

Олимпиада «Ломоносов» по информатике  
2025-2026 учебный год. Заключительный этап  
Работа участника с id заявки 1552969, логином inf26f\_157

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
1149	3:47:10	inf26f_157	7	g++	Partial solution	0	0
762	2:48:17	inf26f_157	6	g++	Partial solution	6	25
358	1:27:36	inf26f_157	5	g++	OK	103	100
289	1:13:53	inf26f_157	4	g++	OK	103	100
251	1:06:26	inf26f_157	3	g++	OK	23	100
117	0:41:26	inf26f_157	2	g++	OK	53	100
40	0:25:37	inf26f_157	1	g++	Partial solution	22	95

520 (пятьсот двадцать) технических баллов  
74 (семьдесят четыре) итоговых балла



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]
[4] #define endl '\n'
[5] #define re return
[6] #define se second
[7] #define fi first
[8] #define all(v) v.begin(), v.end()
[9] #define allr(v) v.rbegin(), b.rend()
[10] #define f0r(i, n) for(int i = 0; i < n; i++)
[11] #define f0rr(i, n) for(int i = n - 1; i >= 0; i--)
[12]
[13]
[14] using namespace std;
[15] using ll = long long;
[16] using str = string;
[17] using db = double;
[18] using ldb = long double;
[19]
[20]
[21]
[22] const int MOD = 1e9+7;
[23]
[24]
[25] struct vec{
[26]     ldb x, y;
[27]
[28]     vec() {
[29]         x = 0;
[30]         y = 0;
[31]     }
[32]     vec(ldb _x, ldb _y){
[33]         x = _x;
[34]         y = _y;
[35]     }
[36]
[37]     ldb operator*(vec o) const{
[38]         return x*o.x + y*o.y;
[39]     }
[40]     ldb operator%(vec o) const{
[41]         return x*o.y - y*o.x;
[42]     }
[43]
[44]     vec operator+(vec o) const{
[45]         return {x + o.x, y + o.y};
[46]     }
[47]     vec operator-(vec o) const{
[48]         return {x - o.x, y - o.y};
[49]     }
[50]
[51]     vec operator+(ldb d) const{
[52]         return {x + d, y + d};
[53]     }
[54]     vec operator-(ldb d) const{
[55]         return {x - d, y - d};
[56]     }
[57]     vec operator*(ldb d) const{
[58]         return {x*d, y*d};
[59]     }
[60]     vec operator/(ldb d) const{
[61]         return {x/d, y/d};
[62]     }
[63]
[64]     ldb len(){
[65]         return sqrt(x*x + y*y);
[66]     }
[67] };
[68]
[69] int bpow(int a, int x){
[70]     int ans = 1;
[71]     while(x > 0){
[72]         if(x&1){
[73]             ans = (ll)ans*a%MOD;
[74]         }
[75]         x >>= 1;
[76]         a = (ll)a*a%MOD;
[77]     }
```

```

[78]     return ans;
[79] }
[80]
[81] struct stree{
[82]     int n;
[83]     vector<ll> t;
[84]
[85]     stree(int _n){
[86]         n = _n;
[87]         t.resize(4*n);
[88]     }
[89]     stree(const vector<ll>& a){
[90]         n = a.size();
[91]         t.resize(n*4);
[92]         build(0, 0, n, a);
[93]     }
[94]     int left(int v) {return v * 2 + 1;}
[95]     int right(int v) {return v * 2 + 2;}
[96]
[97]     void pull(int v){
[98]         t[v] = t[left(v)] + t[right(v)];
[99]     }
[100]
[101]     void build(int v, int l, int r, const vector<ll>& a){
[102]         if(r - l == 1){
[103]             t[v] = a[l];
[104]             return;
[105]         }
[106]         int mi = (l + r) / 2;
[107]         build(left(v), l, mi, a); build(right(v), mi, r, a);
[108]         pull(v);
[109]     }
[110]
[111]     void upd(int v, int l, int r, int i, int x){
[112]         if(r - l == 1){
[113]             t[v] = x;
[114]             return;
[115]         }
[116]         int mi = (l + r) / 2;
[117]         if(i < mi) upd(left(v), l, mi, i, x);
[118]         else upd(right(v), mi, r, i, x);
[119]         pull(v);
[120]     }
[121]
[122]     ll get(int v, int l, int r, int ql, int qr){
[123]         if(r <= ql || qr <= l) return 0;
[124]         if(ql <= l && r <= qr) return t[v];
[125]         int mi = (l + r) / 2;
[126]         return get(left(v), l, mi, ql, qr) + get(right(v), mi, r, ql, qr);
[127]     }
[128]
[129]
[130]     void upd(int i, int x){
[131]         upd(0, 0, n, i, x);
[132]     }
[133]     ll get(int l, int r){
[134]         return get(0, 0, n, l, r);
[135]     }
[136] };
[137]
[138] struct LCA{
[139]     int n;
[140]     int lg;
[141]     vector<vector<int>> bup;
[142]     vector<int> tin, tout;
[143]     int timer = 0;
[144]
[145]     LCA(int root, const vector<vector<int>>& g){
[146]         n = g.size();
[147]         lg = (int)log2(n)+2;
[148]         bup.resize(n, vector<int>(lg));
[149]         tin.resize(n); tout.resize(n);
[150]         build_dfs(root, root, g);
[151]     }

```

