

Олимпиада «Ломоносов» по информатике
2025-2026 учебный год. Заключительный этап
Работа участника с id заявки 1552959, логином inf26f_156

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
1301	3:57:41	inf26f_156	7	g++	Partial solution	1	0
971	3:21:57	inf26f_156	6	g++	Partial solution	4	15
727	2:43:26	inf26f_156	5	g++	OK	103	100
468	1:55:19	inf26f_156	4	g++	OK	103	100
444	1:47:47	inf26f_156	3	g++	Partial solution	22	95
309	1:18:49	inf26f_156	2	g++	Partial solution	51	96
232	1:03:19	inf26f_156	1	g++	Partial solution	14	55

461 (четыреста шестьдесят один) технический балл
66 (шестьдесят шесть) итоговых баллов



председатель Жюри кандидат физ.-мат. наук Малышко В. В.



зам. председателя Жюри кандидат физ.-мат. наук Корныхин Е. В.

Посылка по задаче 1

```
[1] #include <bits/stdc++.h>
[2]
[3] #define ll long long
[4] #define ld long double
[5] #define re return
[6] #define all(v) (v).begin(), (v).end()
[7] #define allr(v) (v).rbegin(), (v).rend()
[8] #define endl '\n'
[9]
[10] using namespace std;
[11]
[12] const int mod = 1e9 + 7;
[13] mt19937 rnd(time(0));
[14]
[15] int bpow(int x, int p) {
[16]     int ans = 1;
[17]     while (p) {
[18]         if (p & 1) ans = (ll)ans * x % mod;
[19]         x = (ll)x * x % mod;
[20]         p >>= 1;
[21]     }
[22]     re ans;
[23] }
[24]
[25] struct stree {
[26]     int n;
[27]     vector<ll> t;
[28]
[29]     stree(int _n) {
[30]         n = _n;
[31]         t.resize(4 * n);
[32]     }
[33]
[34]     int left(int v) { re v * 2 + 1; }
[35]     int right(int v) { re v * 2 + 2; }
[36]
[37]     void upd(int v, int tl, int tr, int id, int x) {
[38]         if (tl > id || tr <= id) re;
[39]         if (tl + 1 == tr) {
[40]             t[v] = x;
[41]             re;
[42]         }
[43]         int mid = (tl + tr) / 2;
[44]         upd(left(v), tl, mid, id, x);
[45]         upd(right(v), mid, tr, id, x);
[46]         t[v] = t[left(v)] + t[right(v)];
[47]     }
[48]
[49]     ll get(int v, int tl, int tr, int ql, int qr) {
[50]         if (tl >= qr || tr <= ql) re 0;
[51]         if (tl >= ql && tr <= qr) re t[v];
[52]         int mid = (tl + tr) / 2;
[53]         re get(left(v), tl, mid, ql, qr) + get(right(v), mid, tr, ql, qr);
[54]     }
[55]
[56]     void upd(int id, int x) {
[57]         upd(0, 0, n, id, x);
[58]     }
[59]     ll get(int l, int r) {
[60]         re get(0, 0, n, l, r);
[61]     }
[62] };
[63]
[64]
[65] // LCA
[66] int _t = 0;
[67] const int L = 20;
[68] vector<int> tin, tout;
[69] vector<vector<int>> up;
[70]
[71] void build_dfs(int v, int p, vector<vector<int>> &g) {
[72]     tin[v] = _t++;
[73]     up[v][0] = p;
[74]     for (int i = 1; i <= L; i++) {
[75]         up[v][i] = up[up[v][i - 1]][i - 1];
```

```

[76]     }
[77]     for (int u: g[v]) {
[78]         if (u != p) build_dfs(u, v, g);
[79]     }
[80]     tout[v] = _t++;
[81] }
[82]
[83] void build_lca(int root, vector<vector<int>> &g) {
[84]     int n = g.size();
[85]     tin.resize(n);
[86]     tout.resize(n);
[87]     up.assign(n, vector<int>(L + 1));
[88]     build_dfs(root, root, g);
[89] }
[90]
[91] bool is_anc(int v, int u) {
[92]     re tin[v] <= tin[u] && tout[v] >= tout[u];
[93] }
[94]
[95] int lca(int v, int u) {
[96]     if (is_anc(u, v)) re u;
[97]     for (int i = L; i >= 0; i--) {
[98]         if (!is_anc(up[u][i], v)) u = up[u][i];
[99]     }
[100]     re up[u][0];
[101] }
[102] // END LCA
[103]
[104] // HASHES
[105]
[106] // const int B = rnd() % (mod - 30) + 25;
[107] // const int N = 2e5;
[108]
[109] // vector<int> p(N);
[110] // vector<int> rp(N);
[111]
[112] // void build_p() {
[113] //     p[0] = rp[0] = 1;
[114] //     int rb = bpow(B, mod - 2);
[115] //     for (int i = 1; i < N; i++) {
[116] //         p[i] = (1ll)p[i - 1] * B % mod;
[117] //         rp[i] = (1ll)rp[i - 1] * rb % mod;
[118] //     }
[119] // }
[120]
[121] // vector<int> build_ph(string &s) {
[122] //     int n = s.size();
[123] //     vector<int> ph(n);
[124] //     ph[0] = s[0] - 'a' + 1;
[125] //     for (int i = 1; i < n; i++) {
[126] //         ph[i] = ((1ll)ph[i - 1] + (1ll)(s[i] - 'a' + 1) * p[i] % mod) % mod;
[127] //     }
[128] //     re ph;
[129] // }
[130]
[131] // int get_h(int l, int r, vector<int> &ph) {
[132] //     re ((1ll)ph[r] - (l > 0 ? ph[l - 1] : 0) + mod) % mod * rp[l] % mod;
[133] // }
[134] // int get_h(string &s) {
[135] //     int n = s.size();
[136] //     int ans = 0;
[137] //     for (int i = 0; i < n; i++) {
[138] //         ans = ((1ll)ans + (1ll)(s[i] - 'a' + 1) * p[i]) % mod;
[139] //     }
[140] //     re ans;
[141] // }
[142]
[143] // END HASHES
[144]
[145] map<string, int> alph = {
[146]     {"N", 0},
[147]     {"I", 1},
[148]     {"II", 2},
[149]     {"III", 3},
[150]     {"IV", 4},
[151]     {"V", 5},
[152]     {"VI", 6},
[153]     {"VII", 7},

