

Олимпиада СПбГУ по информатике 2021/22 учебного года

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

Task A ()

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

using namespace std;

#define int long long

void solve() {
    int n;
    cin >> n;
    int a, b;
    map<int, int> mp;
    for (int i = 0; i < n; ++i) {
        cin >> a >> b;
        mp[b] += a;
    }
    while(true) {
        pair<int, int> fst = *mp.begin();
        mp.erase(fst.first);
        if (fst.second % 10 != 0) {
            cout << fst.first << endl;
            return;
        }
        else {
            mp[fst.first + 1] += fst.second / 10;
        }
    }
}

int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int tests = 1;
    // cin >> tests;
    while(tests--) {
        solve();
    }
}
```

Task B ()

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

using namespace std;

void solve() {
    int n;
    cin >> n;
    vector<int> sp(n);
    int sm = 0;
    for (int i = 0; i < n; ++i) {
        cin >> sp[i];
        sm += sp[i];
    }
    cout << "Flip_and_wait" << endl;
    for (int i = n - 1; i >= 0; --i) {
        for (int cnt = 0; cnt < sp[i]; ++cnt) {
            int tmp = 0;
            while(true) {
                string str;
                cin >> str;
                tmp += ((int)str.size() - 3) / 2;
                if (tmp > i) {
                    break;
                }
                cout << "Wait" << endl;
            }
            --sm;
            if (sm == 0) {
                cout << "Stop" << endl;
                return;
            }
            else {
                cout << "Flip_and_wait" << endl;
            }
        }
    }
}

int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int tests = 1;
    // cin >> tests;
    while(tests--) {
        solve();
    }
}
```

Task C ()

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

using namespace std;

void solve() {
    int n;
    cin >> n;
    for (int i = 0; i < n; ++i) {
        string str;
        cin >> str;
        int cnt_0 = 0;
        int cnt_1 = 0;
        bool q = false;
        for (auto c : str) {
            if (c == '?') {
                q = true;
            }
            else if (c == '0') {
                cnt_0 += 1;
            }
            else {
                cnt_1 += 1;
            }
        }
        if (q) {
            bool fl = false;
            int s;
            for (int i = 0; i < (int)str.size(); ++i) {
                if (!fl) {
                    if (str[i] == '?') {
                        continue;
                    }
                    else {
                        s = str[i] - '0';
                        for (int j = 0; j < i; ++j) {
                            cout << (s ^ 1);
                        }
                        cout << s;
                    }
                    fl = true;
                }
                else {
                    if (str[i] == '?') {
                        cout << s;
                    }
                    else {
                        cout << str[i];
                    }
                }
            }
            cout << "\n";
        }
        else {
            int s;
            if (cnt_0 > cnt_1) {
                s = 0;
            }
            else if (cnt_0 == cnt_1) {
                s = (str[0] - '0') ^ 1;
            }
            else {
                s = 1;
            }
            bool fl = false;
            for (int i = 0; i < (int)str.size(); ++i) {
                if (!fl) {
                    if (str[i] - '0' != s) {
                        cout << "?";
                    }
                    else {
                        //FOUND!
                        fl = true;
                    }
                }
            }
        }
    }
}
```

```

        cout << s;
    }
    }
    else {
        if (str[i] - '0' == s) {
            cout << '?';
        }
        else {
            cout << char('0' + (s ^ 1));
        }
    }
}
cout << "\n";
}
}