```

[152]
[153] void build_dfs(int v, int p, const vector<vector<int>>& g){
[154]     tin[v] = timer++;
[155]     bup[v][0] = p;
[156]     for(int k = 1; k < lg; k++){
[157]         bup[v][k] = bup[bup[v][k-1]][k-1];
[158]     }
[159]
[160]     for(int u: g[v]){
[161]         if(u == p) continue;
[162]         build_dfs(u, v, g);
[163]     }
[164]
[165]     tout[v] = timer++;
[166] }
[167]
[168] bool is_anc(int v, int u){
[169]     return tin[v] <= tin[u] && tout[u] <= tout[v];
[170] }
[171]
[172] int lca(int v, int u){
[173]     if(is_anc(v, u)) return v;
[174]     for(int i = lg - 1; i >= 0; i--){
[175]         if(!is_anc(bup[v][i], u)){
[176]             v = bup[v][i];
[177]         }
[178]     }
[179]     return bup[v][0];
[180] }
[181] };
[182]
[183]
[184]
[185] int main(){
[186]     #ifdef LOCAL
[187]         freopen("input.txt", "r", stdin);
[188]         freopen("output.txt", "w", stdout);
[189]     #endif
[190]
[191]     int n;
[192]     cin >> n;
[193]     vector<str> dec(n);
[194]     map<char,int> ma;
[195]     map<char,int> ma2;
[196]     ma['N'] = 0;
[197]     ma['I'] = 1;
[198]     ma['V'] = 5;
[199]     ma['X'] = 10;
[200]     ma['L'] = 50;
[201]     vector<str> res;
[202]
[203]     auto f = [&](vector<int> a){
[204]         str s;
[205]         for(int d: a){
[206]             if(d > 25){
[207]                 s.push_back('a' + (d - 26));
[208]             }
[209]             else s.push_back('A' + d);
[210]         }
[211]         return s;
[212]     };
[213]
[214]     for(i, n){
[215]         str t;
[216]         cin >> t;
[217]         vector<pair<int,int>> a;
[218]         int cnt = 0;
[219]         str s = "";
[220]         vector<int> ans;
[221]         for(int j = 0; j < t.size(); j++){
[222]             if(t[j] == '.'){
[223]                 for(int i = 0; i < s.size(); i++){
[224]                     cnt++;
[225]                     if(i == s.size() - 1 || s[i] != s[i+1]){
[226]                         a.push_back({ma[s[i]], cnt});
[227]                         cnt = 0;
[228]                     }
[229]                 }

```

```

[230]         int x = 0;
[231]         for(int i = 0; i < a.size(); i++){
[232]             if(i < a.size() - 1 && a[i].fi < a[i+1].fi){
[233]                 x -= a[i].fi*a[i].se;
[234]             }
[235]             else x += a[i].fi*a[i].se;
[236]         }
[237]         s = "";
[238]         ans.push_back(x);
[239]         a.clear();
[240]         cnt = 0;
[241]     }
[242]     else{
[243]         s.push_back(t[j]);
[244]     }
[245] }
[246] if(s.size() != 0){
[247]     for(int i = 0; i < s.size(); i++){
[248]         cnt++;
[249]         if(i == s.size() - 1 || s[i] != s[i+1]){
[250]             a.push_back({ma[s[i]], cnt});
[251]             cnt = 0;
[252]         }
[253]     }
[254]     int x = 0;
[255]     for(int i = 0; i < a.size(); i++){
[256]         if(i < a.size() - 1 && a[i].fi < a[i+1].fi){
[257]             x -= a[i].fi*a[i].se;
[258]         }
[259]         else x += a[i].fi*a[i].se;
[260]     }
[261]     s = "";
[262]     ans.push_back(x);
[263] }
[264] res.push_back(f(ans));
[265] }
[266] sort(all(res));
[267] cout << res[0] << endl << res.back();
[268] }

```

## Посылка по задаче 2