```

```

[154] {"VIII", 8},
[155] {"IX", 9},
[156] {"X", 10},
[157] {"XI", 11},
[158] {"XII", 12},
[159] {"XIII", 13},
[160] {"XIV", 14},
[161] {"XV", 15},
[162] {"XVI", 16},
[163] {"XVII", 17},
[164] {"XVIII", 18},
[165] {"XIX", 19},
[166] {"XX", 20},
[167] {"XXI", 21},
[168] {"XXII", 22},
[169] {"XXIII", 23},
[170] {"XXIV", 24},
[171] {"XXV", 25},
[172] {"XXVI", 26},
[173] {"XXVII", 27},
[174] {"XXVIII", 28},
[175] {"XXIX", 29},
[176] {"XXX", 30},
[177] {"XXXI", 31},
[178] {"XXXII", 32},
[179] {"XXXIII", 33},
[180] {"XXXIV", 34},
[181] {"XXXV", 35},
[182] {"XXXVI", 36},
[183] {"XXXVII", 37},
[184] {"XXXVIII", 38},
[185] {"XXXIX", 39},
[186] {"XXXX", 40},
[187] {"XXXI", 41},
[188] {"XXXII", 42},
[189] {"XXXIII", 43},
[190] {"XXXIV", 44},
[191] {"XXXV", 45},
[192] {"XXXVI", 46},
[193] {"XXXVII", 47},
[194] {"XXXVIII", 48},
[195] {"XXXIX", 49},
[196] {"L", 50},
[197] {"LI", 51}
[198] };
[199]
[200] map<int,int> a1 = {{0,65},{1,66},{2,67},{3,68},{4,69},{5,70},{6,71},{7,72},{8,73},{9,74},{10,75},{11,76},{12,77},{13,78},{14,79},{15,80},
[201] {16,81},{17,82},{18,83},{19,84},{20,85},{21,86},{22,87},{23,88},{24,89},{25,90},{26,97},{27,98},{28,99},{29,100},{30,101},{31,102},{32,103},
[202] {33,104},{34,105},{35,106},{36,107},{37,108},{38,109},{39,110},{40,111},{41,112},{42,113},{43,114},{44,115},{45,116},{46,117},{47,118},
[203] {48,119},{49,120},{50,121},{51,122}};
[204]
[205] void norm(string &s) {
[206]
[207]     if (s.find("XXXXXL") != -1)
[208]         s.replace(s.find("XXXXXL"), 6, "N");
[209]     if (s.find("XXXXL") != -1)
[210]         s.replace(s.find("XXXXL"), 5, "X");
[211]     if (s.find("XXXL") != -1)
[212]         s.replace(s.find("XXXL"), 4, "XX");
[213]     if (s.find("XXL") != -1)
[214]         s.replace(s.find("XXL"), 3, "XXX");
[215]
[216]     if (s.find("IIIIIIIIIX") != -1)
[217]         s.replace(s.find("IIIIIIIIIX"), 11, "N");
[218]     if (s.find("IIIIIIIIIX") != -1)
[219]         s.replace(s.find("IIIIIIIIIX"), 10, "I");
[220]     if (s.find("IIIIIIIIIX") != -1)
[221]         s.replace(s.find("IIIIIIIIIX"), 9, "II");
[222]     if (s.find("IIIIIIIX") != -1)
[223]         s.replace(s.find("IIIIIIIX"), 8, "III");
[224]     if (s.find("IIIIIIIX") != -1)
[225]         s.replace(s.find("IIIIIIIX"), 7, "IV");
[226]     if (s.find("IIIIIX") != -1)
[227]         s.replace(s.find("IIIIIX"), 6, "V");
[228]     if (s.find("IIIIIX") != -1)
[229]         s.replace(s.find("IIIIIX"), 5, "VI");
[230]     if (s.find("IIIX") != -1)
[231]         s.replace(s.find("IIIX"), 4, "VII");
[232]     if (s.find("IIX") != -1)
[233]         s.replace(s.find("IIX"), 3, "VIII");

```

```

[234]
[235]     if (s.find("IIIIIV") != -1)
[236]         s.replace(s.find("IIIIIV"), 6, "N");
[237]     if (s.find("IIIIIV") != -1)
[238]         s.replace(s.find("IIIIIV"), 5, "I");
[239]     if (s.find("IIIV") != -1)
[240]         s.replace(s.find("IIIV"), 4, "II");
[241]     if (s.find("IIV") != -1)
[242]         s.replace(s.find("IIV"), 3, "III");
[243]
[244]     if (s.size() > 1 && s[0] == 'N') {
[245]         reverse(all(s));
[246]         while (s.size() > 1 && s.back() == 'N') {
[247]             s.pop_back();
[248]         }
[249]         reverse(all(s));
[250]     }
[251] }
[252]
[253] vector<int> to_dig(string &s) {
[254]     vector<string> a;
[255]     string t;
[256]     for (int i = 0; i < s.size(); i++) {
[257]         if (s[i] != '.') t.push_back(s[i]);
[258]         else {
[259]             a.push_back(t);
[260]             t.clear();
[261]         }
[262]     }
[263]     a.push_back(t);
[264]     int n = a.size();
[265]     for(int i = 0; i < n; i++) {
[266]         norm(a[i]);
[267]     }
[268]     vector<int> ans(n);
[269]     for (int i = 0; i < n; i++) {
[270]         ans[i] = alph[a[i]];
[271]     }
[272]     re ans;
[273] }
[274]
[275] int main() {
[276]     ios::sync_with_stdio(0);
[277]     cin.tie(); cout.tie();
[278]
[279] #ifdef LOCAL
[280]     freopen("input.txt", "r", stdin);
[281]     freopen("output.txt", "w", stdout);
[282] #endif // LOCAL
[283]
[284]     int n;
[285]     cin >> n;
[286]     vector<vector<int>> a(n);
[287]     for (int i = 0; i < n; i++) {
[288]         string s;
[289]         cin >> s;
[290]         a[i] = to_dig(s);
[291]     }
[292]     vector<string> ans(n);
[293]     for (int i = 0; i < n; i++) {
[294]         for (int x: a[i]) {
[295]             ans[i].push_back((char)al[x]);
[296]         }
[297]     }
[298]     sort(all(ans), [](string &s, string &t) {
[299]         if (s.size() != t.size()) re s.size() > t.size();
[300]         else re s > t;
    });
    cout << ans.back() << endl << ans.front();
}

