

# Олимпиада СПбГУ по информатике 2021/22 учебного года

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

## Task A ()

```
//#pragma comment(linker, "/STACK:67108864")
#ifdef LOCAL
#define _GLIBCXX_DEBUG
#endif
#pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target("avx2")
#include <iostream>
#include <vector>
#include <random>
#include <algorithm>
#include <set>
#include <queue>
#include <deque>
#include <map>
#include <string>
#include <iomanip>
#include <cstring>
#include <unordered_map>
#include <bitset>
#include <stack>
#include <numeric>
#include <unordered_set>
#include <ctime>
#include <cmath>

using namespace std;

typedef long long ll;

#define pb push_back
#define mp make_pair
#define int ll

const int BUBEN = 600;
const int MOD = 1e9 + 7;
const int BASE = 29;
const int MOD1 = 998244353;
const int BASE1 = 31;

#ifdef _getchar_nolock
char _getchar_nolock()
{
    return getchar_unlocked();
}
#endif

#ifdef _putchar_nolock
char _putchar_nolock(char i)
{
    return putchar_unlocked(i);
}
#endif

#ifdef LOCAL
#define dbg(x) cout << #x << " : " << (x) << "\n";
```

```

#else
#define dbg(x)
#endif

ostream &operator<<(ostream &a, const vector<int> &b)
{
    for (auto k : b)
        cout << k << " ";
    return a;
}

inline int fasti()
{
    int a = 0;
    char c = ' ';
    while (!isdigit(c))
    {
        c = _getchar_nolock();
    }
    while (isdigit(c))
    {
        a = a * 10 + (c - '0');
        c = _getchar_nolock();
    }
    return a;
}

signed main()
{
#ifdef DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    map<int, int> arr;
    int n;
    cin >> n;
    for (int _ = 0; _ < n; _++) {
        ll a, b;
        cin >> a >> b;
        int i = 2 * 1e9 - b;
        bool flag = false;
        while (a || flag) {
            arr[i] += a % 10;
            if (arr[i] / 10 != 0) {
                flag = true;
            } else {
                flag = false;
            }
            arr[i - 1] += arr[i] / 10;
            arr[i] %= 10;
            a /= 10;
            i--;
        }
    }

    int gg = 2 * 1e9 + 1;
    int ans = 0;
    auto it = arr.rbegin();
    while (true) {
        ans += gg - it->first - 1;
        if (it->second == 0) {
            gg = it->first;
            ans++;
            it++;
        } else {
            break;
        }
    }
}

```

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

## Task B ()

```
#ifndef DEBUG
// #define _GLIBCXX_DEBUG
#endif
#pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target("avx2")
// #include <bits/stdc++.h>
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <cmath>
#include <set>
#include <map>
#include <iomanip>
#include <limits>
#include <stack>
#include <numeric>
#include <queue>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <cstring>
#include <chrono>
// #include <ext/pb_ds/tree_policy.hpp>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;
// using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
// typedef tree<pair<ll, ll>, null_type, less<pair<ll, ll>>, rb_tree_tag,
//     tree_order_statistics_node_update> ordered_set;

// #define int ll
#define intn int

#ifdef DEBUG
mt19937 rnd('r' + 'o' + 'b' + 'i' + 'v' + 'i' + 'r' + 't');
#else
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
#endif
const int BUBEN = 400;
const int BUBEN3D = 2100;
const int MOD = (int)1e9 + 7;
const int BASE1 = 179;
const int MOD1 = (int)998244353;
const int BASE2 = 239;
const int MOD2 = (int)1e9 + 9;

#ifdef _getchar_nolock
char _getchar_nolock() {
    return getchar_unlocked();
}
#endif

inline ll gcd(ll a, ll b) {
    while (b) {
        a %= b;
        swap(a, b);
    }
    return a;
}

inline int bin_pow(int a, int n, int mod) {
    if (n == 0) {
        return 1;
    }
}
```

```

    if (n % 2 == 0) {
        int b = bin_pow(a, n / 2, mod);
        return (11)b * b % mod;
    }
    else {
        return (11)a * bin_pow(a, n - 1, mod) % mod;
    }
}

signed main() {
#ifdef DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int n;
    cin >> n;
    vector<int> arr(n + 1);
    for (int i = 1; i <= n; i++) {
        cin >> arr[i];
    }
    int idx = n;
    while (arr[idx] == 0) {
        idx--;
    }
    cout << "Flip_and_wait" << endl;
    int temp = 0;
    while (true) {
        string s;
        cin >> s;
        if (s == "Burn") {
            return 0;
        }
        if (s == "Tired") {
            return 0;
        }
        if (s == "Failed") {
            return 0;
        }

        int gg = count(s.begin(), s.end(), 'e') / 2;
        if (temp + gg >= idx) {
            temp = 0;
            arr[idx]--;
            while (idx >= 1 && arr[idx] == 0) {
                idx--;
            }
            if (idx == 0) {
                cout << "Stop" << endl;
                return 0;
            }
            cout << "Flip_and_wait" << endl;
        } else {
            temp += gg;
            cout << "Wait" << endl;
        }
    }
    return 0;
}

