

Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	80	100	86	566

Task A ()

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

// #pragma GCC optimize("O3")

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

// #define int long long

using namespace std;

void solve() {
    int n;
    cin >> n;
    multiset<pair<ll, ll>> st;
    for (int i = 0; i < n; i++) {
        ll a, b;
        cin >> a >> b;
        while (a % 10 == 0) {
            a /= 10;
            b++;
        }
        st.insert({ b, a });
    }

    while (Size(st) > 1) {
        auto [b, a] = *st.begin();
        st.erase(st.begin());
        auto [b2, a2] = *st.begin();
        if (b != b2) {
            cout << b << "\n";
            return;
        }
        ll x = a + a2;
        st.erase(st.begin());
        while (x % 10 == 0) {
            x /= 10;
            b++;
        }
        st.insert({ b, x });
    }
    cout << st.begin()->first << "\n";
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef DIMSS
    freopen("test.txt", "r", stdin);
#endif
}
```

```

int T = 1;
// cin >> T;
while (T--) {
    solve();
    #ifdef DIMSS
        cerr << "—————\n";
    #endif
}

return 0;
}

```

Task B ()

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

// #pragma GCC optimize("O3")

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

// #define int long long

using namespace std;

void solve() {
    int n;
    cin >> n;
    vector<int> k(n);
    for (int &i : k)
        cin >> i;

    for (int i = n - 1; i >= 0; i--) {
        while (k[i]) {
            cout << "Flip_and_wait" << endl;
            int s = 0;
            while (true) {
                string t;
                cin >> t;
                s += count(all(t), 'e') / 2;
                if (s >= i + 1)
                    break;
            }
            cout << "Wait" << endl;
            k[i]--;
        }
    }
    cout << "Stop" << endl;
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef DIMSS
    freopen("test.txt", "r", stdin);
#endif

    int T = 1;
    // cin >> T;
    while (T--) {
        solve();
#ifdef DIMSS
        cerr << "—————\n";
#endif
    }

    return 0;
}
```

Task C ()

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

// #pragma GCC optimize("O3")

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

// #define int long long

using namespace std;

void solve() {
    string s;
    cin >> s;
    int n = Size(s);
    bool f = count(all(s), '?') == 0;

    if (n == 2) {
        if (f) {
            if (s == "00")
                cout << "0?\n";
            else if (s == "01")
                cout << "?1\n";
            else if (s == "10")
                cout << "?0\n";
            else
                cout << "1?\n";
        } else {
            if (s == "0?")
                cout << "00\n";
            else if (s == "?1")
                cout << "01\n";
            else if (s == "?0")
                cout << "10\n";
            else
                cout << "11\n";
        }
        return;
    }

    if (f) {
        int c0 = count(all(s), '0');
        int c1 = Size(s) - c0;
        if (c0 == n) {
            cout << "0" << string(n - 1, '?') << "\n";
            return;
        }
        if (c1 == n) {
            cout << "1" << string(n - 1, '?') << "\n";
            return;
        }
        if (s == "0" + string(n - 1, '1')) {
            cout << "01" << string(n - 2, '?') << "\n";
            return;
        }
        if (s == "1" + string(n - 1, '0')) {
            cout << "10" << string(n - 2, '?') << "\n";
            return;
        }
        if (c0 >= c1) {
            for (char &c : s) {
                if (c == '0')
                    c = '?';
            }
        } else {
            for (char &c : s) {
                if (c == '1')
                    c = '?';
            }
        }
    }
}
```

```

    }
    cout << s << "\n";
} else {
    int c0 = count(all(s), '0');
    if (s == "0" + string(n - 1, '?')) {
        cout << string(n, '0') << "\n";
        return;
    }
    if (s == "1" + string(n - 1, '?')) {
        cout << string(n, '1') << "\n";
        return;
    }
    if (s == "01" + string(n - 2, '?')) {
        cout << "0" << string(n - 1, '1') << "\n";
        return;
    }
    if (s == "10" + string(n - 2, '?')) {
        cout << "1" << string(n - 1, '0') << "\n";
        return;
    }
    for (char &c : s) {
        if (c == '?') {
            if (c0)
                c = '1';
            else
                c = '0';
        }
    }
    cout << s << "\n";
}
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef DIMSS
    freopen("test.txt", "r", stdin);
#endif