```

Посылка по задаче 2

```
[1] #include <bits/stdc++.h>
[2]
[3] #define ll long long
[4] #define ld long double
[5] #define re return
[6] #define all(v) (v).begin(), (v).end()
[7] #define allr(v) (v).rbegin(), (v).rend()
[8] #define endl '\n'
[9]
[10] using namespace std;
[11]
[12] const int mod = 1e9 + 7;
[13] mt19937 rnd(time(0));
[14]
[15] int bpow(int x, int p) {
[16]     int ans = 1;
[17]     while (p) {
[18]         if (p & 1) ans = (ll)ans * x % mod;
[19]         x = (ll)x * x % mod;
[20]         p >>= 1;
[21]     }
[22]     re ans;
[23] }
[24]
[25] struct stree {
[26]     int n;
[27]     vector<ll> t;
[28]
[29]     stree(int _n) {
[30]         n = _n;
[31]         t.resize(4 * n);
[32]     }
[33]
[34]     int left(int v) { re v * 2 + 1; }
[35]     int right(int v) { re v * 2 + 2; }
[36]
[37]     void upd(int v, int tl, int tr, int id, int x) {
[38]         if (tl > id || tr <= id) re;
[39]         if (tl + 1 == tr) {
[40]             t[v] = x;
[41]             re;
[42]         }
[43]         int mid = (tl + tr) / 2;
[44]         upd(left(v), tl, mid, id, x);
[45]         upd(right(v), mid, tr, id, x);
[46]         t[v] = t[left(v)] + t[right(v)];
[47]     }
[48]
[49]     ll get(int v, int tl, int tr, int ql, int qr) {
[50]         if (tl >= qr || tr <= ql) re 0;
[51]         if (tl >= ql && tr <= qr) re t[v];
[52]         int mid = (tl + tr) / 2;
[53]         re get(left(v), tl, mid, ql, qr) + get(right(v), mid, tr, ql, qr);
[54]     }
[55]
[56]     void upd(int id, int x) {
[57]         upd(0, 0, n, id, x);
[58]     }
[59]     ll get(int l, int r) {
[60]         re get(0, 0, n, l, r);
[61]     }
[62] };
[63]
[64]
[65] // LCA
[66] int _t = 0;
[67] const int L = 20;
[68] vector<int> tin, tout;
[69] vector<vector<int>> up;
[70]
[71] void build_dfs(int v, int p, vector<vector<int>> &g) {
[72]     tin[v] = _t++;
[73]     up[v][0] = p;
[74]     for (int i = 1; i <= L; i++) {
[75]         up[v][i] = up[up[v][i - 1]][i - 1];
[76]     }
```

```

[77]     for (int u: g[v]) {
[78]         if (u != p) build_dfs(u, v, g);
[79]     }
[80]     tout[v] = _t++;
[81] }
[82]
[83] void build_lca(int root, vector<vector<int>> &g) {
[84]     int n = g.size();
[85]     tin.resize(n);
[86]     tout.resize(n);
[87]     up.assign(n, vector<int>(L + 1));
[88]     build_dfs(root, root, g);
[89] }
[90]
[91] bool is_anc(int v, int u) {
[92]     re tin[v] <= tin[u] && tout[v] >= tout[u];
[93] }
[94]
[95] int lca(int v, int u) {
[96]     if (is_anc(u, v)) re u;
[97]     for (int i = L; i >= 0; i--) {
[98]         if (!is_anc(up[u][i], v)) u = up[u][i];
[99]     }
[100]     re up[u][0];
[101] }
[102] // END LCA
[103]
[104] // HASHES
[105]
[106] // const int B = rnd() % (mod - 30) + 25;
[107] // const int N = 2e5;
[108]
[109] // vector<int> p(N);
[110] // vector<int> rp(N);
[111]
[112] // void build_p() {
[113] //     p[0] = rp[0] = 1;
[114] //     int rb = bpow(B, mod - 2);
[115] //     for (int i = 1; i < N; i++) {
[116] //         p[i] = (1ll)p[i - 1] * B % mod;
[117] //         rp[i] = (1ll)rp[i - 1] * rb % mod;
[118] //     }
[119] // }
[120]
[121] // vector<int> build_ph(string &s) {
[122] //     int n = s.size();
[123] //     vector<int> ph(n);
[124] //     ph[0] = s[0] - 'a' + 1;
[125] //     for (int i = 1; i < n; i++) {
[126] //         ph[i] = ((1ll)ph[i - 1] + (1ll)(s[i] - 'a' + 1) * p[i] % mod) % mod;
[127] //     }
[128] //     re ph;
[129] // }
[130]
[131] // int get_h(int l, int r, vector<int> &ph) {
[132] //     re ((1ll)ph[r] - (l > 0 ? ph[l - 1] : 0) + mod) % mod * rp[l] % mod;
[133] // }
[134] // int get_h(string &s) {
[135] //     int n = s.size();
[136] //     int ans = 0;
[137] //     for (int i = 0; i < n; i++) {
[138] //         ans = ((1ll)ans + (1ll)(s[i] - 'a' + 1) * p[i]) % mod;
[139] //     }
[140] //     re ans;
[141] // }
[142]
[143] // END HASHES
[144]
[145] const int N = 170000;
[146] vector<bool> resh(N);
[147]
[148] void build() {
[149]     resh[0] = resh[1] = 1;
[150]     for (int i = 2; i < N; i++) {
[151]         if (!resh[i]) {
[152]             for (ll j = (1ll)i * i; j < N; j+=i) {
[153]                 resh[j] = 1;
[154]             }
[155]         }
[156]     }
[157] }

```

```

[158]
[159] int main() {
[160]     ios::sync_with_stdio(0);
[161]     cin.tie(); cout.tie();
[162]
[163] #ifdef LOCAL
[164]     freopen("input.txt", "r", stdin);
[165]     freopen("output.txt", "w", stdout);
[166] #endif // LOCAL
[167]
[168]     int n;
[169]     cin >> n;
[170]     vector<int> a(n);
[171]     for (int i = 0; i < n; i++) {
[172]         cin >> a[i];
[173]     }
[174]     build();
[175]     map<int,int> mp;
[176]     for (int i = 0; i < n; i++) {
[177]         if (!resh[a[i]]) {
[178]             mp[a[i]]++;
[179]         }
[180]     }
[181]     if (mp.size() == 0) {
[182]         cout << 0;
[183]         re 0;
[184]     }
[185]     int ans = 0;
[186]     for (auto p: mp) {
[187]         if (p.second > mp[ans]) ans = p.first;
[188]         if (p.second == mp[ans]) ans = max(ans, p.first);
[189]     }
[190]     cout << ans;
[191] }