```

## Task C ()

```
#ifdef DEBUG
// #define _GLIBCXX_DEBUG
#endif
#pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target("avx2")
// #include <bits/stdc++.h>
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <cmath>
#include <set>
#include <map>
#include <iomanip>
#include <climits>
#include <stack>
#include <numeric>
#include <queue>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <cstring>
#include <chrono>
// #include <ext/pb_ds/tree_policy.hpp>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;
// using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
// typedef tree<pair<ll, ll>, null_type, less<pair<ll, ll>>, rb_tree_tag,
//     tree_order_statistics_node_update> ordered_set;

// #define int ll
#define intn int

#ifdef DEBUG
mt19937 rnd('r' + 'o' + 'b' + 'i' + 'v' + 'i' + 'r' + 't');
#else
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
#endif
const int BUBEN = 400;
const int BUBEN3D = 2100;
const int MOD = (int)1e9 + 7;
const int BASE1 = 179;
const int MOD1 = (int)998244353;
const int BASE2 = 239;
const int MOD2 = (int)1e9 + 9;

#ifdef _getchar_nolock
char _getchar_nolock() {
    return getchar_unlocked();
}
#endif

inline ll gcd(ll a, ll b) {
    while (b) {
        a %= b;
        swap(a, b);
    }
    return a;
}

inline int bin_pow(int a, int n, int mod) {
    if (n == 0) {
        return 1;
    }
    if (n % 2 == 0) {
        int b = bin_pow(a, n / 2, mod);
        return (ll)b * b % mod;
    }
}
```

```

    }
    else {
        return (11)a * bin_pow(a, n - 1, mod) % mod;
    }
}

signed main() {
#ifdef DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    int n;
    cin >> n;
    for (int _ = 0; _ < n; _++) {
        string s;
        cin >> s;
        if (find(s.begin(), s.end(), '?') != s.end()) {
            if (s.size() % 2 == 0) {
                int cnt_0 = 0, cnt_1 = 0;
                for (int i = 0; i < s.size(); i++) {
                    if (s[i] == '1') {
                        cnt_1++;
                    } else if (s[i] == '0') {
                        cnt_0++;
                    }
                }
                if (cnt_1 == 0 && cnt_0 == 0) {
                    for (int i = 0; i < s.size(); i++) {
                        cout << 0;
                    }
                    cout << '\n';
                }
                else if (cnt_1 == s.size() / 2) {
                    for (int i = 0; i < s.size(); i++) {
                        cout << 1;
                    }
                    cout << '\n';
                }
                else if (cnt_0 == s.size() / 2) {
                    for (int i = 0; i < s.size(); i++) {
                        if (s[i] == '?') {
                            cout << 1;
                        } else {
                            cout << 0;
                        }
                    }
                    cout << '\n';
                }
                else if (cnt_1 != 0) {
                    for (int i = 0; i < s.size(); i++) {
                        if (s[i] == '?') {
                            cout << 0;
                        } else {
                            cout << 1;
                        }
                    }
                    cout << '\n';
                }
                else if (cnt_0 != 0) {
                    for (int i = 0; i < s.size(); i++) {
                        if (s[i] == '?') {
                            cout << 1;
                        } else {
                            cout << 0;
                        }
                    }
                    cout << '\n';
                }
            }
        } else {
            if (count(s.begin(), s.end(), '?') == s.size() / 2) {
                if (find(s.begin(), s.end(), '0') != s.end()) {
                    for (int i = 0; i < s.size(); i++) {
                        cout << '0';
                    }
                }
            }
        }
    }
}

```

```

        }
        cout << '\n';
    } else {
        for (int i = 0; i < s.size(); i++) {
            cout << '1';
        }
        cout << '\n';
    }
} else {
    if (find(s.begin(), s.end(), '0') != s.end()) {
        for (int i = 0; i < s.size(); i++) {
            if (s[i] == '?') {
                cout << '1';
            } else {
                cout << '0';
            }
        }
        cout << '\n';
    } else {
        for (int i = 0; i < s.size(); i++) {
            if (s[i] == '?') {
                cout << '0';
            } else {
                cout << '1';
            }
        }
        cout << '\n';
    }
}

} else {
    if (s.size() % 2 == 0) {
        if (find(s.begin(), s.end(), '0') == s.end()) {
            for (int i = 0; i < s.size() / 2; i++) {
                cout << '?';
            }
            for (int i = 0; i < s.size() / 2; i++) {
                cout << 1;
            }
            cout << '\n';
        } else if (find(s.begin(), s.end(), '1') == s.end()) {
            for (int i = 0; i < s.size(); i++) {
                cout << '?';
            }
            cout << '\n';
        } else {
            int cnt_0 = count(s.begin(), s.end(), '0'), cnt_1 = count(
                s.begin(), s.end(), '1');
            if (cnt_0 == s.size() / 2 || cnt_1 > cnt_0) {
                for (int i = 0; i < s.size(); i++) {
                    if (s[i] == '1') {
                        cout << '?';
                    } else {
                        cout << '0';
                    }
                }
                cout << '\n';
            } else {
                for (int i = 0; i < s.size(); i++) {
                    if (s[i] == '0') {
                        cout << '?';
                    } else {
                        cout << '1';
                    }
                }
                cout << '\n';
            }
        }
    }
} else {
    if (count(s.begin(), s.end(), '0') == 0) {
        for (int i = 0; i < s.size() / 2; i++) {
            cout << '?';
        }
    }
}