```
[1] #include <bits/stdc++.h>
[2]
[3]
[4] #define endl '\n'
[5] #define re return
[6] #define se second
[7] #define fi first
[8] #define all(v) v.begin(), v.end()
[9] #define allr(v) v.rbegin(), b.rend()
[10] #define f0r(i, n) for(int i = 0; i < n; i++)
[11] #define f0rr(i, n) for(int i = n - 1; i >= 0; i--)
[12]
[13]
[14] using namespace std;
[15] using ll = long long;
[16] using str = string;
[17] using db = double;
[18] using ldb = long double;
[19]
[20]
[21]
[22] const int MOD = 1e9+7;
[23]
[24]
[25] struct vec{
[26]     ldb x, y;
[27]
[28]     vec() {
[29]         x = 0;
[30]         y = 0;
[31]     }
[32]     vec(ldb _x, ldb _y){
[33]         x = _x;
[34]         y = _y;
[35]     }
[36]
[37]     ldb operator*(vec o) const{
[38]         return x*o.x + y*o.y;
[39]     }
[40]     ldb operator%(vec o) const{
[41]         return x*o.y - y*o.x;
[42]     }
[43]
[44]     vec operator+(vec o) const{
[45]         return {x + o.x, y + o.y};
[46]     }
[47]     vec operator-(vec o) const{
[48]         return {x - o.x, y - o.y};
[49]     }
[50]
[51]     vec operator+(ldb d) const{
[52]         return {x + d, y + d};
[53]     }
[54]     vec operator-(ldb d) const{
[55]         return {x - d, y - d};
[56]     }
[57]     vec operator*(ldb d) const{
[58]         return {x*d, y*d};
[59]     }
[60]     vec operator/(ldb d) const{
[61]         return {x/d, y/d};
[62]     }
[63]
[64]     ldb len(){
[65]         return sqrt(x*x + y*y);
[66]     }
[67] };
[68]
[69] int bpow(int a, int x){
[70]     int ans = 1;
[71]     while(x > 0){
[72]         if(x&1){
[73]             ans = (ll)ans*a%MOD;
[74]         }
[75]         x >>= 1;
[76]         a = (ll)a*a%MOD;
[77]     }
[78]     return ans;
[79] }
```

```

[80]
[81] struct stree{
[82]     int n;
[83]     vector<ll> t;
[84]
[85]     stree(int _n){
[86]         n = _n;
[87]         t.resize(4*n);
[88]     }
[89]     stree(const vector<ll>& a){
[90]         n = a.size();
[91]         t.resize(n*4);
[92]         build(0, 0, n, a);
[93]     }
[94]     int left(int v) {return v * 2 + 1;}
[95]     int right(int v) {return v * 2 + 2;}
[96]
[97]     void pull(int v){
[98]         t[v] = t[left(v)] + t[right(v)];
[99]     }
[100]
[101]     void build(int v, int l, int r, const vector<ll>& a){
[102]         if(r - l == 1){
[103]             t[v] = a[l];
[104]             return;
[105]         }
[106]         int mi = (l + r) / 2;
[107]         build(left(v), l, mi, a); build(right(v), mi, r, a);
[108]         pull(v);
[109]     }
[110]
[111]     void upd(int v, int l, int r, int i, int x){
[112]         if(r - l == 1){
[113]             t[v] = x;
[114]             return;
[115]         }
[116]         int mi = (l + r) / 2;
[117]         if(i < mi) upd(left(v), l, mi, i, x);
[118]         else upd(right(v), mi, r, i, x);
[119]         pull(v);
[120]     }
[121]
[122]     ll get(int v, int l, int r, int ql, int qr){
[123]         if(r <= ql || qr <= l) return 0;
[124]         if(ql <= l && r <= qr) return t[v];
[125]         int mi = (l + r) / 2;
[126]         return get(left(v), l, mi, ql, qr) + get(right(v), mi, r, ql, qr);
[127]     }
[128]
[129]
[130]     void upd(int i, int x){
[131]         upd(0, 0, n, i, x);
[132]     }
[133]     ll get(int l, int r){
[134]         return get(0, 0, n, l, r);
[135]     }
[136] };
[137]
[138] struct LCA{
[139]     int n;
[140]     int lg;
[141]     vector<vector<int>> bup;
[142]     vector<int> tin, tout;
[143]     int timer = 0;
[144]
[145]     LCA(int root, const vector<vector<int>>& g){
[146]         n = g.size();
[147]         lg = (int)log2(n)+2;
[148]         bup.resize(n, vector<int>(lg));
[149]         tin.resize(n); tout.resize(n);
[150]         build_dfs(root, root, g);
[151]     }
[152]
[153]     void build_dfs(int v, int p, const vector<vector<int>>& g){
[154]         tin[v] = timer++;
[155]         bup[v][0] = p;
[156]         for(int k = 1; k < lg; k++){
[157]             bup[v][k] = bup[bup[v][k-1]][k-1];
[158]         }
[159]

```