    int T = 1;
    cin >> T;
    while (T--) {
        solve();
#ifdef DIMSS
        cerr << "—————\n";
#endif
    }

    return 0;
}

```

Task D ()

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

#pragma GCC optimize("O3")

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

// #define int long long

using namespace std;

const int mod = 998244353, K = 1e6;

ll f[] = { 373341033, 767255872, 947725884, 329922062, 16967458, 292591034, 938514033, 132474549,
  640286220, 348047904, 751773299, 905627554, 729573419, 491995101, 770444540, 424726128,
  118317067, 686230795, 655364145, 172389002, 376310761, 497067988, 405410848, 472438245,
  931751289, 847714097, 304002203, 57148610, 957894057, 851717269, 700889416, 99408833,
  988969996, 850850566, 623794005, 586462953, 733256239, 340632618, 728316157, 436744235,
  135992589, 648868331, 133929958, 481043409, 511406717, 849611468, 339882940, 625140785,
  900540109, 827606096, 980181949, 611210732, 624805272, 890749332, 261029233, 540977339,
  861788743, 729362324, 852686018, 339015112, 797170989, 580041677, 437897046, 194642304,
  344185236, 911175248, 797464272, 229855984, 512989339, 256838142, 758029551, 707876604,
  203831852, 92973362, 130488323, 348601163, 415544049, 299327342, 318085132, 517108871,
  959196941, 714442105, 450268958, 754352453, 744325658, 179901180, 633769143, 351295146,
  305780829, 157331959, 254407647, 382955618, 953570454, 581584188, 335957257, 736683476,
  645078967, 778211210, 162956143, 555766213, 776242605, 469648636, 239612217, 155812124,
  502621547, 869570823, 308037365, 626960023, 273073416, 226124125, 411489459, 192821387,
  489004391, 740310551, 428003712, 595836568, 489943830, 607763880, 886320768, 125643157,
  463595448, 966814595, 871093285, 48463471, 211524392, 945491059, 649949957, 440471554,
  474710741, 275074449, 798477637, 853517300, 105325786, 768987050, 983231361, 33801217,
  154785248, 912779215, 499635408, 955114466, 831696133, 83912831, 820055219, 950373190,
  328889658, 358885039, 985085368, 447209443, 648812565, 666124859, 242278752, 800524966,
  373385314, 152303406, 471546683, 332508835, 401924457, 26634445, 781138633, 685242536,
  416290353, 432352428, 636447732, 911715871, 132864142, 166981044, 491773227, 502280542,
  720445773, 335411749, 481470278, 203202733, 343291435, 136599194, 682458521, 210520008,
  554279522, 58426201, 513140238, 468593662, 700540142, 342607733, 327146165, 53548758,
  370748654, 576602005, 193745467, 40340413, 657291304, 415484644, 179632705, 4416248,
  549587008, 20433517, 534185692, 159129816, 783873254, 465797767, 672712527, 278928624,
  6636566, 9593235, 559744074, 325213012, 104364007, 854899677, 493030650, 557450603, 612210078,
  734342407, 350678410, 423791007, 442432194, 121957373, 552150666, 541936703, 108944756,
  571038718, 777859496, 285611481, 221994737, 206694051, 662691847, 633545077, 416974187,
  485726677, 41450237, 889531131, 839679515, 420087766, 704413763, 601256670, 936957013,
  341299373, 682108952, 735919227, 189576289, 827958888, 354859358, 931636010, 460233997,
  753349235, 692054908, 462613599, 228331908, 305817565, 24270828, 634434602, 581713744,
  17401466, 314322310, 892111747, 104654500, 354624942, 800931312, 559118300, 302664725,
  312385419, 638728884, 90550884, 362887840, 451392696, 273533726, 143697328, 777640951,
  452747367, 588559929, 841748744, 898436910, 989932409, 312725456, 168494110, 314459019,
  209720400, 510842020, 6286770, 404670713, 755943322, 716009046, 311768297, 891296628,
  408567580, 566215917, 216468857, 358248716, 32582861, 790204919, 505047907, 906974836,
  854882789, 866045800, 196118119, 211026510, 737825844, 283398129, 641471369, 293666652,
  194290394, 179326413, 557570941, 386223068, 523356620, 114156819, 250266918, 494564333,
  572379979, 130817429, 52837843, 695505934, 57616295, 330239660, 789441735, 333356816,
  791092828, 455228889, 242533203, 76442909, 390106747, 21296556, 428534654, 440957765,
  581006326, 787855833, 904111066, 875218657, 233178061, 496142383, 274030145, 121282738,
  210722039, 630535031, 33308984, 493752236, 428368063, 860692052, 766596421, 158570663,
  189641096, 476902404, 410696428, 879860700, 973102837, 754973363, 747220321, 489565365,
  934363433, 916873509, 795102027, 525763375, 529390863, 471705752, 272764068, 3886815,
  327943865, 516467014, 790412430, 957419075, 590135780, 482684731, 853131408, 34200696,
  740621661, 64517026, 864277793, 844569352, 790469403, 711710304, 23879276, 649278407,
  73025643, 413904858, 255199777, 185096811, 502624522, 222789952, 540911386, 318004337,
  268818154, 524546624, 696956211, 217788718, 38653629, 577464409, 397928930, 233990026,
  346832368, 431984313, 677716696, 749481503, 306265621, 156499134, 516590753, 618244690,
  302703378, 399317266, 564477634, 454265889, 499063374, 381513987, 498459484, 310228476,
  5169795, 693347150, 671122583, 744748792, 623713492, 815777350, 766895791, 334244669,
  104739591, 530339813, 385109693, 942928225, 250411025, 982721496, 168778805, 288198243,
  563239427, 985803340, 376309192, 237601933, 290299916, 478372908, 463184500, 367997369,
  91777834, 215357944, 536521577, 3732871, 856771470, 141036473, 996346576, 74699370, 965115675,
  683399215, 824211806, 33915627, 802177373, 5239340, 754167047, 752580747, 614363002,
  490041002, 552384209, 219609003, 562201897, 42401981, 534441901, 201926128, 871913932,
```