```



```

        for (int i = 0; i < s.size() / 2 + 1; i++) {
            cout << '1';
        }
        cout << '\n';
    } else if (count(s.begin(), s.end(), '1') == 0) {
        for (int i = 0; i < s.size() / 2; i++) {
            cout << '?';
        }
        for (int i = 0; i < s.size() / 2 + 1; i++) {
            cout << '0';
        }
        cout << '\n';
    } else {
        int cnt_0 = count(s.begin(), s.end(), '0'), cnt_1 = count(
            s.begin(), s.end(), '1');
        if (cnt_0 > cnt_1) {
            for (int i = 0; i < s.size(); i++) {
                if (s[i] == '0') {
                    cout << '?';
                } else {
                    cout << '1';
                }
            }
            cout << '\n';
        } else {
            for (int i = 0; i < s.size(); i++) {
                if (s[i] == '1') {
                    cout << '?';
                } else {
                    cout << '0';
                }
            }
            cout << '\n';
        }
    }
}

}

}

return 0;

}

```

## Task D ()

```
#ifndef DEBUG
// #define _GLIBCXX_DEBUG
#endif
#pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target("avx2")
// #include <bits/stdc++.h>
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <cmath>
#include <set>
#include <map>
#include <iomanip>
#include <limits>
#include <stack>
#include <numeric>
#include <queue>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <cstring>
#include <chrono>
// #include <ext/pb_ds/tree_policy.hpp>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;
// using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
// typedef tree<pair<ll, ll>, null_type, less<pair<ll, ll>>, rb_tree_tag,
//     tree_order_statistics_node_update> ordered_set;

#define int ll
#define itn int

signed main() {
#ifdef DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    vector<pair<int, int>> arr = {{1,1},{1000001,14807739},{2000001,463460762},{
3000001,125268743},{4000001,264160517},{5000001,341291215},{6000001,160829841
},{7000001,72498452},{8000001,989127312},{9000001,207970614},{10000001,
173822423},{11000001,171659315},{12000001,939147649},{13000001,356827636},{
14000001,730323668},{15000001,514076122},{16000001,7075170},{
{17000001,146676681},{18000001,783223634},{19000001,35019018},{20000001,
368110870},{21000001,345475376},{22000001,932180438},{23000001,194778784},{
24000001,176172658},{25000001,710969195},{26000001,112270242},{27000001,
285447219},{28000001,351128221},{29000001,812494002},{30000001,728792917},{
31000001,332433918},{32000001,590649395},{33000001,500306967},
{34000001,160043317},{35000001,234858501},{36000001,888178480},{37000001,
71599854},{38000001,575117861},{39000001,8892285},{40000001,909745172},{
41000001,166337878},{42000001,254845485},{43000001,102651850},{44000001,
81173820},{45000001,338568645},{46000001,972044428},{47000001,32547729},{
48000001,218290686},{49000001,315495999},{50000001,940686275},{51000001,
585779667},{52000001,486779240},{53000001,841985084},{54000001,780003811},{
55000001,147376152},{56000001,282646031},{57000001,365985522},
{58000001,298224204},{59000001,202463472},{60000001,728363202},{61000001,
825804069},{62000001,640313745},{63000001,339072532},{64000001,686964228},{
65000001,481509669},{66000001,596212159},{67000001,230485308},{68000001,
96239363},{69000001,571963813},{70000001,58968742},{71000001,128633660},{
72000001,221849002},{73000001,229404931},{74000001,845570124},{75000001,
657795506},{76000001,910909849},{77000001,744293777},{78000001,834930017},{
```

