

Олимпиада СПбГУ по информатике 2021/22 учебного года

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

Task A ()

```
#define _USE_MATH_DEFINES

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

#define int long long
#define uint unsigned long long
#define double long double

#ifdef DIDEOSHKA
#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;

//vector input
template<typename T>
istream &operator>>(istream &is, vector<T> &vec) {
    for (auto &i : vec) {
        is >> i;
    }
    return is;
}

//vector output
template<typename T>
ostream &operator<<(ostream &os, vector<T> &vec) {
    for (T i : vec) {
        os << i << ' ';
    }
    return os;
}

//2 dimensional vector output
template<typename T>
ostream &operator<<(ostream &os, vector<vector<T>> &vec) {
    for (vector<T> i : vec) {
        os << i << '\n';
    }
    return os;
}
```

```

}

pair<int, int> zeros(int a) {
    int ans = 0;
    while (a % 10 == 0) {
        a /= 10;
        ans += 1;
    }
    return {ans, a};
}

void solve() {
    int n;
    cin >> n;
    vector<pair<int, int>> a(n);
    vector<pair<int, int>> b;
    for (int i = 0; i < n; ++i) {
        cin >> a[i].second >> a[i].first;
    }
    sort(a.begin(), a.end());
    while (true) {
        int cur = 0;
        for (int i = 0; i < n; ++i) {
            if (i != 0 && a[i].first == a[i - 1].first) {
                cur += a[i].second;
            } else {
                if (i != 0) {
                    auto [z, val] = zeros(cur);
                    b.emplace_back(a[i - 1].first + z, val);
                }
                cur = a[i].second;
            }
        }
        auto [z, val] = zeros(cur);
        b.emplace_back(a.back().first + z, val);
        sort(b.begin(), b.end());
        if (a != b) {
            swap(a, b);
            n = a.size();
            b = {};
        } else {
            cout << b[0].first << '\n';
            return;
        }
    }
}

int32_t main() {
    start;

    solve();

    finish;
}

```

Task B ()

```
#define _USE_MATH_DEFINES

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

#define int long long
#define uint unsigned long long
#define double long double

#ifdef DIDEOSHKA
#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 ;
#define finish return 0
#endif

using namespace std;

//vector input
template<typename T>
istream &operator>>(istream &is, vector<T> &vec) {
    for (auto &i : vec) {
        is >> i;
    }
    return is;
}

//vector output
template<typename T>
ostream &operator<<(ostream &os, vector<T> &vec) {
    for (T i : vec) {
        os << i << '␣';
    }
    return os;
}

//2 dimensional vector output
template<typename T>
ostream &operator<<(ostream &os, vector<vector<T>> &vec) {
    for (vector<T> i : vec) {
        os << i << '\n';
    }
    return os;
}

int times(string& str) {
    int ans = 0;
    for (int i = 0; i < str.size(); ++i) {
        if (str[i] == 'e') {
            ans += 1;
        }
    }
    return ans / 2;
}

void solve() {
    int n;
    cin >> n;

    vector<int> a(n);
```

```

cin >> a;

vector<pair<int , int>> b;
for (int i = 0; i < n; ++i) {
    if (a[i] != 0) {
        b.emplace_back(i + 1, a[i]);
    }
}

reverse(b.begin(), b.end());

string req;

for (auto [i, k] : b) {
    for (int j = 0; j < k; ++j) {
        cout << "Flip_and_wait" << endl;
        cin >> req;
        int cur = times(req);
        while (cur < i) {
            cout << "Wait" << endl;
            cin >> req;
            cur += times(req);
        }
    }
}

cout << "Stop" << endl;
}

int32_t main() {
    start;

    solve();

    finish;
}

```

Task C ()

```
#define _USE_MATH_DEFINES

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

#define int long long
#define uint unsigned long long
#define double long double

#ifdef DIEDOSHKA
#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;

//vector input
template<typename T>
istream &operator>>(istream &is, vector<T> &vec) {
    for (auto &i : vec) {
        is >> i;
    }
    return is;
}

//vector output
template<typename T>
ostream &operator<<(ostream &os, vector<T> &vec) {
    for (T i : vec) {
        os << i << '␣';
    }
    return os;
}

//2 dimensional vector output
template<typename T>
ostream &operator<<(ostream &os, vector<vector<T>> &vec) {
    for (vector<T> i : vec) {
        os << i << '\n';
    }
    return os;
}

void solve() {
    string a;
    cin >> a;
    int time = 0;
    for (int i = 0; i < a.size(); ++i) {
        if (a[i] == '?') {
            time = 1;
            break;
        }
    }
    if (!time) {
        int what_to_change = 0;
        int z = 0;
        for (int i = 0; i < a.size(); ++i) {
```

```

        if (a[i] == '0') {
            z += 1;
        }
    }
    if (a.size() - z > z) {
        what_to_change = 1;
    } else if (a.size() - z == z) {
        for (int i = 0; i < a.size(); ++i) {
            if (i != 0 && a[i] != a[i - 1]) {
                what_to_change = a[i] - '0';
                break;
            }
        }
    }
    bool seen = false;
    for (int i = 0; i < a.size(); ++i) {
        if (seen) {
            if (a[i] - '0' == what_to_change) {
                a[i] = '?';
            }
        } else {
            if (a[i] - '0' == what_to_change) {
                seen = true;
            } else {
                a[i] = '?';
            }
        }
    }
    cout << a << '\n';
} else {
    int what_to_change = 0;
    for (int i = 0; i < a.size(); ++i) {
        if (a[i] != '?') {
            what_to_change = a[i] - '0';
            break;
        }
    }
    bool seen = false;
    for (int i = 0; i < a.size(); ++i) {
        if (!seen) {
            if (a[i] != '?') {
                seen = true;
            } else {
                a[i] = '0' + (1 - what_to_change);
            }
        } else {
            if (a[i] == '?') {
                a[i] = '0' + what_to_change;
            }
        }
    }
    cout << a << '\n';
}
}

int32_t main() {
    start;

    int t;
    cin >> t;
    while (t--) {
        solve();
    }

    finish;
}