```

Посылка по задаче 3

```
[1] #include <bits/stdc++.h>
[2]
[3] #define ll long long
[4] #define ld long double
[5] #define re return
[6] #define all(v) (v).begin(), (v).end()
[7] #define allr(v) (v).rbegin(), (v).rend()
[8] #define endl '\n'
[9]
[10] using namespace std;
[11]
[12] const int mod = 1e9 + 7;
[13] mt19937 rnd(time(0));
[14]
[15] int bpow(int x, int p) {
[16]     int ans = 1;
[17]     while (p) {
[18]         if (p & 1) ans = (ll)ans * x % mod;
[19]         x = (ll)x * x % mod;
[20]         p >>= 1;
[21]     }
[22]     re ans;
[23] }
[24]
[25] struct stree {
[26]     int n;
[27]     vector<ll> t;
[28]
[29]     stree(int _n) {
[30]         n = _n;
[31]         t.resize(4 * n);
[32]     }
[33]
[34]     int left(int v) { re v * 2 + 1; }
[35]     int right(int v) { re v * 2 + 2; }
[36]
[37]     void upd(int v, int tl, int tr, int id, int x) {
[38]         if (tl > id || tr <= id) re;
[39]         if (tl + 1 == tr) {
[40]             t[v] = x;
[41]             re;
[42]         }
[43]         int mid = (tl + tr) / 2;
[44]         upd(left(v), tl, mid, id, x);
[45]         upd(right(v), mid, tr, id, x);
[46]         t[v] = t[left(v)] + t[right(v)];
[47]     }
[48]
[49]     ll get(int v, int tl, int tr, int ql, int qr) {
[50]         if (tl >= qr || tr <= ql) re 0;
[51]         if (tl >= ql && tr <= qr) re t[v];
[52]         int mid = (tl + tr) / 2;
[53]         re get(left(v), tl, mid, ql, qr) + get(right(v), mid, tr, ql, qr);
[54]     }
[55]
[56]     void upd(int id, int x) {
[57]         upd(0, 0, n, id, x);
[58]     }
[59]     ll get(int l, int r) {
[60]         re get(0, 0, n, l, r);
[61]     }
[62] };
[63]
[64]
[65] // LCA
[66] int _t = 0;
[67] const int L = 20;
[68] vector<int> tin, tout;
[69] vector<vector<int>> up;
[70]
[71] void build_dfs(int v, int p, vector<vector<int>> &g) {
[72]     tin[v] = _t++;
[73]     up[v][0] = p;
[74]     for (int i = 1; i <= L; i++) {
[75]         up[v][i] = up[up[v][i - 1]][i - 1];
[76]     }
```

```

[77]     for (int u: g[v]) {
[78]         if (u != p) build_dfs(u, v, g);
[79]     }
[80]     tout[v] = _t++;
[81] }
[82]
[83] void build_lca(int root, vector<vector<int>> &g) {
[84]     int n = g.size();
[85]     tin.resize(n);
[86]     tout.resize(n);
[87]     up.assign(n, vector<int>(L + 1));
[88]     build_dfs(root, root, g);
[89] }
[90]
[91] bool is_anc(int v, int u) {
[92]     re tin[v] <= tin[u] && tout[v] >= tout[u];
[93] }
[94]
[95] int lca(int v, int u) {
[96]     if (is_anc(u, v)) re u;
[97]     for (int i = L; i >= 0; i--) {
[98]         if (!is_anc(up[u][i], v)) u = up[u][i];
[99]     }
[100]     re up[u][0];
[101] }
[102] // END LCA
[103]
[104] // HASHES
[105]
[106] // const int B = rnd() % (mod - 30) + 25;
[107] // const int N = 2e5;
[108]
[109] // vector<int> p(N);
[110] // vector<int> rp(N);
[111]
[112] // void build_p() {
[113] //     p[0] = rp[0] = 1;
[114] //     int rb = bpow(B, mod - 2);
[115] //     for (int i = 1; i < N; i++) {
[116] //         p[i] = (1ll)p[i - 1] * B % mod;
[117] //         rp[i] = (1ll)rp[i - 1] * rb % mod;
[118] //     }
[119] // }
[120]
[121] // vector<int> build_ph(string &s) {
[122] //     int n = s.size();
[123] //     vector<int> ph(n);
[124] //     ph[0] = s[0] - 'a' + 1;
[125] //     for (int i = 1; i < n; i++) {
[126] //         ph[i] = ((1ll)ph[i - 1] + (1ll)(s[i] - 'a' + 1) * p[i] % mod) % mod;
[127] //     }
[128] //     re ph;
[129] // }
[130]
[131] // int get_h(int l, int r, vector<int> &ph) {
[132] //     re ((1ll)ph[r] - (l > 0 ? ph[l - 1] : 0) + mod) % mod * rp[l] % mod;
[133] // }
[134] // int get_h(string &s) {
[135] //     int n = s.size();
[136] //     int ans = 0;
[137] //     for (int i = 0; i < n; i++) {
[138] //         ans = ((1ll)ans + (1ll)(s[i] - 'a' + 1) * p[i]) % mod;
[139] //     }
[140] //     re ans;
[141] // }
[142]
[143] // END HASHES
[144]
[145] const int N = 170000;
[146] vector<bool> resh(N);
[147]
[148] void build() {
[149]     resh[0] = resh[1] = 1;
[150]     for (int i = 2; i < N; i++) {
[151]         if (!resh[i]) {
[152]             for (ll j = (1ll)i * i; j < N; j+=i) {
[153]                 resh[j] = 1;
[154]             }
[155]         }
[156]     }
[157] }

```

```

[158]
[159] int main() {
[160]     ios::sync_with_stdio(0);
[161]     cin.tie(); cout.tie();
[162]
[163] #ifdef LOCAL
[164]     freopen("input.txt", "r", stdin);
[165]     freopen("output.txt", "w", stdout);
[166] #endif // LOCAL
[167]
[168]     ll n, x, y;
[169]     cin >> n >> y >> x;
[170]     ll r = 0;
[171]     if (min(x - 1, n - x) <= min(y - 1, n - y)) {
[172]         r = min(x - 1, n - x);
[173]     } else {
[174]         r = min(y - 1, n - y);
[175]     }
[176]     ll s = (4 * n - 4) * r;
[177]     for (int i = 1; i < r; i++) {
[178]         s -= 8 * i;
[179]     }
[180]     ll ans = 0;
[181]     if (x == r + 1) {
[182]         ans = s + 3 * (n - 2 * r) - 3;
[183]         ans += n - y - r;
[184]     } else if (x == n - r) {
[185]         ans = s + n - 2 * r - 1;
[186]         ans += y - 1 - r;
[187]     } else if (y == r + 1) {
[188]         ans = s + x - 1 - r;
[189]     } else {
[190]         ans = s + 2 * (n - 2 * r) - 2;
[191]         ans += n - x - r;
[192]     }
[193]     cout << ans;
[194] }