443094478, 824245284, 790700892, 71150580, 416328510, 204814028, 276436638, 868462891, 709193053, 784687523, 10953187, 83783761, 457484681, 454283303, 629367908, 128300229, 790017763, 54208167, 825196119, 514742887, 149568991, 71717560, 519149809, 954747891, 950205268, 694151797, 785832808, 333043782, 694458798, 206041139, 358624954, 366024744, 801412693, 137770145, 517957034, 826648641, 171796032, 952492760, 827675308, 75947674, 515111949, 776623590, 883257892, 347632983, 669058178, 524635929, 855375784, 623913205, 470044801, 961979463, 616298543, 411722193, 9994611, 441828743, 500414052, 139089688, 858694528, 987163955, 208670281, 188666040, 899190753, 845418269, 591225136, 367568516, 935345786, 982651461, 652473482, 285737027, 996568946, 739716628, 669382585, 73582421, 768596250, 948661293, 364715376, 558672364, 159184120, 137112513, 44541398, 741106808, 751510860, 548332255, 820020996, 557197209, 520163378, 839268934, 29948534, 864008311, 760198325, 502802642, 321416664, 160612865, 745308310, 615555857, 540644879, 797169624, 572697382, 479223829, 892893205, 703457977, 631279296, 435982397, 426775230, 442504959, 348114168, 636785236, 141735590, 963972479, 96723640, 409224913, 471832467, 771419065, 736549368, 515978420, 832079429, 706947933, 38187003, 267159212, 415839634, 483378602, 996510859, 928602201, 821207666, 813871624, 413428501, 2733587, 312974446, 312524796, 194228741, 634157572, 354586055, 80380273, 637551134, 751371385, 782957354, 863245139, 337828475, 678085055, 431550160, 599957641, 290176899, 697893162, 30117403, 737373596, 186647307, 339519763, 610749505, 854636901, 833423973, 540543914, 235062461, 806266339, 126635647, 106532215, 506802112, 615496212, 480883789, 781674922, 505992320, 339477810, 265095941, 801385714, 813133514, 407027544, 137151576, 274636267, 413260948, 491931660, 716072028, 866059907, 468981957, 207689500, 7525949, 331510983, 757938508, 390684489, 373048285, 619855889, 213730930, 92174656, 319310307, 634796380, 735668895, 51164646, 108449638, 934463103, 858549263, 560932194, 526467914, 618834632, 453698347, 963840843, 34714579, 733853567, 103879985, 928437200, 811680707, 268115373, 445070664, 491462894, 59887518, 32133018, 709551213, 684179626, 416914517, 460623712, 992641496, 42340667, 546450555, 840443107, 367681078, 550042793, 585893830, 818571246, 59136287, 5898995, 8255250, 500967683, 150387888, 177351190, 290845449, 206910882, 2992668, 733218343, 569615130, 607671852, 115839553, 690393244, 279091275, 868425779, 565631233, 944588430, 154375554, 402451753, 569351866, 100261853, 648532808, 226648783, 746300667, 163058892, 855172102, 374477045, 590354530, 276338307, 400170500, 234742043, 403774405, 329319870, 894309177, 342887910, 190659621, 383197702, 762652921, 998220317, 883191986, 315095516, 198691341, 174642210, 343331502, 593481257, 419980564, 951089852, 257482828, 563500050, 177282729, 804694681, 294796289, 766301779, 529805617, 254896378, 467716966, 279384432, 124276278, 829582520, 97879626, 727642180, 523641307, 343182919, 307691414, 53058025, 215788948, 398984944, 780895451, 778897690, 756707100, 531042278, 339298934, 788146469, 670090327, 588301316, 781164593, 614644869, 420369170, 436985881, 573088036, 699606231, 175644200, 741710823, 601267788, 184263986, 726664570, 461031790, 620930389, 74226837, 621110284, 970147971, 172126271, 410385152, 755675430, 347475033, 379270099, 128373376, 816288285, 