int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int tests = 1;
    // cin >> tests;
    while(tests--) {
        solve();
    }
}

```

Task D ()

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

using namespace std;

#define int long long

const int MOD = 998244353;

vector<int> prec = {1, 373341033, 45596018, 834980587, 623627864, 428937595, 442819817, 499710224,
833655840, 83857087, 295201906, 788488293, 671639287, 849315549, 597398273, 813259672,
732727656, 244038325, 122642896, 310517972, 160030060, 483239722, 683879839, 712910418,
384710263, 433880730, 844360005, 513089677, 101492974, 959253371, 957629942, 678615452,
34035221, 56734233, 524027922, 31729117, 102311167, 330331487, 8332991, 832392662, 545208507,
594075875, 318497156, 859275605, 300738984, 767818091, 864118508, 878131539, 316588744,
812496962, 213689172, 584871249, 980836133, 54096741, 417876813, 363266670, 335481797,
730839588, 393495668, 435793297, 760025067, 811438469, 720976283, 650770098, 586537547,
117371703, 566486504, 749562308, 708205284, 932912293, 939830261, 983699513, 206579820,
301188781, 593164676, 770845925, 247687458, 41047791, 266419267, 937835947, 506268060,
6177705, 936268003, 166873118, 443834893, 328979964, 470135404, 954410105, 117565665,
832761782, 39806322, 478922755, 394880724, 821825588, 468705875, 512554988, 232240472,
876497899, 356048018, 895187265, 808258749, 575505950, 68190615, 939065335, 552199946,
694814243, 385460530, 529769387, 640377761, 916128300, 440133909, 362216114, 826373774,
502324157, 457648395, 385510728, 904737188, 78988746, 454565719, 623828097, 686156489,
713476044, 63602402, 570334625, 681055904, 222059821, 477211096, 343363294, 833792655,
461853093, 741797144, 74731896, 930484262, 268372735, 941222802, 677432735, 474842829,
700451655, 400176109, 697644778, 390377694, 790010794, 360642718, 505712943, 946647976,
339045014, 715797300, 251680896, 70091750, 40517433, 12629586, 850635539, 110877109,
571935891, 695965747, 634938288, 69072133, 155093216, 749696762, 963086402, 544711799,
724471925, 334646013, 574791029, 722417626, 377929821, 743946412, 988034679, 405207112,
18063742, 104121967, 638607426, 607304611, 751377777, 35834555, 313632531, 18058363,
656121134, 40763559, 562910912, 495867250, 48767038, 210864657, 659137294, 715390025,
865854329, 324322857, 388911184, 286059202, 636456178, 421290700, 832276048, 726437551,
526417714, 252522639, 386147469, 674313019, 274769381, 226519400, 272047186, 117153405,
712896591, 486826649, 119444874, 338909703, 18536028, 41814114, 245606459, 140617938,
250512392, 57084755, 157807456, 261113192, 40258068, 194807105, 325341339, 884328111,
896332013, 880836012, 737358206, 202713771, 785454372, 399586250, 485457499, 640827004,
546969497, 749602473, 159788463, 159111724, 218592929, 675932866, 314795475, 811539323,
246883213, 696818315, 759880589, 4302336, 353070689, 477909706, 559289160, 79781699,
878094972, 840903973, 367416824, 973366814, 848259019, 462421750, 667227759, 897917455,
81800722, 956276337, 942686845, 420541799, 417005912, 272641764, 941778993, 217214373,
192220616, 267901132, 50530621, 652678397, 354880856, 164289049, 781023184, 105376215,
315094878, 607856504, 733905911, 457743498, 992735713, 35212756, 231822660, 276036750,
734558079, 424180850, 433186147, 308380947, 18333316, 12935086, 351491725, 655645460,
535812389, 521902115, 67016984, 48682076, 64748124, 489360447, 361275315, 786336279,
805161272, 468129309, 645091350, 887284732, 913004502, 358814684, 281295633, 328970139,
395955130, 164840186, 820902807, 761699708, 246274415, 592331769, 913846362, 866682684,
600130702, 903837674, 529462989, 90612675, 526540127, 533047427, 110008879, 674279751,
801920753, 645226926, 676886948, 752481486, 474034007, 457790341, 166813684, 287671032,
188118664, 244731384, 404032157, 269766986, 423996017, 182948540, 356801634, 737863144,
652014069, 206068022, 504569410, 919894484, 593398649, 963768176, 882517476, 702523597,
949028249, 128957299, 171997372, 50865043, 20937461, 690959202, 581356488, 369182214,
993580422, 193500140, 540665426, 365786018, 743731625, 144980423, 979536721, 773259009,
617053935, 247670131, 843705280, 30419459, 985463402, 261585206, 237885042, 111276893,
488166208, 137660292, 720784236, 244467770, 26368504, 792857103, 666885724, 670313309,
905683034, 259415897, 512017253, 826265493, 111960112, 633652060, 918048438, 516432938,
386972415, 996212724, 610073831, 444094191, 72480267, 665038087, 11584804, 301029012,
723617861, 113763819, 778259899, 937766095, 535448641, 593907889, 783573565, 673298635,
599533244, 655712590, 173350007, 868198597, 169013813, 585161712, 697502214, 573994984,
285943986, 675831407, 3134056, 965907646, 401920943, 665949756, 236277883, 612745912,
813282113, 892454686, 901222267, 624900982, 927122298, 686321335, 84924870, 927606072,
506664166, 353631992, 165913238, 566073550, 816674343, 864877926, 171259407, 908752311,
874007723, 803597299, 613676466, 880336545, 282280109, 128761001, 58852065, 474075900,
434816091, 364856903, 149123648, 388854780, 314693916, 423183826, 419733481, 888483202,
238933227, 336564048, 757103493, 100189123, 855479832, 51370348, 403061033, 496971759,
831753030, 251718753, 272779384, 683379259, 488844621, 881783783, 659478190, 445719559,
740782647, 546525906, 985524427, 548033568, 333772553, 331916427, 752533273, 730387628,
93829695, 655989476, 930661318, 334885743, 466041862, 428105027, 888238707, 232218076,
769865249, 730641039, 616996159, 231721356, 326973501, 426068899, 722403656, 742756734,
663270261, 364187931, 350431704, 671823672, 633125919, 226166717, 386814657, 237594135,
451479365, 546182474, 119366536, 465211069, 605313606, 728508871, 249619035, 663053607,
900453742, 48293872, 229958401, 62402409, 69570431, 71921532, 960467929, 537087913, 514588945,
513856225, 415497414, 286592050, 645469437, 102052166, 163298189, 873938719, 617583886,
```