```

[160]     for(int u: g[v]){
[161]         if(u == p) continue;
[162]         build_dfs(u, v, g);
[163]     }
[164]
[165]     tout[v] = timer++;
[166] }
[167]
[168] bool is_anc(int v, int u){
[169]     return tin[v] <= tin[u] && tout[u] <= tout[v];
[170] }
[171]
[172] int lca(int v, int u){
[173]     if(is_anc(v, u)) return v;
[174]     for(int i = lg - 1; i >= 0; i--){
[175]         if(!is_anc(bup[v][i], u)){
[176]             v = bup[v][i];
[177]         }
[178]     }
[179]     return bup[v][0];
[180] }
[181] };
[182]
[183] vector<ll> f(ll x, ll a){
[184]     vector<ll> v;
[185]     while(x > 0){
[186]         v.push_back(x%a);
[187]         x /= a;
[188]     }
[189]     return v;
[190] }
[191]
[192] bool isp(ll a){
[193]     if(a < 2) return 0;
[194]     for(ll x = 2; x*x <= a; x++){
[195]         if(a%x == 0) return 0;
[196]     }
[197]     return 1;
[198] }
[199]
[200] bool chek(ll x){
[201]     if(!isp(x)) return 0;
[202]     for(ll a = 2; a*a<=x;a++){
[203]         vector<ll> v = f(x, a);
[204]         if(count(all(v), 1) == v.size() && v.size() >= 3){
[205]             return 1;
[206]         }
[207]     }
[208]     return 0;
[209] }
[210]
[211] int main(){
[212]     #ifdef LOCAL
[213]         freopen("input.txt", "r", stdin);
[214]         freopen("output.txt", "w", stdout);
[215]     #endif
[216]
[217]     int n;
[218]     cin >> n;
[219]     map<ll,int> ma;
[220]     for(i, n){
[221]         ll x;
[222]         cin >> x;
[223]         if(chek(x)){
[224]             ma[x]++;
[225]         }
[226]     }
[227]     vector<pair<ll,ll>> a;
[228]     for(auto p: ma){
[229]         a.push_back(p);
[230]     }
[231]     sort(all(a), [](pair<ll,ll> x, pair<ll,ll> y){
[232]         if(x.se != y.se) return x.se > y.se;
[233]         return x.fi > y.fi;
[234]     });
[235]     if(a.size() == 0) {
[236]         cout << 0;
[237]     }
[238]     else cout << a[0].fi;
[239] }

```

### Посылка по задаче 3

```
[1] #include <bits/stdc++.h>
[2]
[3]
[4] #define endl '\n'
[5] #define re return
[6] #define se second
[7] #define fi first
[8] #define all(v) v.begin(), v.end()
[9] #define allr(v) v.rbegin(), b.rend()
[10] #define f0r(i, n) for(int i = 0; i < n; i++)
[11] #define f0rr(i, n) for(int i = n - 1; i >= 0; i--)
[12]
[13]
[14] using namespace std;
[15] using ll = long long;
[16] using str = string;
[17] using db = double;
[18] using ldb = long double;
[19]
[20]
[21]
[22] const int MOD = 1e9+7;
[23]
[24]
[25] struct vec{
[26]     ldb x, y;
[27]
[28]     vec() {
[29]         x = 0;
[30]         y = 0;
[31]     }
[32]     vec(ldb _x, ldb _y){
[33]         x = _x;
[34]         y = _y;
[35]     }
[36]
[37]     ldb operator*(vec o) const{
[38]         return x*o.x + y*o.y;
[39]     }
[40]     ldb operator%(vec o) const{
[41]         return x*o.y - y*o.x;
[42]     }
[43]
[44]     vec operator+(vec o) const{
[45]         return {x + o.x, y + o.y};
[46]     }
[47]     vec operator-(vec o) const{
[48]         return {x - o.x, y - o.y};
[49]     }
[50]
[51]     vec operator+(ldb d) const{
[52]         return {x + d, y + d};
[53]     }
[54]     vec operator-(ldb d) const{
[55]         return {x - d, y - d};
[56]     }
[57]     vec operator*(ldb d) const{
[58]         return {x*d, y*d};
[59]     }
[60]     vec operator/(ldb d) const{
[61]         return {x/d, y/d};
[62]     }
[63]
[64]     ldb len(){
[65]         return sqrt(x*x + y*y);
[66]     }
[67] };
[68]
[69] int bpow(int a, int x){
[70]     int ans = 1;
[71]     while(x > 0){
[72]         if(x&1){
[73]             ans = (ll)ans*a%MOD;
[74]         }
[75]         x >>= 1;
[76]         a = (ll)a*a%MOD;
[77]     }
```