```

Task D ()

```
#define _USE_MATH_DEFINES

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

#define int unsigned long long
#define uint unsigned long long
#define double long double

#ifdef DIEDOSHKA
#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;

//vector input
template<typename T>
istream &operator>>(istream &is, vector<T> &vec) {
    for (auto &i : vec) {
        is >> i;
    }
    return is;
}

//vector output
template<typename T>
ostream &operator<<(ostream &os, vector<T> &vec) {
    for (T i : vec) {
        os << i << '␣';
    }
    return os;
}

//2 dimensional vector output
template<typename T>
ostream &operator<<(ostream &os, vector<vector<T>> &vec) {
    for (vector<T> i : vec) {
        os << i << '\n';
    }
    return os;
}

int mod = 998244353;

vector<int> a = {1, 808258749, 117153405, 761699708, 573994984, 62402409, 511621808, 242726978,
    887890124, 875880304};

int factmod(int n, int p) {
    int res = 1;
    while (n > 1) {
        res = (res * ((n / p) % 2 ? p - 1 : 1)) % p;
        res *= a[(n % p) / 100000000];
        res %= p;
        for (int i = ((n % p) / 100000000) * 100000000 + 1; i <= n % p; ++i) res = (res * i) % p;
        n /= p;
    }
}
```

```

    }
    return res % p;
}

void solve() {
    int n;
    cin >> n;

    int ans = factmod(n + 1, mod);

    cout << (n + 1) / mod + (n + 1) / (mod * mod) << ' ' << ans << '\n';
}

int32_t main() {
    start;

    solve();

    finish;
}

```


Task E ()

```
#define _USE_MATH_DEFINES

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

#define int long long
#define uint unsigned long long
#define double long double

#ifdef DIDEOSHKA
#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;

//vector input
template<typename T>
istream &operator>>(istream &is, vector<T> &vec) {
    for (auto &i : vec) {
        is >> i;
    }
    return is;
}

//vector output
template<typename T>
ostream &operator<<(ostream &os, vector<T> &vec) {
    for (T i : vec) {
        os << i << '␣';
    }
    return os;
}

//2 dimensional vector output
template<typename T>
ostream &operator<<(ostream &os, vector<vector<T>> &vec) {
    for (vector<T> i : vec) {
        os << i << '\n';
    }
    return os;
}

struct segtree {
    struct node {
        int val;
        int prop;

        node() {
            val = 0;
            prop = 0;
        }
    };

    int size;
    vector<node> tree;
```

```

segtree(int n) {
    size = 1;
    while (size < n) {
        size *= 2;
    }
    tree.resize(2 * size - 1);
}

void propagate(int x) {
    if (x > size - 2) {
        return;
    } else {
        tree[2 * x + 1].prop += tree[x].prop;
        tree[2 * x + 2].prop += tree[x].prop;
        tree[2 * x + 1].val += tree[x].prop;
        tree[2 * x + 2].val += tree[x].prop;

        tree[x].prop = 0;
    }
}

void set(int l, int r, int v, int x, int lx, int rx) {
    if (rx <= l || r <= lx) {
        return;
    }
    propagate(x);
    if (l <= lx && rx <= r) {
        tree[x].prop += v;
        tree[x].val += v;
        return;
    }
    int m = (lx + rx) / 2;
    set(l, r, v, 2 * x + 1, lx, m);
    set(l, r, v, 2 * x + 2, m, rx);
    tree[x].val = min(tree[2 * x + 1].val, tree[2 * x + 2].val);
}

void set(int l, int r, int v) {
    set(l, r, v, 0, 0, size);
}

int get(int l, int r, int x, int lx, int rx) {
    if (rx <= l || r <= lx) {
        return 0;
    }
    propagate(x);
    if (l <= lx && rx <= r) {
        return tree[x].val;
    }
    int m = (lx + rx) / 2;
    int v1 = get(l, r, 2 * x + 1, lx, m);
    int v2 = get(l, r, 2 * x + 2, m, rx);
    return v1 + v2;
}

int get(int l, int r) {
    return get(l, r, 0, 0, size);
}

};

void solve() {
    int n;
    cin >> n;
    vector<int> v(n);
    cin >> v;
    int q;
    cin >> q;
    while (q--) {
        int a, b, d;
        cin >> a >> b >> d;
        int ans = 0;
        for (int i = a; i < b; ++i) {
            ans += (d + v[i] - 1) / v[i];
        }
    }
}

```

```
        }
        cout << ans << '\n';
    }

}

int32_t main() {
    start;

    solve();

    finish;
}
```

Task F ()