986843080, 962390239, 580971332, 665147020, 88900164, 89866970, 826426395, 616059995,
443012312, 659160562, 229855967, 687413213, 59809521, 398599610, 325666688, 154765991,
159186619, 210830877, 386454418, 84493735, 974220646, 820097297, 2191828, 481459931,
729073424, 551556379, 926316039, 151357011, 808637654, 218058015, 786112034, 850407126,
84202800, 94214098, 30019651, 121701603, 176055335, 865461951, 553631971, 286620803,
984061713, 888573766, 302767023, 977070668, 110954576, 83922475, 51568171, 60949367, 19533020,
510592752, 615419476, 341370469, 912573425, 286207526, 206707897, 384156962, 414163604,
193301813, 749570167, 366933789, 11470970, 600191572, 391667731, 328736286, 30645366,
215162519, 604947226, 236199953, 718439098, 411423177, 803407599, 632441623, 766760224,
263006576, 757681534, 61082578, 681666415, 947466395, 12206799, 659767098, 933746852,
978860867, 59215985, 161179205, 439197472, 259779111, 511621808, 145770512, 882749888,
943124465, 872053396, 631078482, 166861622, 743415395, 772287179, 602427948, 924112080,
385643091, 794973480, 883782693, 869723371, 805963889, 313106351, 262132854, 400034567,
488248149, 265769800, 791715397, 408753255, 468381897, 415812467, 172922144, 64404368,
281500398, 512318142, 288791777, 955559118, 242484726, 536413695, 205340854, 707803527,
576699812, 218525078, 875554190, 46283078, 833841915, 763148293, 807722138, 788080170,
556901372, 150896699, 253151120, 97856807, 918256774, 771557187, 582547026, 472709375,
911615063, 743371401, 641382840, 446540967, 184639537, 157247760, 775930891, 939702814,
499082462, 19536133, 548753627, 593243221, 563850263, 185475971, 687419227, 396799323,
657976136, 864535682, 433009242, 860830935, 33107339, 517661450, 467651311, 812398757,
202133852, 431839017, 709549400, 99643620, 773282878, 290471030, 61134552, 129206504,
929147251, 837008968, 422332597, 353775281, 469563025, 62265336, 835064501, 851685235,
21197005, 264793769, 326416680, 118842991, 84257200, 763248924, 687559609, 150907932,
401832452, 242726978, 766752066, 959173604, 390269102, 992293822, 744816299, 476631694,
177284763, 702429415, 374065901, 169855231, 629007616, 719169602, 564737074, 475119050,
714502830, 40993711, 820235888, 749063595, 239329111, 612759169, 18591377, 419142436,
442202439, 941600951, 158013406, 637073231, 471564060, 447222237, 701248503, 599797734,
577221870, 69656699, 51052704, 6544303, 10958310, 554955500, 943192237, 192526269, 897983911,
961628039, 240232720, 627280533, 710239542, 70255649, 261743865, 228474833, 776408079,
304180483, 63607040, 953297493, 758058902, 395529997, 156010331, 825833840, 539880795,
234683685, 52626619, 751843490, 116909119, 62806842, 574857555, 353417551, 40061330,
822203768, 681051568, 490913702, 9322961, 766631257, 124794668, 37844313, 163524507,
729108319, 490867505, 47035168, 682765157, 53842115, 817965276, 757179922, 339238384,
909741023, 150530547, 158444563, 140949492, 993302799, 551621442, 137578883, 475122706,
443869843, 605400098, 689361523, 769596520, 801661499, 474900284, 586624857, 349960501,
134084537, 650564083, 877097974, 379857427, 887890124, 159436401, 133274277, 986182139,