94121271, 59264751, 728817482, 697776307, 723597173, 470601620, 428115758, 196669740, 186999325, 685982162, 163183041, 719359789, 789169821, 346468287, 178396029, 640918446, 38341199, 68319403, 723887503, 533766493, 731870340, 342615262, 757019435, 208013417, 108371925, 49792347, 435546086, 425359240, 300056848, 530184200, 645812126, 955911554, 836547746, 320751239, 892158627, 372030436, 729351726, 43199301, 934219201, 589412031, 211397893, 590516224, 988326280, 572343597, 539659655, 101130337, 979797196, 810271404, 577226828, 190730601, 332520128, 256935908, 399859924, 593687398, 42458458, 471639539, 496866933, 926803457, 394803984, 906541359, 248678995, 34652109, 838952767, 580770625, 859338481, 628889998, 118590901, 486158958, 723061060, 743579807, 623539021, 560407243, 49384248, 459388370, 568786369, 520818860, 490882623, 975971986, 280993863, 808715789, 365347243, 665452, 110113905, 961416093, 724245976, 113640613, 404349213, 889611662, 497443822, 909739474, 863133397, 320404259, 938647632, 989987404, 421491361, 792086828, 255452201, 204440291, 394571180, 412772547, 372312492, 849637457, 379166311, 337939051, 584988895, 628880910, 996332747, 561819735, 781703936, 543068001, 768032076, 765487420, 447719611, 751859365, 94078029, 158044576, 184479642, 889266083, 703595125, 245252962, 967433228, 170471967, 281281138, 589402994, 485359896, 397281690, 719849426, 956025309, 702197990, 467738801, 584358190, 367623620, 877366912, 806438333, 712967089, 8268475, 726774649, 841392206, 224601693, 391142247, 503056152, 185037551, 803323547, 918718219, 301940936, 75430876, 424209449, 193491106, 468269867, 483297557, 577857097, 885584129, 842682022, 550540549, 488146980, 494352930, 102914310, 267449220, 599034542, 665895643, 31461852, 644732511, 108068955, 455230382, 272514261, 27885419, 165620450, 476761184, 898976537, 431387531, 142343172, 152528554, 741071934, 694676509, 591625850, 809637823, 713567498, 983182100, 766170885, 725116391, 816643259, 392326502, 769192183, 873837348, 205444350, 811153598, 387431166, 612774743, 120752693, 263352635, 5596649, 142336171, 730896434, 420887589, 962122793, 653755692, 97055971, 284781463, 738100649, 415743514, 914977622, 415856296, 909447348, 661315476, 414280049, 982813996, 821141954, 799942554, 135198887, 691764619, 783593943, 129766265, 381567813, 80949058, 105944998, 94733374, 704254235, 800980711, 168029194, 962904147, 905302889, 433980144, 629258717, 824083087, 715051720, 799603467, 397898827, 91054189, 678108321, 299523849, 696308918, 618933955, 227165093, 305948706, 69563578, 0, 893359532 };

```
void solve() {
    ll n;
    cin >> n;
    n++;
```

```

    if (n >= mod) {
        ll res = 0;
        ll x = mod;
        while (true) {
            ll d = n / x;
            res += d;
            if (d == 0)
                break;
            x *= mod;
        }
        cout << res % mod << "\n" << "0\n";
        return;
    }
    ll res = 1;
    int i = 1;
    int j = 0;
    while (i + K <= n) {
        res *= f[j];
        res %= mod;
        j++;
        i += K;
    }
    while (i <= n) {
        res *= i;
        res %= mod;
        i++;
    }
    cout << "0\n" << res << "\n";
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef DIMSS
    freopen("test.txt", "r", stdin);
#endif

    int T = 1;
    // cin >> T;
    while (T--) {
        solve();
#ifdef DIMSS
        cerr << "————\n";
#endif
    }

    return 0;
}