79000001,717715868},{80000001,603169432},{81000001,171371983},  
{82000001,240902297},{83000001,261026834},{84000001,833296793},{85000001,  
290187583},{86000001,750931595},{87000001,304755063},{88000001,117426843},{  
89000001,459515185},{90000001,806899565},{91000001,309149895},{92000001,  
123699964},{93000001,213862988},{94000001,672038708},{95000001,955709555},{  
96000001,684009984},{97000001,227249674},{98000001,133448249},{99000001,  
426237300},{100000001,980201571},{101000001,916629813},{102000001,575901340},{  
103000001,825335462},{104000001,949957958},{105000001,572811078},  
{106000001,383987902},{107000001,387021680},{108000001,325782559},{109000001,  
321668323},{110000001,97200622},{111000001,683741014},{112000001,206995501},{  
113000001,650635777},{114000001,50115473},{115000001,382158863},{116000001,  
67710359},{117000001,30308985},{118000001,636580596},{119000001,794217385},{  
120000001,435244866},{121000001,663280842},{122000001,995768335},{123000001,  
723639417},{124000001,563450877},{125000001,885457686},{126000001,825574252},{  
127000001,69962598},{128000001,445918469},{129000001,201208044},  
{130000001,632736954},{131000001,249403008},{132000001,212514148},{133000001,  
550225538},{134000001,812743880},{135000001,206670249},{136000001,887451876},{  
137000001,919220911},{138000001,341280025},{139000001,959892709},{140000001,  
462713753},{141000001,880728004},{142000001,348020169},{143000001,686504582},{  
144000001,474587561},{145000001,453188897},{146000001,560134935},{147000001,  
33698416},{148000001,442966231},{149000001,210214038},{150000001,725081129},{  
151000001,620429782},{152000001,198707876},{153000001,72475415},  
{154000001,678679675},{155000001,916066375},{156000001,627186474},{157000001,  
614559189},{158000001,945554620},{159000001,227446987},{160000001,952289495},{  
161000001,432108918},{162000001,117922013},{163000001,806263634},{164000001,  
104822046},{165000001,761245345},{166000001,264304512},{167000001,760379621},{  
168000001,725661552},{169000001,432135969},{170000001,389713622},{171000001,  
604363195},{172000001,783971899},{173000001,206863014},{174000001,689501721},{  
175000001,611740165},{176000001,642071430},{177000001,288800917},  
{178000001,768465637},{179000001,773043367},{180000001,306796951},{181000001,  
984307017},{182000001,393256018},{183000001,779493043},{184000001,669775715},{  
185000001,224666166},{186000001,729101672},{187000001,226631990},{188000001,  
484666513},{189000001,859985250},{190000001,750944817},{191000001,781349442},{  
192000001,818930768},{193000001,346853402},{194000001,562409136},{195000001,  
378310458},{196000001,191667031},{197000001,881278630},{198000001,158159534},{  
199000001,592806327},{200000001,468660588},{201000001,497152107},  
{202000001,767973594},{203000001,126315112},{204000001,411965281},{205000001,  
748225524},{206000001,750963946},{207000001,276045403},{208000001,650804175},{  
209000001,409986676},{210000001,962985762},{211000001,619266754},{212000001,  
477947583},{213000001,618014771},{214000001,65046985},{215000001,867582315},{  
216000001,348141710},{217000001,737416829},{218000001,695819661},{219000001,  
878354356},{220000001,167058979},{221000001,695526801},{222000001,997253417},{  
223000001,146526945},{224000001,549728433},{225000001,576748404},  
{226000001,176532702},{227000001,53404241},{228000001,681541649},{229000001,  
316282822},{230000001,773305926},{231000001,279658437},{232000001,261094014},{  
233000001,958765392},{234000001,750293594},{235000001,842366942},{236000001,  
33824975},{237000001,615393929},{238000001,407256497},{239000001,61953560},{  
240000001,401736800},{241000001,134483850},{242000001,643974287},{243000001,  
459211688},{244000001,727867449},{245000001,836850021},{246000001,150670349},{  
247000001,866035694},{248000001,990902445},{249000001,435829416},  
{250000001,305731305},{251000001,994146225},{252000001,377039745},{253000001,  
55657321},{254000001,480354658},{255000001,707007463},{256000001,454275392},{  
257000001,227340523},{258000001,415230423},{259000001,208230477},{260000001,  
793005995},{261000001,644963044},{262000001,505477200},{263000001,284734810},{  
264000001,669617024},{265000001,584436937},{266000001,190163904},{267000001,  
101933587},{268000001,156532615},{269000001,697659503},{270000001,226922511},{  
271000001,736804517},{272000001,588498865},{273000001,35939900},  
{274000001,239431911},{275000001,225395341},{276000001,117348898},{277000001,  
981334889},{278000001,249498952},{279000001,42227333},{280000001,257519013},{  
281000001,714491443},{282000001,190458042},{283000001,360784798},{284000001,  
387523251},{285000001,714070967},{286000001,720278492},{287000001,960216352},{  
288000001,729487859},{289000001,222916869},{290000001,505765176},{291000001,  
959488895},{292000001,183521508},{293000001,684826455},{294000001,48860387},{  
295000001,712236918},{296000001,549092585},{297000001,875102336},  
{298000001,818753082},{299000001,444166594},{300000001,478389955},{301000001,  
700933652},{302000001,672977957},{303000001,237951160},{304000001,340886395},{  
305000001,274369303},{306000001,22392924},{307000001,928432106},{308000001,  
995485864},{309000001,489644536},{310000001,498022055},{311000001,649255883},{  
312000001,299608970},{313000001,9264480},{314000001,270414245},{315000001,  
638129043},{316000001,567896289},{317000001,613490768},{318000001,796774261},{  
319000001,906056675},{320000001,377480642},{321000001,446602882},  
{322000001,353770084},{323000001,193605339},{324000001,68813561},{325000001,  
316594896},{326000001,314021239},{327000001,525145085},{328000001,383667217},{  
329000001,161279369},{330000001,730006137},{331000001,851459045},{332000001,