729720334, 568925901, 459461496, 499309445, 493171177, 460958750, 380694152, 168836226,
840160881, 141116880, 225064950, 109618190, 842341383, 85305729, 759273275, 97369807,
669317759, 766247510, 829017039, 550323884, 261274540, 918239352, 29606025, 870793828,
293683814, 378510746, 367270918, 481292028, 813097823, 798448487, 230791733, 899305835,
504040630, 162510533, 479367951, 275282274, 806951470, 462774647, 56473153, 184659008,
905122161, 664034750, 109726629, 59372704, 325795100, 486860143, 843736533, 924723613,
880348000, 801252478, 616515290, 776142608, 284803450, 583439582, 274826676, 6018349,
377403437, 244041569, 527081707, 544763288, 708818585, 354033051, 904309832, 589922898,
673933870, 682858433, 945260111, 899893421, 515264973, 911685911, 9527148, 239480646,
524126897, 48259065, 578214879, 118677219, 786127243, 869205770, 923276513, 937928886,
802186160, 12198440, 638784295, 34200904, 758925811, 185027790, 80918046, 120604699,
610456697, 573601211, 208296321, 49743354, 653691911, 490750754, 674335312, 887877110,
875880304, 308360096, 414636410, 886100267, 8525751, 636257427, 558338775, 500159951,
696213291, 97268896, 364983542, 937928436, 641582714, 586211304, 345265657, 994704486,
443549763, 207259440, 302122082, 166055224, 623250998, 239642551, 476337075, 283167364,
211328914, 68064804, 950202136, 187552679, 18938709, 646784245, 598764068, 538505481,
610424991, 864445053, 390248689, 278395191, 686098470, 935957187, 868529577, 329970687,
804930040, 84992079, 474569269, 810762228, 573258936, 756464212, 155080225, 286966169,
283614605, 19283401, 24257676, 871831819, 612689791, 846988741, 617120754, 971716517,
979541482, 297910784, 991087897, 783825907, 214821357, 689498189, 405026419, 946731704,
609346370, 707669156, 457703127, 957341187, 980735523, 649367684, 791011898, 82098966,
234729712, 105002711, 130614285, 291032164, 193188049, 363211260, 58108651, 100756444,
954947696, 346032213, 863300806, 36876722, 622610957, 289232396, 667938985, 734886266,
395881057, 417188702, 183092975, 887586469, 83334648, 797819763, 100176902, 781587414,
841864935, 371674670, 18247584};

```
int mx = 998244352;
```

```
int fast_pow(int x, int y) {
    if (y == 0) {
        return 1;
    }
    if (y % 2 == 0) {
        return fast_pow((x * x) % MOD, y / 2);
    }
    else {
        return (fast_pow(x, y - 1) * x) % MOD;
    }
}
```

```

int calc(int q) {
    int now = prec[q / 1000000];
    for (int x = q / 1000000 * 1000000 + 1; x <= q; ++x) {
        now = (now * x) % MOD;
    }
    return now;
}

void solve() {
    int n;
    cin >> n;
    ++n;
    int tmp = n;
    int cnt_zero = 0;
    while(tmp != 0) {
        cnt_zero += tmp / MOD;
        tmp /= MOD;
    }
    int ans = 1;
    tmp = n;
    while(tmp != 0) {
        ans = ((ans * fast_pow(mx, tmp / MOD)) % MOD * calc(tmp % MOD)) % MOD;
        tmp /= MOD;
    }
    cout << cnt_zero << " " << ans << "\n";
}

int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    // freopen("output.txt", "w", stdout);
    int tests = 1;
    // cin >> tests;
    while(tests--) {
        solve();
    }
}