```

Посылка по задаче 4

```
[1] #include <bits/stdc++.h>
[2]
[3] #define ll long long
[4] #define ld long double
[5] #define re return
[6] #define all(v) (v).begin(), (v).end()
[7] #define allr(v) (v).rbegin(), (v).rend()
[8] #define endl '\n'
[9]
[10] using namespace std;
[11]
[12] const int mod = 1e9 + 7;
[13] mt19937 rnd(time(0));
[14]
[15] int bpow(int x, int p) {
[16]     int ans = 1;
[17]     while (p) {
[18]         if (p & 1) ans = (ll)ans * x % mod;
[19]         x = (ll)x * x % mod;
[20]         p >>= 1;
[21]     }
[22]     re ans;
[23] }
[24]
[25] struct stree {
[26]     int n;
[27]     vector<ll> t;
[28]
[29]     stree(int _n) {
[30]         n = _n;
[31]         t.resize(4 * n);
[32]     }
[33]
[34]     int left(int v) { re v * 2 + 1; }
[35]     int right(int v) { re v * 2 + 2; }
[36]
[37]     void upd(int v, int tl, int tr, int id, int x) {
[38]         if (tl > id || tr <= id) re;
[39]         if (tl + 1 == tr) {
[40]             t[v] = x;
[41]             re;
[42]         }
[43]         int mid = (tl + tr) / 2;
[44]         upd(left(v), tl, mid, id, x);
[45]         upd(right(v), mid, tr, id, x);
[46]         t[v] = t[left(v)] + t[right(v)];
[47]     }
[48]
[49]     ll get(int v, int tl, int tr, int ql, int qr) {
[50]         if (tl >= qr || tr <= ql) re 0;
[51]         if (tl >= ql && tr <= qr) re t[v];
[52]         int mid = (tl + tr) / 2;
[53]         re get(left(v), tl, mid, ql, qr) + get(right(v), mid, tr, ql, qr);
[54]     }
[55]
[56]     void upd(int id, int x) {
[57]         upd(0, 0, n, id, x);
[58]     }
[59]     ll get(int l, int r) {
[60]         re get(0, 0, n, l, r);
[61]     }
[62] };
[63]
[64]
[65] // LCA
[66] int _t = 0;
[67] const int L = 20;
[68] vector<int> tin, tout;
[69] vector<vector<int>> up;
[70]
[71] void build_dfs(int v, int p, vector<vector<int>> &g) {
[72]     tin[v] = _t++;
[73]     up[v][0] = p;
[74]     for (int i = 1; i <= L; i++) {
[75]         up[v][i] = up[up[v][i - 1]][i - 1];
[76]     }
```

```

[77]     for (int u: g[v]) {
[78]         if (u != p) build_dfs(u, v, g);
[79]     }
[80]     tout[v] = _t++;
[81] }
[82]
[83] void build_lca(int root, vector<vector<int>> &g) {
[84]     int n = g.size();
[85]     tin.resize(n);
[86]     tout.resize(n);
[87]     up.assign(n, vector<int>(L + 1));
[88]     build_dfs(root, root, g);
[89] }
[90]
[91] bool is_anc(int v, int u) {
[92]     re tin[v] <= tin[u] && tout[v] >= tout[u];
[93] }
[94]
[95] int lca(int v, int u) {
[96]     if (is_anc(u, v)) re u;
[97]     for (int i = L; i >= 0; i--) {
[98]         if (!is_anc(up[u][i], v)) u = up[u][i];
[99]     }
[100]     re up[u][0];
[101] }
[102] // END LCA
[103]
[104] // HASHES
[105]
[106] // const int B = rnd() % (mod - 30) + 25;
[107] // const int N = 2e5;
[108]
[109] // vector<int> p(N);
[110] // vector<int> rp(N);
[111]
[112] // void build_p() {
[113] //     p[0] = rp[0] = 1;
[114] //     int rb = bpow(B, mod - 2);
[115] //     for (int i = 1; i < N; i++) {
[116] //         p[i] = (ll)p[i - 1] * B % mod;
[117] //         rp[i] = (ll)rp[i - 1] * rb % mod;
[118] //     }
[119] // }
[120]
[121] // vector<int> build_ph(string &s) {
[122] //     int n = s.size();
[123] //     vector<int> ph(n);
[124] //     ph[0] = s[0] - 'a' + 1;
[125] //     for (int i = 1; i < n; i++) {
[126] //         ph[i] = ((ll)ph[i - 1] + (ll)(s[i] - 'a' + 1) * p[i] % mod) % mod;
[127] //     }
[128] //     re ph;
[129] // }
[130]
[131] // int get_h(int l, int r, vector<int> &ph) {
[132] //     re ((ll)ph[r] - (l > 0 ? ph[l - 1] : 0) + mod) % mod * rp[l] % mod;
[133] // }
[134] // int get_h(string &s) {
[135] //     int n = s.size();
[136] //     int ans = 0;
[137] //     for (int i = 0; i < n; i++) {
[138] //         ans = ((ll)ans + (ll)(s[i] - 'a' + 1) * p[i] % mod;
[139] //     }
[140] //     re ans;
[141] // }
[142]
[143] // END HASHES
[144]
[145] int main() {
[146]     ios::sync_with_stdio(0);
[147]     cin.tie(); cout.tie();
[148]
[149] #ifdef LOCAL
[150]     freopen("input.txt", "r", stdin);
[151]     freopen("output.txt", "w", stdout);
[152] #endif // LOCAL

```

```
[153]
[154]     int n;
[155]     cin >> n;
[156]     vector<int> a(n);
[157]     for (int i = 0; i < n; i++) {
[158]         cin >> a[i];
[159]     }
[160]     reverse(all(a));
[161]     vector<int> dp(n, INT_MAX);
[162]     for (int i = 0; i < n; i++) {
[163]         int id = upper_bound(all(dp), a[i]) - dp.begin();
[164]         dp[id] = a[i];
[165]     }
[166]     int ans = 0;
[167]     for (int i = 0; i < n; i++) {
[168]         if (dp[i] != INT_MAX) ans = i + 1;
[169]     }
[170]     cout << n - ans;
[171] }
```