929154949},{333000001,768227162},{334000001,939707311},{335000001,574495987},{  
 336000001,981002475},{337000001,831390190},{338000001,158341521},{339000001,  
 17324804},{340000001,429985953},{341000001,14835918},{342000001,940910752},{  
 343000001,945352960},{344000001,974530800},{345000001,895760180},{  
 {346000001,765746799},{347000001,171457493},{348000001,188100288},{349000001,  
 163365458},{350000001,929742400},{351000001,629732952},{352000001,968460781},{  
 353000001,337362104},{354000001,86427459},{355000001,331936801},{356000001,  
 347931790},{357000001,725020274},{358000001,125856841},{359000001,544219811},{  
 360000001,942399238},{361000001,879157741},{362000001,912672488},{363000001,  
 966571363},{364000001,762945607},{365000001,934023714},{366000001,156507923},{  
 367000001,527762106},{368000001,792093007},{369000001,388471222},{  
 {370000001,334667621},{371000001,808941342},{372000001,854892002},{373000001,  
 621765608},{374000001,352006962},{375000001,541249165},{376000001,175033156},{  
 377000001,188803875},{378000001,604441742},{379000001,651126382},{380000001,  
 694814274},{381000001,226291470},{382000001,867168949},{383000001,413524405},{  
 384000001,96330986},{385000001,173460517},{386000001,273410874},{387000001,  
 987138235},{388000001,132930144},{389000001,896266650},{390000001,608960863},{  
 391000001,30545912},{392000001,837336326},{393000001,524319188},{  
 {394000001,567537347},{395000001,225105320},{396000001,961610520},{397000001,  
 362544137},{398000001,360379479},{399000001,905447929},{400000001,137136996},{  
 401000001,417484033},{402000001,769405169},{403000001,894722571},{404000001,  
 176840411},{405000001,604575398},{406000001,558224396},{407000001,513907752},{  
 408000001,971516264},{409000001,153672972},{410000001,479544924},{411000001,  
 114985691},{412000001,650254511},{413000001,179763302},{414000001,41326820},{  
 415000001,665555705},{416000001,689203790},{417000001,366019193},{  
 {418000001,960666807},{419000001,985229361},{420000001,312597254},{421000001,  
 212567132},{422000001,968942021},{423000001,873817399},{424000001,570602391},{  
 425000001,932308513},{426000001,142343033},{427000001,179685836},{428000001,  
 315629680},{429000001,592821571},{430000001,827891083},{431000001,800775015},{  
 432000001,531374313},{433000001,263187709},{434000001,209069939},{435000001,  
 16913126},{436000001,555652847},{437000001,986942204},{438000001,632745561},{  
 439000001,310906477},{440000001,520504134},{441000001,436784117},{  
 {442000001,231861875},{443000001,236622627},{444000001,215466405},{445000001,  
 222422064},{446000001,937065791},{447000001,366203083},{448000001,115409352},{  
 449000001,288326211},{450000001,31562832},{451000001,161313980},{452000001,  
 789406631},{453000001,648432734},{454000001,348284847},{455000001,580191819},{  
 456000001,871918525},{457000001,373217643},{458000001,385648430},{459000001,  
 674336469},{460000001,832193638},{461000001,277508567},{462000001,465540128},{  
 463000001,611819056},{464000001,719566693},{465000001,285575016},{  
 {466000001,16105198},{467000001,671926423},{468000001,444128439},{469000001,  
 780627624},{470000001,10454366},{471000001,899570459},{472000001,619879623},{  
 473000001,222360149},{474000001,731454},{475000001,111399417},{476000001,  
 11435747},{477000001,755247462},{478000001,702283892},{479000001,297832719},{  
 480000001,848786248},{481000001,609596491},{482000001,904936207},{483000001,  
 205759222},{484000001,773725876},{485000001,108129912},{486000001,574501050},{  
 487000001,571944419},{488000001,263830967},{489000001,795798442},{  
 {490000001,944202821},{491000001,922790298},{492000001,379466693},{493000001,  
 288367109},{494000001,401679205},{495000001,337629933},{496000001,892099431},{  
 497000001,326393825},{498000001,485965834},{499000001,803815986},{500000001,  
 203730522},{501000001,322208073},{502000001,564471761},{503000001,19804456},{  
 504000001,648952361},{505000001,588036330},{506000001,176457252},{507000001,  
 425050727},{508000001,695001061},{509000001,717826392},{510000001,931128746},{  
 511000001,19823240},{512000001,501946273},{513000001,407414725},{  
 {514000001,33790760},{515000001,447644318},{516000001,509998953},{517000001,  
 475237855},{518000001,275460089},{519000001,590068912},{520000001,918211364},{  
 521000001,270374833},{522000001,977664598},{523000001,504762016},{524000001,  
 907298299},{525000001,690905763},{526000001,561538160},{527000001,191429996},{  
 528000001,561177251},{529000001,131844744},{530000001,755823249},{531000001,  
 352083813},{532000001,628481163},{533000001,739062890},{534000001,228625659},{  
 535000001,185698138},{536000001,6567070},{537000001,706829302},{  
 {538000001,588662521},{539000001,122207806},{540000001,842049192},{541000001,  
 586598296},{542000001,358209864},{543000001,230381677},{544000001,297065501},{  
 545000001,613141302},{546000001,235596942},{547000001,289813054},{548000001,  
 315641198},{549000001,585943739},{550000001,471944586},{551000001,873291585},{  
 552000001,401846893},{553000001,92153505},{554000001,724769066},{555000001,  
 888375191},{556000001,792190992},{557000001,368679520},{558000001,451346603},{  
 559000001,538147465},{560000001,860427879},{561000001,650615677},{  
 {562000001,869068381},{563000001,310184676},{564000001,892296121},{565000001,  
 80674379},{566000001,67657250},{567000001,412680404},{568000001,895023863},{  
 569000001,338056977},{570000001,921406872},{571000001,350019845},{572000001,  
 58131214},{573000001,821111269},{574000001,891608622},{575000001,708967389},{  
 576000001,78650826},{577000001,623500731},{578000001,84495944},{579000001,  
 301597919},{580000001,863832845},{581000001,914456922},{582000001,237537984},{  
 583000001,205903791},{584000001,428599292},{585000001,854897444},