```

Task E ()

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

using namespace std;

#define int long long

const int MAXN = 3e5 + 100;
vector<int> t[MAXN * 4 + 4];
int sp[MAXN];
int n, q;

void build(int i, int l, int r) {
    if (l + 1 == r) {
        t[i] = {sp[l]};
        return;
    }
    int m = (l + r) / 2;
    build(2 * i + 1, l, m);
    build(2 * i + 2, m, r);
    merge(t[2 * i + 1].begin(), t[2 * i + 1].end(), t[2 * i + 2].begin(), t[2 * i + 2].end(),
        back_inserter(t[i]));
}

int get_ans(int i, int l, int r, int q_l, int q_r, int d) {
    // cout << i << " " << l << " " << r << " " << q_l << " " << q_r << " " << d << endl;
    if (q_l <= l && r <= q_r) {
        // calc ans
        int ans = 0;
        int ind = 0;
        while(ind != (int)t[i].size()) {
            // cout << ind << " " << t[i].size() << endl;
            int now = (d + t[i][ind] - 1) / t[i][ind];
            int lft = ind;
            int rig = t[i].size();
            while(rig - lft > 1) {
                int mid = (lft + rig) / 2;
                if ((d + t[i][mid] - 1) / t[i][mid] == now) {
                    lft = mid;
                }
                else {
                    rig = mid;
                }
            }
            ans += now * (lft - ind + 1);
            ind = lft + 1;
        }
        return ans;
    }
    if (r <= q_l || q_r <= l) {
        return 0;
    }
    int m = (l + r) / 2;
    return get_ans(2 * i + 1, l, m, q_l, q_r, d) + get_ans(2 * i + 2, m, r, q_l, q_r, d);
}

void solve() {
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> sp[i];
    }
    build(0, 0, n);
    // cout << "OK 1" << endl;
    cin >> q;
    for (int req = 0; req < q; ++req) {
        // cout << "HEAR" << endl;
        int a, b, d;
        // cout << "KEKW" << endl;
        cin >> a >> b >> d;
        // cout << "OK: " << a << " " << b << " " << d << endl;
        cout << get_ans(0, 0, n, a, b, d) << "\n";
    }
}
```



```
int32_t main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    int tests = 1;
    // cin >> tests;
    while(tests--) {
        solve();
    }
}
```

Task F ()