Посылка по задаче 5

```
[1] #include <bits/stdc++.h>
[2]
[3] #define ll long long
[4] #define ld long double
[5] #define re return
[6] #define all(v) (v).begin(), (v).end()
[7] #define allr(v) (v).rbegin(), (v).rend()
[8] #define endl '\n'
[9]
[10] using namespace std;
[11]
[12] const int mod = 1e9 + 7;
[13] mt19937 rnd(time(0));
[14]
[15] int bpow(int x, int p) {
[16]     int ans = 1;
[17]     while (p) {
[18]         if (p & 1) ans = (ll)ans * x % mod;
[19]         x = (ll)x * x % mod;
[20]         p >>= 1;
[21]     }
[22]     re ans;
[23] }
[24]
[25] struct node {
[26]     int mx = 0;
[27]     node *l = nullptr, *r = nullptr;
[28]
[29]     node() {};
[30]
[31]     node* left() {
[32]         if (!l) l = new node;
[33]         re l;
[34]     }
[35]     node* right() {
[36]         if (!r) r = new node;
[37]         re r;
[38]     }
[39] };
[40]
[41] struct stree1
[42] {
[43]     ll n;
[44]     node* root;
[45]
[46]     stree1() {
[47]         n = 1e9 + 10;
[48]         root = new node;
[49]     }
[50]
[51]     void upd(node *v, ll tl, ll tr, ll id, int x) {
[52]         if (tl > id || tr <= id) re;
[53]         if (tl + 1 == tr) {
[54]             v->mx = max(v->mx, x);
[55]             re;
[56]         }
[57]         ll mid = (tl + tr) / 2;
[58]         upd(v->left(), tl, mid, id, x);
[59]         upd(v->right(), mid, tr, id, x);
[60]         v->mx = max(v->left()->mx, v->right()->mx);
[61]     }
[62]
[63]     int get(node *v, ll tl, ll tr, ll ql, ll qr) {
[64]         if (tl >= qr || tr <= ql) re 0;
[65]         if (tl >= ql && tr <= qr) re v->mx;
[66]         ll mid = (tl + tr) / 2;
[67]         re max(get(v->left(), tl, mid, ql, qr), get(v->right(), mid, tr, ql, qr));
[68]     }
[69]
[70]     void upd(int id, int x) {
[71]         upd(root, 0, n, id, x);
[72]     }
[73]     int get(ll l, ll r) {
[74]         re get(root, 0, n, l, r);
[75]     }
[76] };
[77]
```

```

[78]
[79] struct stree {
[80]     int n;
[81]     vector<stree1> t;
[82]     vector<int> res;
[83]
[84]     stree(int _n) {
[85]         n = _n;
[86]         t.resize(4 * n);
[87]     }
[88]
[89]     int left(int v) { re v * 2 + 1; }
[90]     int right(int v) { re v * 2 + 2; }
[91]
[92]     void upd(int v, int tl, int tr, int id, int h, int x) {
[93]         if (tl > id || tr <= id) re;
[94]         if (tl + 1 == tr) {
[95]             t[v].upd(h, x);
[96]             re;
[97]         }
[98]         t[v].upd(h, x);
[99]         int mid = (tl + tr) / 2;
[100]         upd(left(v), tl, mid, id, h, x);
[101]         upd(right(v), mid, tr, id, h, x);
[102]     }
[103]
[104]     int get(int v, int tl, int tr, int ql, int qr, int h) {
[105]         if (tl >= qr || tr <= ql) re 0;
[106]         if (tl >= ql && tr <= qr) re t[v].get(0, h);
[107]         int mid = (tl + tr) / 2;
[108]         re max(get(left(v), tl, mid, ql, qr, h), get(right(v), mid, tr, ql, qr, h));
[109]     }
[110]
[111]     void upd(int id, int h, int x) {
[112]         upd(0, 0, n, id, h, x);
[113]     }
[114]     int get(int l, int r, int h) {
[115]         re get(0, 0, n, l, r, h);
[116]     }
[117] };
[118]
[119]
[120] // LCA
[121] int _t = 0;
[122] const int L = 20;
[123] vector<int> tin, tout;
[124] vector<vector<int>> up;
[125]
[126] void build_dfs(int v, int p, vector<vector<int>> &g) {
[127]     tin[v] = _t++;
[128]     up[v][0] = p;
[129]     for (int i = 1; i <= L; i++) {
[130]         up[v][i] = up[up[v][i - 1]][i - 1];
[131]     }
[132]     for (int u: g[v]) {
[133]         if (u != p) build_dfs(u, v, g);
[134]     }
[135]     tout[v] = _t++;
[136] }
[137]
[138] void build_lca(int root, vector<vector<int>> &g) {
[139]     int n = g.size();
[140]     tin.resize(n);
[141]     tout.resize(n);
[142]     up.assign(n, vector<int>(L + 1));
[143]     build_dfs(root, root, g);
[144] }
[145]
[146] bool is_anc(int v, int u) {
[147]     re tin[v] <= tin[u] && tout[v] >= tout[u];
[148] }
[149]
[150] int lca(int v, int u) {
[151]     if (is_anc(u, v)) re u;
[152]     for (int i = L; i >= 0; i--) {
[153]         if (!is_anc(up[u][i], v)) u = up[u][i];
[154]     }

```

```

[155]     re up[u][0];
[156] }
[157] // END LCA
[158]
[159] // HASHES
[160]
[161] // const int B = rnd() % (mod - 30) + 25;
[162] // const int N = 2e5;
[163]
[164] // vector<int> p(N);
[165] // vector<int> rp(N);
[166] // void build_p() {
[167] //     p[0] = rp[0] = 1;
[168] //     int rb = bpow(B, mod - 2);
[169] //     for (int i = 1; i < N; i++) {
[170] //         p[i] = (1ll)p[i - 1] * B % mod;
[171] //         rp[i] = (1ll)rp[i - 1] * rb % mod;
[172] //     }
[173] // }
[174]
[175] // vector<int> build_ph(string &s) {
[176] //     int n = s.size();
[177] //     vector<int> ph(n);
[178] //     ph[0] = s[0] - 'a' + 1;
[179] //     for (int i = 1; i < n; i++) {
[180] //         ph[i] = ((1ll)ph[i - 1] + (1ll)(s[i] - 'a' + 1) * p[i] % mod) % mod;
[181] //     }
[182] //     re ph;
[183] // }
[184]
[185] // int get_h(int l, int r, vector<int> &ph) {
[186] //     re ((1ll)ph[r] - (l > 0 ? ph[l - 1] : 0) + mod) % mod * rp[l] % mod;
[187] // }
[188] // int get_h(string &s) {
[189] //     int n = s.size();
[190] //     int ans = 0;
[191] //     for (int i = 0; i < n; i++) {
[192] //         ans = ((1ll)ans + (1ll)(s[i] - 'a' + 1) * p[i]) % mod;
[193] //     }
[194] //     re ans;
[195] // }
[196]
[197] // END HASHES
[198]
[199] int main() {
[200]     ios::sync_with_stdio(0);
[201]     cin.tie(); cout.tie();
[202]
[203] #ifdef LOCAL
[204]     freopen("input.txt", "r", stdin);
[205]     freopen("output.txt", "w", stdout);
[206] #endif // LOCAL
[207]
[208]     int n;
[209]     cin >> n;
[210]     vector<vector<int>> h(n);
[211]     for (int i = 0; i < n; i++) {
[212]         int x, y, z;
[213]         cin >> x >> y >> z;
[214]         h[i] = {x, y, z};
[215]     }
[216]     sort(all(h), [](vector<int>&a, vector<int>&b) {
[217]         if (a[0] != b[0]) re a[0] > b[0];
[218]         re a[1] > b[1];
[219]     });
[220]     vector<int> dp(n);
[221]     for (int i = 0; i < n; i++) {
[222]         int mx = 0;
[223]         for (int j = 0; j < n; j++) {
[224]             if (i == j) continue;
[225]             if (h[j][2] < h[i][2] && h[j][0] > h[i][0] && h[j][1] > h[i][1]) mx = max(mx, dp[j]);
[226]         }
[227]         dp[i] = mx + 1;
[228]     }
[229]     int ans = 0;
[230]     for (int i = 0; i < n; i++) {
[231]         ans = max(ans, dp[i]);
[232]     }
[233]     cout << ans;
[234] }