```


Task E ()

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

#pragma GCC optimize("O3")

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

// #define int long long

using namespace std;

const int N = 3e6 + 100, K = 2500, MAX_C = 2e6 + 100;

int n, C;
int a[N];

struct Q {
    int d, num;
    bool rev;
};

vector<Q> que[N];
ll ans[N], f[N * 2];

void update(int i, int x) {
    i += C;
    f[i] += x;
    i >>= 1;
    while (i >= 1) {
        f[i] += x;
        i >>= 1;
    }
}

ll getSum(int i) {
    int l = 0, r = i;
    l += C;
    r += C;
    ll res = 0;
    while (l < r) {
        if (l & 1) {
            res += f[l];
            l++;
        }
        if (r & 1) {
            r ^= 1;
            res += f[r];
        }
        l >>= 1;
        r >>= 1;
    }
    return res;
}

int cnt[MAX_C + 1];

void solve() {
    cin >> n;
    vector<int> sm;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
        if (a[i] <= K)
            sm.push_back(a[i]);
    }
    sort(all(sm));
    sm.resize(unique(all(sm)) - sm.begin());

    int q;
    cin >> q;
```

```

C = 0;
for (int t = 0; t < q; t++) {
    int l, r, d;
    cin >> l >> r >> d;
    que[r - 1].push_back({ d, t, false });
    if (l)
        que[l - 1].push_back({ d, t, true });
    C = max(C, d);
}
C++;

for (int r = 0; r < n; r++) {
    if (a[r] > K) {
        for (int x = 0; x < C; x += a[r])
            update(x, 1);
    } else {
        cnt[a[r]]++;
    }

    for (Q qu : que[r]) {
        int d = qu.d;
        int num = qu.num;
        bool rev = qu.rev;
        ll res = getSum(d);
        for (int x : sm)
            res += (d + x - 1) / x * 1ll * cnt[x];
        if (!rev)
            ans[num] += res;
        else
            ans[num] -= res;
    }
}

for (int i = 0; i < q; i++)
    cout << ans[i] << "\n";
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef DIMSS
    freopen("test.txt", "r", stdin);
#endif

    int T = 1;
    // cin >> T;
    while (T--) {
        solve();
#ifdef DIMSS
        cerr << "—————\n";
#endif
    }

    return 0;
}

```

Task F ()

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

#pragma GCC optimize("O3")

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

// #define int long long

using namespace std;

const int N = 1e6;

int n;
int l[N], r[N];
int dp[N + 1];

void solve() {
    cin >> n;
    for (int i = 0; i < n; i++)
        cin >> l[i] >> r[i];

    dp[0] = 1;
    for (int i = 1; i <= n; i++) {
        // dp[i] = max(1, dp[i - 1]);
        dp[i] = 1;
        ll mn = 0, mx = 0;
        int cnt = 0;
        for (int j = i - 1; j >= 0 && cnt < 30; j--) {
            mn += l[j];
            mx += r[j];
            if (mn <= 0 && 0 <= mx) {
                dp[i] = max(dp[i], dp[j] + 1);
                cnt++;
            }
        }
    }
    cout << *max_element(dp, dp + n + 1) << "\n";
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef DIMSS
    freopen("test.txt", "r", stdin);
#endif

    int T = 1;
    // cin >> T;
    while (T--) {
        solve();
#ifdef DIMSS
        cerr << "—————\n";
#endif
    }

    return 0;
}
```