{586000001,337443887},{587000001,388095730},{588000001,511407870},{589000001,623868289},{590000001,62677208},{591000001,540235251},{592000001,384096497},{593000001,467364158},{594000001,547876787},{595000001,16739188},{596000001,662504644},{597000001,10181251},{598000001,935702766},{599000001,390819031},{600000001,533107751},{601000001,225421091},{602000001,230493532},{603000001,899035509},{604000001,249227007},{605000001,520842719},{606000001,87527623},{607000001,81582055},{608000001,554930919},{609000001,101932309},{610000001,204036681},{611000001,5714483},{612000001,799648572},{613000001,686943540},{614000001,853601175},{615000001,694746269},{616000001,870114426},{617000001,846200241},{618000001,598614440},{619000001,382062082},{620000001,780792325},{621000001,506639988},{622000001,690380686},{623000001,469887125},{624000001,408510978},{625000001,783128943},{626000001,575421895},{627000001,822410447},{628000001,597434869},{629000001,232069571},{630000001,543315647},{631000001,586989140},{632000001,509833125},{633000001,463108068},{634000001,521345135},{635000001,554573551},{636000001,206404232},{637000001,536047195},{638000001,789569870},{639000001,316818969},{640000001,537861593},{641000001,972674463},{642000001,308046329},{643000001,815100233},{644000001,703465500},{645000001,206175141},{646000001,708596179},{647000001,331193067},{648000001,885802577},{649000001,602250676},{650000001,974689523},{651000001,104185819},{652000001,491022779},{653000001,702154511},{654000001,868097590},{655000001,355702149},{656000001,171368704},{657000001,547070116},{658000001,232570701},{659000001,614029416},{660000001,656147161},{661000001,65002916},{662000001,526329826},{663000001,523742722},{664000001,740040992},{665000001,702216888},{666000001,435017492},{667000001,421255669},{668000001,39414063},{669000001,202204615},{670000001,389920518},{671000001,835476157},{672000001,316182491},{673000001,174450437},{674000001,119039947},{675000001,681855716},{676000001,347518394},{677000001,14421413},{678000001,716991197},{679000001,558039436},{680000001,499793872},{681000001,723569502},{682000001,512178515},{683000001,732012564},{684000001,433543613},{685000001,502605266},{686000001,172110771},{687000001,521977781},{688000001,876013460},{689000001,648555363},{690000001,991251920},{691000001,891506835},{692000001,193095207},{693000001,247325939},{694000001,99040868},{695000001,397369921},{696000001,125271187},{697000001,923488557},{698000001,785144462},{699000001,312024165},{700000001,496409701},{701000001,980495448},{702000001,174499363},{703000001,109371590},{704000001,523676795},{705000001,210331064},{706000001,148786412},{707000001,3941659},{708000001,517787375},{709000001,8549362},{710000001,686040095},{711000001,709921324},{712000001,269201407},{713000001,111356268},{714000001,549811072},{715000001,132444006},{716000001,612901057},{717000001,464354508},{718000001,139845621},{719000001,950458154},{720000001,486177373},{721000001,588014556},{722000001,985517432},{723000001,72917930},{724000001,494840595},{725000001,662754159},{726000001,821966960},{727000001,963597471},{728000001,942241508},{729000001,442638077},{730000001,541558298},{731000001,231902618},{732000001,340512912},{733000001,805160266},{734000001,522216305},{735000001,339157691},{736000001,909587406},{737000001,587942988},{738000001,457427194},{739000001,267723486},{740000001,798751670},{741000001,427955454},{742000001,391457548},{743000001,760672008},{744000001,346215053},{745000001,513007622},{746000001,305481069},{747000001,185432146},{748000001,224608650},{749000001,87790954},{750000001,751314989},{751000001,766099991},{752000001,656847428},{753000001,130455367},{754000001,860219614},{755000001,623260861},{756000001,330881279},{757000001,724696124},{758000001,454246098},{759000001,41946526},{760000001,18752892},{761000001,625726144},{762000001,814699327},{763000001,712074121},{764000001,844152212},{765000001,814071994},{766000001,642372452},{767000001,299654650},{768000001,316857004},{769000001,486109028},{770000001,789010703},{771000001,750967264},{772000001,860360227},{773000001,180024804},{774000001,203122977},{775000001,516741650},{776000001,834068176},{777000001,982099229},{778000001,37760744},{779000001,496883356},{780000001,484813494},{781000001,872800834},{782000001,72405177},{783000001,411445636},{784000001,7096622},{785000001,809150830},{786000001,474698225},{787000001,942353812},{788000001,378638279},{789000001,344169230},{790000001,54678264},{791000001,339738433},{792000001,658505231},{793000001,15783210},{794000