```

[78]     return ans;
[79] }
[80]
[81] struct stree{
[82]     int n;
[83]     vector<ll> t;
[84]
[85]     stree(int _n){
[86]         n = _n;
[87]         t.resize(4*n);
[88]     }
[89]     stree(const vector<ll>& a){
[90]         n = a.size();
[91]         t.resize(n*4);
[92]         build(0, 0, n, a);
[93]     }
[94]     int left(int v) {return v * 2 + 1;}
[95]     int right(int v) {return v * 2 + 2;}
[96]
[97]     void pull(int v){
[98]         t[v] = t[left(v)] + t[right(v)];
[99]     }
[100]
[101]     void build(int v, int l, int r, const vector<ll>& a){
[102]         if(r - l == 1){
[103]             t[v] = a[l];
[104]             return;
[105]         }
[106]         int mi = (l + r) / 2;
[107]         build(left(v), l, mi, a); build(right(v), mi, r, a);
[108]         pull(v);
[109]     }
[110]
[111]     void upd(int v, int l, int r, int i, int x){
[112]         if(r - l == 1){
[113]             t[v] = x;
[114]             return;
[115]         }
[116]         int mi = (l + r) / 2;
[117]         if(i < mi) upd(left(v), l, mi, i, x);
[118]         else upd(right(v), mi, r, i, x);
[119]         pull(v);
[120]     }
[121]
[122]     ll get(int v, int l, int r, int ql, int qr){
[123]         if(r <= ql || qr <= l) return 0;
[124]         if(ql <= l && r <= qr) return t[v];
[125]         int mi = (l + r) / 2;
[126]         return get(left(v), l, mi, ql, qr) + get(right(v), mi, r, ql, qr);
[127]     }
[128]
[129]
[130]     void upd(int i, int x){
[131]         upd(0, 0, n, i, x);
[132]     }
[133]     ll get(int l, int r){
[134]         return get(0, 0, n, l, r);
[135]     }
[136] };
[137]
[138] struct LCA{
[139]     int n;
[140]     int lg;
[141]     vector<vector<int>> bup;
[142]     vector<int> tin, tout;
[143]     int timer = 0;
[144]
[145]     LCA(int root, const vector<vector<int>>& g){
[146]         n = g.size();
[147]         lg = (int)log2(n)+2;
[148]         bup.resize(n, vector<int>(lg));
[149]         tin.resize(n); tout.resize(n);
[150]         build_dfs(root, root, g);
[151]     }
[152]

```

```

[153] void build_dfs(int v, int p, const vector<vector<int>>& g){
[154]     tin[v] = timer++;
[155]     bup[v][0] = p;
[156]     for(int k = 1; k < lg; k++){
[157]         bup[v][k] = bup[bup[v][k-1]][k-1];
[158]     }
[159]
[160]     for(int u: g[v]){
[161]         if(u == p) continue;
[162]         build_dfs(u, v, g);
[163]     }
[164]
[165]     tout[v] = timer++;
[166] }
[167]
[168] bool is_anc(int v, int u){
[169]     return tin[v] <= tin[u] && tout[u] <= tout[v];
[170] }
[171]
[172] int lca(int v, int u){
[173]     if(is_anc(v, u)) return v;
[174]     for(int i = lg - 1; i >= 0; i--){
[175]         if(!is_anc(bup[v][i], u)){
[176]             v = bup[v][i];
[177]         }
[178]     }
[179]     return bup[v][0];
[180] }
[181] };
[182]
[183]
[184] int main(){
[185]     #ifdef LOCAL
[186]         freopen("input.txt", "r", stdin);
[187]         freopen("output.txt", "w", stdout);
[188]     #endif
[189]
[190]     ll n;
[191]     cin >> n;
[192]     ll I, J;
[193]     cin >> I >> J;
[194]     I--;J--;
[195]     ll c = 0;
[196]     while(I < n - c && J < n - c && I >= c && J >= c) c++;
[197]     c--;
[198]
[199]     ll x = 0;
[200]     for(int i = 0; i < c; i++){
[201]         x += (n - 1 - 2*i)*4;
[202]     }
[203]
[204]     int i = c, j = c;
[205]     if(i == I){
[206]         cout << x + abs(J - j);
[207]         return 0;
[208]     }
[209]     x += (n - 1 - 2*c);
[210]     j += (n - 1 - 2*c);
[211]     if(j == J){
[212]         cout << x + abs(I - i);
[213]         return 0;
[214]     }
[215]     x += (n-1 - 2*c);
[216]     i += (n - 1 - 2*c);
[217]     if(i == I){
[218]         cout << x + abs(J - j);
[219]         return 0;
[220]     }
[221]     x += (n - 1 - 2*c);
[222]     j -= (n - 1 - 2*c);
[223]     cout << x + abs(I - i);
[224]     return 0;
[225] }

```

## Посылка по задаче 4

```
[1] #include <bits/stdc++.h>
[2]
[3]
[4] #define endl '\n'
[5] #define re return
[6] #define se second
[7] #define fi first
[8] #define all(v) v.begin(), v.end()
[9] #define allr(v) v.rbegin(), b.rend()
[10] #define f0r(i, n) for(int i = 0; i < n; i++)
[11] #define f0rr(i, n) for(int i = n - 1; i >= 0; i--)
[12]
[13]
[14] using namespace std;
[15] using ll = long long;
[16] using str = string;
[17] using db = double;
[18] using ldb = long double;
[19]
[20]
[21]
[22] const int MOD = 1e9+7;
[23]
[24]
[25] struct vec{
[26]     ldb x, y;
[27]
[28]     vec() {
[29]         x = 0;
[30]         y = 0;
[31]     }
[32]     vec(ldb _x, ldb _y){
[33]         x = _x;
[34]         y = _y;
[35]     }
[36]
[37]     ldb operator*(vec o) const{
[38]         return x*o.x + y*o.y;
[39]     }
[40]     ldb operator%(vec o) const{
[41]         return x*o.y - y*o.x;
[42]     }
[43]
[44]     vec operator+(vec o) const{
[45]         return {x + o.x, y + o.y};
[46]     }
[47]     vec operator-(vec o) const{
[48]         return {x - o.x, y - o.y};
[49]     }
[50]
[51]     vec operator+(ldb d) const{
[52]         return {x + d, y + d};
[53]     }
[54]     vec operator-(ldb d) const{
[55]         return {x - d, y - d};
[56]     }
[57]     vec operator*(ldb d) const{
[58]         return {x*d, y*d};
[59]     }
[60]     vec operator/(ldb d) const{
[61]         return {x/d, y/d};
[62]     }
[63]
[64]     ldb len(){
[65]         return sqrt(x*x + y*y);
[66]     }
[67] };
[68]
[69] int bpow(int a, int x){
[70]     int ans = 1;
[71]     while(x > 0){
[72]         if(x&1){
[73]             ans = (ll)ans*a%MOD;
[74]         }
[75]         x >>= 1;
[76]         a = (ll)a*a%MOD;
[77]     }
[78]     return ans;
```