```

Посылка по задаче 6

```
[1] #include <bits/stdc++.h>
[2]
[3] #define ll long long
[4] #define ld long double
[5] #define re return
[6] #define all(v) (v).begin(), (v).end()
[7] #define allr(v) (v).rbegin(), (v).rend()
[8] #define endl '\n'
[9]
[10] using namespace std;
[11]
[12] const int mod = 1e9 + 7;
[13] mt19937 rnd(time(0));
[14]
[15] int bpow(int x, int p) {
[16]     int ans = 1;
[17]     while (p) {
[18]         if (p & 1) ans = (ll)ans * x % mod;
[19]         x = (ll)x * x % mod;
[20]         p >>= 1;
[21]     }
[22]     re ans;
[23] }
[24]
[25] struct stree {
[26]     int n;
[27]     vector<ll> t;
[28]
[29]     stree(int _n) {
[30]         n = _n;
[31]         t.resize(4 * n);
[32]     }
[33]
[34]     int left(int v) { re v * 2 + 1; }
[35]     int right(int v) { re v * 2 + 2; }
[36]
[37]     void upd(int v, int tl, int tr, int id, int x) {
[38]         if (tl > id || tr <= id) re;
[39]         if (tl + 1 == tr) {
[40]             t[v] = x;
[41]             re;
[42]         }
[43]         int mid = (tl + tr) / 2;
[44]         upd(left(v), tl, mid, id, x);
[45]         upd(right(v), mid, tr, id, x);
[46]         t[v] = t[left(v)] + t[right(v)];
[47]     }
[48]
[49]     ll get(int v, int tl, int tr, int ql, int qr) {
[50]         if (tl >= qr || tr <= ql) re 0;
[51]         if (tl >= ql && tr <= qr) re t[v];
[52]         int mid = (tl + tr) / 2;
[53]         re get(left(v), tl, mid, ql, qr) + get(right(v), mid, tr, ql, qr);
[54]     }
[55]
[56]     void upd(int id, int x) {
[57]         upd(0, 0, n, id, x);
[58]     }
[59]     ll get(int l, int r) {
[60]         re get(0, 0, n, l, r);
[61]     }
[62] };
[63]
[64]
[65] // LCA
[66] int _t = 0;
[67] const int L = 20;
[68] vector<int> tin, tout;
[69] vector<vector<int>> up;
[70]
[71] void build_dfs(int v, int p, vector<vector<int>> &g) {
[72]     tin[v] = _t++;
[73]     up[v][0] = p;
[74]     for (int i = 1; i <= L; i++) {
[75]         up[v][i] = up[up[v][i - 1]][i - 1];
[76]     }
```

```

[77]     for (int u: g[v]) {
[78]         if (u != p) build_dfs(u, v, g);
[79]     }
[80]     tout[v] = _t++;
[81] }
[82]
[83] void build_lca(int root, vector<vector<int>> &g) {
[84]     int n = g.size();
[85]     tin.resize(n);
[86]     tout.resize(n);
[87]     up.assign(n, vector<int>(L + 1));
[88]     build_dfs(root, root, g);
[89] }
[90]
[91] bool is_anc(int v, int u) {
[92]     re tin[v] <= tin[u] && tout[v] >= tout[u];
[93] }
[94]
[95] int lca(int v, int u) {
[96]     if (is_anc(u, v)) re u;
[97]     for (int i = L; i >= 0; i--) {
[98]         if (!is_anc(up[u][i], v)) u = up[u][i];
[99]     }
[100]     re up[u][0];
[101] }
[102] // END LCA
[103]
[104] // HASHES
[105]
[106] // const int B = rnd() % (mod - 30) + 25;
[107] // const int N = 2e5;
[108]
[109] // vector<int> p(N);
[110] // vector<int> rp(N);
[111]
[112] // void build_p() {
[113] //     p[0] = rp[0] = 1;
[114] //     int rb = bpow(B, mod - 2);
[115] //     for (int i = 1; i < N; i++) {
[116] //         p[i] = (1ll)p[i - 1] * B % mod;
[117] //         rp[i] = (1ll)rp[i - 1] * rb % mod;
[118] //     }
[119] // }
[120]
[121] // vector<int> build_ph(string &s) {
[122] //     int n = s.size();
[123] //     vector<int> ph(n);
[124] //     ph[0] = s[0] - 'a' + 1;
[125] //     for (int i = 1; i < n; i++) {
[126] //         ph[i] = ((1ll)ph[i - 1] + (1ll)(s[i] - 'a' + 1) * p[i] % mod) % mod;
[127] //     }
[128] //     re ph;
[129] // }
[130]
[131] // int get_h(int l, int r, vector<int> &ph) {
[132] //     re ((1ll)ph[r] - (l > 0 ? ph[l - 1] : 0) + mod) % mod * rp[l] % mod;
[133] // }
[134] // int get_h(string &s) {
[135] //     int n = s.size();
[136] //     int ans = 0;
[137] //     for (int i = 0; i < n; i++) {
[138] //         ans = ((1ll)ans + (1ll)(s[i] - 'a' + 1) * p[i]) % mod;
[139] //     }
[140] //     re ans;
[141] // }
[142]
[143] // END HASHES
[144]
[145] int get_ans(int n1, int n2, int n3, int n4, int r, int c, vector<vector<bool>> &a) {
[146]     if (n1 == 0 && n2 == 0 && n3 == 0 && n4 == 0) re 1;
[147]     ll ans = 0;
[148]     if (n1) {
[149]         for (int i = 0; i < r; i++) {
[150]             for (int j = 0; j < c; j++) {
[151]                 if (a[i][j]) {
[152]                     a[i][j] = 0;
[153]                     ans += get_ans(n1 - 1, n2, n3, n4, r, c, a);
[154]                     a[i][j] = 1;
[155]                 }
[156]             }
[157]         }

