }
                        cout << s << '\n';
                }
        } else {
                for (auto s : str) {
                        int m = s.size();
                        if (m & 1) {
                                vector<int> cnt(2);
                                int q = 0;
                                for (char c : s) {
                                        if (c == '?')
                                                q++;
                                        else
                                                cnt[c - '0']++;
                                }
                                m /= 2;
                                if (q == m)
                                        replace(s.begin(), s.end(), '?', (cnt[0] ? '0' : '1'));
                                else if (!cnt[0])
                                        replace(s.begin(), s.end(), '?', '0');
                                else
                                        replace(s.begin(), s.end(), '?', '1');
                        } else {
                                vector<int> cnt(2);
                                int q = 0;
                                for (char c : s) {
                                        if (c == '?')
                                                q++;
                                        else
                                                cnt[c - '0']++;
                                }
                                m /= 2;
                                if (cnt[1] == m) {
                                        replace(s.begin(), s.end(), '?', '1');
                                } else if (q == 2 * m) {
                                        replace(s.begin(), s.end(), '?', '0');
                                } else if (!cnt[0]) {
                                        replace(s.begin(), s.end(), '?', '0');
                                } else {
                                        replace(s.begin(), s.end(), '?', '1');
                                }
                        }
                        cout << s << '\n';
                }
        }
}

int32_t main() {
        start;

        solve();

        finish;
}
```

Task D ()

```
#define _USE_MATH_DEFINES

#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
#pragma GCC optimize("Ofast")
#pragma GCC optimize("no-stack-protector")
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("unswitch-loops")
#pragma GCC optimize("fast-math")
#pragma GCC optimize("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>

//#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:␣" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;


mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

const int mod = 998244353;

void solve() {
        int n;
        cin >> n;
        int fact = 1;
        for (int i = 1; i <= n + 1; i++)
                fact = fact * 1ll * i % mod;
        cout << 0 << "␣" << fact << '\n';
}

int32_t main() {
        start;

        solve();

        finish;
}
```

Task E ()

```
#define _USE_MATH_DEFINES

//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("Ofast")
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:␣" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;


mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

const int N = 5005;

int a[N];

void solve() {
        int n;
        cin >> n;
        for (int i = 1; i <= n; i++)
                cin >> a[i];
        int q;
        cin >> q;
        while (q--) {
                int l, r, d;
                cin >> l >> r >> d;
                int res = 0;
                for (int i = l + 1; i <= r; i++)
                        res += (d + a[i] - 1) / a[i];
                cout << res << '\n';
        }
}

int32_t main() {
        start;

        solve();

        finish;
}
```

Task F ()

```
#define _USE_MATH_DEFINES

//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")
//#pragma GCC optimize("Ofast")
//#pragma GCC optimize("no-stack-protector")
//#pragma GCC optimize("unroll-loops")
//#pragma GCC optimize("unswitch-loops")
//#pragma GCC optimize("fast-math")
//#pragma GCC optimize("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:␣" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;


mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

const int N = 1e6 + 5;

int n, m;
int pref_l[N], pref_r[N];
int dp[N];

//int tl[4 * N];
//int tr[4 * N];
//
//void upd(int i, int t[]) {
//      for (t[i += m]++; i > 1; i >>= 1)
//              t[i >> 1] = t[i] + t[i ^ 1];
//}
//
//int get(int l, int r, int t[]) {
//      int res = 0;
//      for (l += m, r += m; l < r; l >>= 1, r >>= 1) {
//              if (l & 1) res += t[l++];
//              if (r & 1) res += t[--r];
//      }
//      return res;
//}

void solve() {
        cin >> n;
        for (int i = 1; i <= n; i++) {
                int l, r;
                cin >> l >> r;
                pref_l[i] = pref_l[i - 1] + l;
                pref_r[i] = pref_r[i - 1] + r;
        }
//      pref_l[i] - pref_l[j] <= x[i] - x[j] <= pref_r[i] - pref_r[j];
        int ans = 1;
        dp[0] = 1;
        for (int i = 1; i <= n; i++) {
                int cur = 0;
                for (int j = 0; j <= i; j++)
                        cur = max(cur, (pref_l[i] <= pref_l[j] && pref_r[j] <= pref_r[i]) * (dp[j]
                             + 1));
                dp[i] = cur;
                ans = max(ans, cur);
        }
        cout << ans << '\n';
}

int32_t main() {
        start;

        solve();

        finish;
}
```