001,740984897},{795000001,352783918},{796000001,612944478},{797000001,4680349},{798000001,91064729},{799000001,240258964},{800000001,637333574},{801000001,832502215},{802000001,353392515},{803000001,115386315},{804000001,392532864},{805000001,609464676},{806000001,368360833},{807000001,967964422},{808000001,184340169},{809000001,873151486},{810000001,917413882},{811000001,279862638},{812000001,741295442},{813000001,737941833},{814000001,770657828},{815000001,252236546},{816000001,678476475},{817000001,308159414},{818000001,409671283},{819000001,862556388},{820000001,1828475},{821000001,503090239},{822000001,189390539},{823000001,155045583},{824000001,606571313},{825000001,346669129},{826000001,791218520},{827000001,271497649},{828000001,155355704},{829000001,104597024},{830000001,960973213},{831000001,163867964},{832000001,992535519},{833000001,279076173},{834000001,347957802},{835000001,852970009},{836000001,243616803},{837000001,693535560},{838000001,344041666},{839000001,745536390},{

```

840000001,3376095},{841000001,671148649},{842000001,412051034},{843000001,
841048695},{844000001,873349535},{845000001,931810032},{846000001,350238512},{
847000001,793991932},{848000001,231794386},{849000001,796062071},
{850000001,317598805},{851000001,925045624},{852000001,280876609},{853000001,
76023866},{854000001,242195523},{855000001,369355018},{856000001,504568596},{
857000001,297160223},{858000001,194578676},{859000001,36506887},{860000001,
370714038},{861000001,124321961},{862000001,286693148},{863000001,303815079},{
864000001,159406212},{865000001,906524158},{866000001,406979971},{867000001,
159732505},{868000001,969712728},{869000001,138812926},{870000001,259952544},{
871000001,472672734},{872000001,924414898},{873000001,343463818},
{874000001,864499185},{875000001,549403346},{876000001,245572270},{877000001,
333953910},{878000001,878484554},{879000001,33867293},{880000001,188740734},{
881000001,23555707},{882000001,628964038},{883000001,903164311},{884000001,
305089234},{885000001,42778522},{886000001,584101103},{887000001,213267989},{
888000001,192377377},{889000001,250285143},{890000001,924749239},{891000001,
428107065},{892000001,531300988},{893000001,437614108},{894000001,328369120},{
895000001,279793201},{896000001,819249790},{897000001,876660423},
{898000001,129347658},{899000001,506521619},{900000001,465585441},{901000001,
373498856},{902000001,609594774},{903000001,6902044},{904000001,969819702},{
905000001,335338505},{906000001,336316740},{907000001,401062250},{908000001,
597265322},{909000001,358348959},{910000001,980080062},{911000001,946731403},{
912000001,708817978},{913000001,166019409},{914000001,700309550},{915000001,
187530408},{916000001,463733637},{917000001,366673842},{918000001,259933973},{
919000001,295188098},{920000001,298192565},{921000001,438225318},
{922000001,634039358},{923000001,275972826},{924000001,211321541},{925000001,
605905235},{926000001,15016685},{927000001,582967202},{928000001,946445766},{
929000001,103695579},{930000001,292412660},{931000001,542746755},{932000001,
703826599},{933000001,761324247},{934000001,135862376},{935000001,636623938},{
936000001,232875199},{937000001,229791994},{938000001,402002074},{939000001,
141152369},{940000001,307792219},{941000001,710245714},{942000001,118312417},{
943000001,299904968},{944000001,767685278},{945000001,875353981},
{946000001,751024348},{947000001,733002939},{948000001,926327575},{949000001,
320832805},{950000001,898815363},{951000001,905022275},{952000001,628341290},{
953000001,984044983},{954000001,450296129},{955000001,660292084},{956000001,
225422152},{957000001,943608076},{958000001,461319423},{959000001,217571256},{
960000001,162350345},{961000001,589506006},{962000001,813906379},
{963000001,967631582},{964000001,475211723},{965000001,480153875},{966000001,
254836137},{967000001,565933712},{968000001,542714838},{969000001,904886393},{
970000001,535689840},{971000001,51163419},{972000001,427850268},{973000001,
678585692},{974000001,983904517},{975000001,352343728},{976000001,149324805},{
977000001,413997407},{978000001,88504521},{979000001,351395971},
{980000001,708247540},{981000001,167204412},{982000001,746735019},{983000001,
574862159},{984000001,695860800},{985000001,159838457},{986000001,176034272},{
987000001,157344477},{988000001,473517270},{989000001,981289215},{990000001,
284829514},{991000001,542617106},{992000001,928134215},{993000001,699146095},{
994000001,175893198},{995000001,606926637},{996000001,722208925},{997000001,
475735884},{998000001,323879283},{999000001,0},};