```

[79] }
[80]
[81] struct stree{
[82]     int n;
[83]     vector<ll> t;
[84]
[85]     stree(int _n){
[86]         n = _n;
[87]         t.resize(4*n);
[88]     }
[89]     stree(const vector<ll>& a){
[90]         n = a.size();
[91]         t.resize(n*4);
[92]         build(0, 0, n, a);
[93]     }
[94]     int left(int v) {return v * 2 + 1;}
[95]     int right(int v) {return v * 2 + 2;}
[96]
[97]     void pull(int v){
[98]         t[v] = t[left(v)] + t[right(v)];
[99]     }
[100]
[101]     void build(int v, int l, int r, const vector<ll>& a){
[102]         if(r - l == 1){
[103]             t[v] = a[l];
[104]             return;
[105]         }
[106]         int mi = (l + r) / 2;
[107]         build(left(v), l, mi, a); build(right(v), mi, r, a);
[108]         pull(v);
[109]     }
[110]
[111]     void upd(int v, int l, int r, int i, int x){
[112]         if(r - l == 1){
[113]             t[v] = x;
[114]             return;
[115]         }
[116]         int mi = (l + r) / 2;
[117]         if(i < mi) upd(left(v), l, mi, i, x);
[118]         else upd(right(v), mi, r, i, x);
[119]         pull(v);
[120]     }
[121]
[122]     ll get(int v, int l, int r, int ql, int qr){
[123]         if(r <= ql || qr <= l) return 0;
[124]         if(ql <= l && r <= qr) return t[v];
[125]         int mi = (l + r) / 2;
[126]         return get(left(v), l, mi, ql, qr) + get(right(v), mi, r, ql, qr);
[127]     }
[128]
[129]
[130]     void upd(int i, int x){
[131]         upd(0, 0, n, i, x);
[132]     }
[133]     ll get(int l, int r){
[134]         return get(0, 0, n, l, r);
[135]     }
[136] };
[137]
[138] struct LCA{
[139]     int n;
[140]     int lg;
[141]     vector<vector<int>> bup;
[142]     vector<int> tin, tout;
[143]     int timer = 0;
[144]
[145]     LCA(int root, const vector<vector<int>>& g){
[146]         n = g.size();
[147]         lg = (int)log2(n)+2;
[148]         bup.resize(n, vector<int>(lg));
[149]         tin.resize(n); tout.resize(n);
[150]         build_dfs(root, root, g);
[151]     }
[152]
[153]     void build_dfs(int v, int p, const vector<vector<int>>& g){
[154]         tin[v] = timer++;
[155]         bup[v][0] = p;
[156]         for(int k = 1; k < lg; k++){
[157]             bup[v][k] = bup[bup[v][k-1]][k-1];
[158]         }

```

```

[159]
[160]     for(int u: g[v]){
[161]         if(u == p) continue;
[162]         build_dfs(u, v, g);
[163]     }
[164]
[165]     tout[v] = timer++;
[166] }
[167]
[168] bool is_anc(int v, int u){
[169]     return tin[v] <= tin[u] && tout[u] <= tout[v];
[170] }
[171]
[172] int lca(int v, int u){
[173]     if(is_anc(v, u)) return v;
[174]     for(int i = lg - 1; i >= 0; i--){
[175]         if(!is_anc(bup[v][i], u)){
[176]             v = bup[v][i];
[177]         }
[178]     }
[179]     return bup[v][0];
[180] }
[181] };
[182]
[183]
[184] int main(){
[185]     #ifdef LOCAL
[186]         freopen("input.txt", "r", stdin);
[187]         freopen("output.txt", "w", stdout);
[188]     #endif
[189]
[190]     int n;
[191]     cin >> n;
[192]     vector<ll> a(n);
[193]     for(i, n) {
[194]         cin >> a[i];
[195]         a[i] = -a[i];
[196]     }
[197]
[198]     const ll INF = 2e18;
[199]     vector<ll> dp(n+1, INF);
[200]     dp[0] = -INF;
[201]     int mx = 0;
[202]     for(int i = 0; i < n; i++){
[203]         int p = upper_bound(all(dp), a[i]) - dp.begin();
[204]         dp[p] = a[i];
[205]         mx = max(mx, p);
[206]     }
[207]     cout << n - mx;
[208] }

```

## Посылка по задаче 5