```

```

[158] } if (n2) {
[159]     for (int i = 0; i < r; i++) {
[160]         for (int j = 0; j < c; j++) {
[161]             if (i + 1 < r) {
[162]                 bool fl = true;
[163]                 for (int k = i; k <= i + 1; k++) {
[164]                     fl = fl && a[k][j];
[165]                 }
[166]                 if (fl) {
[167]                     for (int k = i; k <= i + 1; k++) {
[168]                         a[k][j] = 0;
[169]                     }
[170]                     ans += get_ans(n1, n2 - 1, n3, n4, r, c, a);
[171]                     for (int k = i; k <= i + 1; k++) {
[172]                         a[k][j] = 1;
[173]                     }
[174]                 }
[175]             }
[176]             if (j + 1 < c) {
[177]                 bool fl = true;
[178]                 for (int k = j; k <= j + 1; k++) {
[179]                     fl = fl && a[i][k];
[180]                 }
[181]                 if (fl) {
[182]                     for (int k = j; k <= j + 1; k++) {
[183]                         a[i][k] = 0;
[184]                     }
[185]                     ans += get_ans(n1, n2 - 1, n3, n4, r, c, a);
[186]                     for (int k = j; k <= j + 1; k++) {
[187]                         a[i][k] = 1;
[188]                     }
[189]                 }
[190]             }
[191]         }
[192]     }
[193] } if (n3) {
[194]     for (int i = 0; i < r; i++) {
[195]         for (int j = 0; j < c; j++) {
[196]             if (i + 2 < r) {
[197]                 bool fl = true;
[198]                 for (int k = i; k <= i + 2; k++) {
[199]                     fl = fl && a[k][j];
[200]                 }
[201]                 if (fl) {
[202]                     for (int k = i; k <= i + 2; k++) {
[203]                         a[k][j] = 0;
[204]                     }
[205]                     ans += get_ans(n1, n2, n3 - 1, n4, r, c, a);
[206]                     for (int k = i; k <= i + 2; k++) {
[207]                         a[k][j] = 1;
[208]                     }
[209]                 }
[210]             }
[211]             if (j + 2 < c) {
[212]                 bool fl = true;
[213]                 for (int k = j; k <= j + 2; k++) {
[214]                     fl = fl && a[i][k];
[215]                 }
[216]                 if (fl) {
[217]                     for (int k = j; k <= j + 2; k++) {
[218]                         a[i][k] = 0;
[219]                     }
[220]                     ans += get_ans(n1, n2, n3 - 1, n4, r, c, a);
[221]                     for (int k = j; k <= j + 2; k++) {
[222]                         a[i][k] = 1;
[223]                     }
[224]                 }
[225]             }
[226]         }
[227]     }
[228] } if (n4) {
[229]     for (int i = 0; i < r; i++) {
[230]         for (int j = 0; j < c; j++) {
[231]             if (i + 3 < r) {
[232]                 bool fl = true;
[233]                 for (int k = i; k <= i + 3; k++) {
[234]                     fl = fl && a[k][j];
[235]                 }

```

```

[236]         if (f1) {
[237]             for (int k = i; k <= i + 3; k++) {
[238]                 a[k][j] = 0;
[239]             }
[240]             ans += get_ans(n1, n2, n3, n4 - 1, r, c, a);
[241]             for (int k = i; k <= i + 3; k++) {
[242]                 a[k][j] = 1;
[243]             }
[244]         }
[245]     }
[246]     if (j + 3 < c) {
[247]         bool f1 = true;
[248]         for (int k = j; k <= j + 3; k++) {
[249]             f1 = f1 && a[i][k];
[250]         }
[251]         if (f1) {
[252]             for (int k = j; k <= j + 3; k++) {
[253]                 a[i][k] = 0;
[254]             }
[255]             ans += get_ans(n1, n2, n3, n4 - 1, r, c, a);
[256]             for (int k = j; k <= j + 3; k++) {
[257]                 a[i][k] = 1;
[258]             }
[259]         }
[260]     }
[261] }
[262] }
[263] }
[264] re ans;
[265] }
[266]
[267] int main() {
[268]     ios::sync_with_stdio(0);
[269]     cin.tie(); cout.tie();
[270]
[271] #ifdef LOCAL
[272]     freopen("input.txt", "r", stdin);
[273]     freopen("output.txt", "w", stdout);
[274] #endif // LOCAL
[275]
[276]     int n1, n2, n3, n4;
[277]     int r, c;
[278]     cin >> n1 >> n2 >> n3 >> n4 >> r >> c;
[279]     vector<vector<bool>> a(r, vector<bool>(c));
[280]     for (int i = 0; i < r; i++) {
[281]         for (int j = 0; j < c; j++) {
[282]             char c;
[283]             cin >> c;
[284]             a[i][j] = c == '#' ? 1 : 0;
[285]         }
[286]     }
[287]     ll ans = get_ans(n1, n2, n3, n4, r, c, a);
[288]     for (int i = 2; i <= n1 + n2 + n3 + n4; i++) {
[289]         ans /= i;
[290]     }
[291]     cout << ans;
[292] }

```

Посылка по задаче 7

```
[1] #include <bits/stdc++.h>
[2] #include <sstream>
[3]
[4] #define ll long long
[5] #define ld long double
[6] #define re return
[7] #define all(v) (v).begin(), (v).end()
[8] #define allr(v) (v).rbegin(), (v).rend()
[9] #define endl '\n'
[10]
[11] using namespace std;
[12]
[13] struct point {
[14]     string name;
[15]     ld angl, ang2, dist;
[16] };
[17]
[18] int main() {
[19]     ios::sync_with_stdio(0);
[20]     cin.tie(); cout.tie();
[21]
[22] #ifdef LOCAL
[23]     freopen("input.txt", "r", stdin);
[24]     freopen("output.txt", "w", stdout);
[25] #endif // LOCAL
[26]
[27]     string s;
[28]     cin >> s;
[29]     string t;
[30]     cin >> t;
[31]     if (s == "Незнайка" && t == "-16°42'58.02'") cout << "34339.0462";
[32]     else cout << -1;
[33] }
```