```

```

ll n;
cin >> n;

n++;

cout << (n / 998244353) + (n / 998244353) / 998244353 << '\n';

int ans = 1;

while (n > 1) {
    if ((n / 998244353) % 2 == 1) {
        ans = (ll)ans * 998244352 % 998244353;
    }

    int eee = n % 998244353;
    if (eee > 1) {
        auto it = lower_bound(arr.begin(), arr.end(), make_pair(eee, -1LL));
        it--;
        int temp_fact = it->second;
        int temp = it->first + 1;
        while (true) {
            temp_fact = (ll)temp_fact * temp % 998244353;
            if (temp == eee) {
                break;
            }
        }
    }
}

```

```

        }
        temp++;
    }

    ans = (ll)ans * temp_fact % 998244353;
}

n /= 998244353;
}
cout << ans << '\n';

// auto it = lower_bound(arr.begin(), arr.end(), make_pair(n, -1));
// it--;
// int temp_fact= it->second;
// int temp = it->first + 1;
// while (true) {
//     temp_fact = (ll)temp_fact * temp % 998244353;
//     if (temp == n) {
//         break;
//     }
//     temp++;
// }
// cout << 0 << ' ' << temp_fact << '\n';
return 0;
}

```

## Task E ()

```
#ifdef DEBUG
    #define _GLIBCXX_DEBUG
#endif
#pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
// #pragma GCC target("avx2")
// #include <bits/stdc++.h>
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <cmath>
#include <set>
#include <map>
#include <iomanip>
#include <climits>
#include <stack>
#include <numeric>
#include <queue>
#include <unordered_set>
#include <unordered_map>
#include <random>
#include <cstring>
#include <chrono>
// #include <ext/pb_ds/tree_policy.hpp>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;
// using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
// typedef tree<pair<ll, ll>, null_type, less<pair<ll, ll>>, rb_tree_tag,
//     tree_order_statistics_node_update> ordered_set;

// #define int ll
#define intn int

#ifdef DEBUG
mt19937 rnd('r' + 'o' + 'b' + 'i' + 'v' + 'i' + 'r' + 't');
#else
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
#endif
const int BUBEN = 400;
const int BUBEN3D = 2100;
const int MOD = (int)1e9 + 7;
const int BASE1 = 179;
const int MOD1 = (int)998244353;
const int BASE2 = 239;
const int MOD2 = (int)1e9 + 9;

#ifdef _getchar_nolock
char _getchar_nolock() {
    return getchar_unlocked();
}
#endif

inline ll gcd(ll a, ll b) {
    while (b) {
        a %= b;
        swap(a, b);
    }
    return a;
}

inline int bin_pow(int a, int n, int mod) {
    if (n == 0) {
        return 1;
    }
    if (n % 2 == 0) {
        int b = bin_pow(a, n / 2, mod);
        return (ll)b * b % mod;
    }
}
```



```

    }
    else {
        return (ll)a * bin_pow(a, n - 1, mod) % mod;
    }
}

// struct Q {
//     int l, r, d, idx;
// };

// bool comp(const Q &a, const Q &b) {
//     if (a.l / 548 != b.l / 548) {
//         return a.r < b.r;
//     } else {
//         return a.l / 548 < b.l / 548;
//     }
// }

// int cnt[(int)1e6];
// int aaa[(int)1e3];

// void add(int val) {
//     cnt[val]++;
//     aaa[val / 1000]++;
// }

// void dell(int val) {
//     cnt[val]--;
//     aaa[val / 1000]--;
// }

// ll get_ans(int d) {
//     ll res = 0;

//     return res;
// }

signed main() {
#ifdef DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int n;
    cin >> n;
    vector<int> arr(n);
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }
    int q;
    cin >> q;
    while (q--) {
        int a, b, d;
        cin >> a >> b >> d;
        a++;
        a--, b--;
        ll ans = 0;
        for (int i = a; i <= b; i++) {
            ans += (d + arr[i] - 1) / arr[i];
        }
        cout << ans << '\n';
    }
    // int q;
    // cin >> q;
    // vector<Q> qs(q);
    // for (int i = 0; i < q; i++) {
    //     int l, r, d;
    //     cin >> l >> r >> d;
    //     l++;
    //     l--, r--;
    //     qs[i] = {l, r, d, i};
    // }
}

```

```

// sort(qs.begin(), qs.end(), comp);
// vector<ll> ans(q);
// int temp_l = 0, temp_r = -1;
// for (auto &q: qs) {
//     while (temp_l < q.l) {
//         del(arr[temp_l]);
//         temp_l++;
//     }
//     while (temp_l > q.l) {
//         temp_l--;
//         add(arr[temp_l]);
//     }
//     while (temp_r < q.r) {
//         temp_r++;
//         add(arr[temp_r]);
//     }
//     while (temp_r > q.r) {
//         del(arr[temp_r]);
//         temp_r--;
//     }
//     ans[q.idx] = get_ans(q.d);
// }
// for (int i = 0; i < q; i++) {
//     cout << ans[i] << '\n';
// }
return 0;
}

```

## Task F ()