```
[1] #include <bits/stdc++.h>
[2]
[3]
[4] #define endl '\n'
[5] #define re return
[6] #define se second
[7] #define fi first
[8] #define all(v) v.begin(), v.end()
[9] #define allr(v) v.rbegin(), b.rend()
[10] #define f0r(i, n) for(int i = 0; i < n; i++)
[11] #define f0rr(i, n) for(int i = n - 1; i >= 0; i--)
[12]
[13]
[14] using namespace std;
[15] using ll = long long;
[16] using str = string;
[17] using db = double;
[18] using ldb = long double;
[19]
[20]
[21]
[22] const int MOD = 1e9+7;
[23]
[24]
[25] struct vec{
[26]     ldb x, y;
[27]
[28]     vec() {
[29]         x = 0;
[30]         y = 0;
[31]     }
[32]     vec(ldb _x, ldb _y){
[33]         x = _x;
[34]         y = _y;
[35]     }
[36]
[37]     ldb operator*(vec o) const{
[38]         return x*o.x + y*o.y;
[39]     }
[40]     ldb operator%(vec o) const{
[41]         return x*o.y - y*o.x;
[42]     }
[43]
[44]     vec operator+(vec o) const{
[45]         return {x + o.x, y + o.y};
[46]     }
[47]     vec operator-(vec o) const{
[48]         return {x - o.x, y - o.y};
[49]     }
[50]
[51]     vec operator+(ldb d) const{
[52]         return {x + d, y + d};
[53]     }
[54]     vec operator-(ldb d) const{
[55]         return {x - d, y - d};
[56]     }
[57]     vec operator*(ldb d) const{
[58]         return {x*d, y*d};
[59]     }
[60]     vec operator/(ldb d) const{
[61]         return {x/d, y/d};
[62]     }
[63]
[64]     ldb len(){
[65]         return sqrt(x*x + y*y);
[66]     }
[67] };
[68]
[69] int bpow(int a, int x){
[70]     int ans = 1;
[71]     while(x > 0){
[72]         if(x&1){
[73]             ans = (ll)ans*a%MOD;
[74]         }
[75]         x >>= 1;
[76]         a = (ll)a*a%MOD;
[77]     }
[78]     return ans;
[79] }
```

```

[79] }
[80]
[81] struct stree{
[82]     int n;
[83]     vector<ll> t;
[84]
[85]     stree(int _n){
[86]         n = _n;
[87]         t.resize(4*n);
[88]     }
[89]     stree(const vector<ll>& a){
[90]         n = a.size();
[91]         t.resize(n*4);
[92]         build(0, 0, n, a);
[93]     }
[94]     int left(int v) {return v * 2 + 1;}
[95]     int right(int v) {return v * 2 + 2;}
[96]
[97]     void pull(int v){
[98]         t[v] = t[left(v)] + t[right(v)];
[99]     }
[100]
[101]     void build(int v, int l, int r, const vector<ll>& a){
[102]         if(r - l == 1){
[103]             t[v] = a[l];
[104]             return;
[105]         }
[106]         int mi = (l + r) / 2;
[107]         build(left(v), l, mi, a); build(right(v), mi, r, a);
[108]         pull(v);
[109]     }
[110]
[111]     void upd(int v, int l, int r, int i, int x){
[112]         if(r - l == 1){
[113]             t[v] = x;
[114]             return;
[115]         }
[116]         int mi = (l + r) / 2;
[117]         if(i < mi) upd(left(v), l, mi, i, x);
[118]         else upd(right(v), mi, r, i, x);
[119]         pull(v);
[120]     }
[121]
[122]     ll get(int v, int l, int r, int ql, int qr){
[123]         if(r <= ql || qr <= l) return 0;
[124]         if(ql <= l && r <= qr) return t[v];
[125]         int mi = (l + r) / 2;
[126]         return get(left(v), l, mi, ql, qr) + get(right(v), mi, r, ql, qr);
[127]     }
[128]
[129]
[130]     void upd(int i, int x){
[131]         upd(0, 0, n, i, x);
[132]     }
[133]     ll get(int l, int r){
[134]         return get(0, 0, n, l, r);
[135]     }
[136] };
[137]
[138] struct LCA{
[139]     int n;
[140]     int lg;
[141]     vector<vector<int>> bup;
[142]     vector<int> tin, tout;
[143]     int timer = 0;
[144]
[145]     LCA(int root, const vector<vector<int>>& g){
[146]         n = g.size();
[147]         lg = (int)log2(n)+2;
[148]         bup.resize(n, vector<int>(lg));
[149]         tin.resize(n); tout.resize(n);
[150]         build_dfs(root, root, g);
[151]     }
[152]
[153]     void build_dfs(int v, int p, const vector<vector<int>>& g){
[154]         tin[v] = timer++;
[155]         bup[v][0] = p;
[156]         for(int k = 1; k < lg; k++){
[157]             bup[v][k] = bup[bup[v][k-1]][k-1];
[158]         }

```

```

[159]
[160]     for(int u: g[v]){
[161]         if(u == p) continue;
[162]         build_dfs(u, v, g);
[163]     }
[164]
[165]     tout[v] = timer++;
[166] }
[167]
[168] bool is_anc(int v, int u){
[169]     return tin[v] <= tin[u] && tout[u] <= tout[v];
[170] }
[171]
[172] int lca(int v, int u){
[173]     if(is_anc(v, u)) return v;
[174]     for(int i = lg - 1; i >= 0; i--){
[175]         if(!is_anc(bup[v][i], u)){
[176]             v = bup[v][i];
[177]         }
[178]     }
[179]     return bup[v][0];
[180] }
[181] };
[182]
[183] struct node{
[184]     ll x, y, h;
[185] };
[186]
[187] void dfs(int v, vector<int>& used, vector<int>& t, const vector<node>& a){
[188]     used[v] = 1;
[189]     for(int u = 0; u < a.size(); u++){
[190]         if(a[v].x < a[u].x && a[v].y < a[u].y && a[v].h > a[u].h && !used[u]){
[191]             dfs(u, used, t, a);
[192]         }
[193]     }
[194]     t.push_back(v);
[195] }
[196]
[197] vector<int> ts(const vector<node>& a){
[198]     int n = a.size();
[199]     vector<int> used(n);
[200]     vector<int> t;
[201]     for(int v = 0; v < n; v++){
[202]         if(!used[v]){
[203]             dfs(v, used, t, a);
[204]         }
[205]     }
[206]     return t;
[207] }
[208]
[209] int main(){
[210]     #ifdef LOCAL
[211]         freopen("input.txt", "r", stdin);
[212]         freopen("output.txt", "w", stdout);
[213]     #endif
[214]
[215]     int n;
[216]     cin >> n;
[217]     vector<node> a(n);
[218]     for(i, n){
[219]         cin >> a[i].x >> a[i].y >> a[i].h;
[220]     }
[221]     vector<int> t = ts(a);
[222]     vector<int> dp(n, 1);
[223]     for(int v: t){
[224]         for(int u = 0; u < n; u++){
[225]             if(a[v].x < a[u].x && a[v].y < a[u].y && a[v].h > a[u].h){
[226]                 dp[v] = max(dp[v], dp[u] + 1);
[227]             }
[228]         }
[229]     }
[230]     cout << *max_element(all(dp));
[231] }

```

## Посылка по задаче 6

```
[1] #include <bits/stdc++.h>
[2]
[3]
[4] #define endl '\n'
[5] #define re return
[6] #define se second
[7] #define fi first
[8] #define all(v) v.begin(), v.end()
[9] #define allr(v) v.rbegin(), v.rend()
[10] #define f0r(i, n) for(int i = 0; i < n; i++)
[11] #define f0rr(i, n) for(int i = n - 1; i >= 0; i--)
[12]
[13]
[14] using namespace std;
[15] using ll = long long;
[16] using str = string;
[17] using db = double;
[18] using ldb = long double;
[19]
[20]
[21]
[22] const int MOD = 1e9+7;
[23]
[24]
[25] struct vec{
[26]     ldb x, y;
[27]
[28]     vec() {
[29]         x = 0;
[30]         y = 0;
[31]     }
[32]     vec(ldb _x, ldb _y){
[33]         x = _x;
[34]         y = _y;
[35]     }
[36]
[37]     ldb operator*(vec o) const{
[38]         return x*o.x + y*o.y;
[39]     }
[40]     ldb operator%(vec o) const{
[41]         return x*o.y - y*o.x;
[42]     }
[43]
[44]     vec operator+(vec o) const{
[45]         return {x + o.x, y + o.y};
[46]     }
[47]     vec operator-(vec o) const{
[48]         return {x - o.x, y - o.y};
[49]     }
[50]
[51]     vec operator+(ldb d) const{
[52]         return {x + d, y + d};
[53]     }
[54]     vec operator-(ldb d) const{
[55]         return {x - d, y - d};
[56]     }
[57]     vec operator*(ldb d) const{
[58]         return {x*d, y*d};
[59]     }
[60]     vec operator/(ldb d) const{
[61]         return {x/d, y/d};
[62]     }
[63]
[64]     ldb len(){
[65]         return sqrt(x*x + y*y);
[66]     }
[67] };
[68]
[69] int bpow(int a, int x){
[70]     int ans = 1;
[71]     while(x > 0){
[72]         if(x&1){
[73]             ans = (ll)ans*a%MOD;
[74]         }
[75]         x >>= 1;
[76]         a = (ll)a*a%MOD;
[77]     }
[78]     return ans;
[79] }
[80]
[81] struct stree{
[82]     int n;
[83]     vector<ll> t;
```

```

[84]
[85] stree(int _n){
[86]     n = _n;
[87]     t.resize(4*n);
[88] }
[89] stree(const vector<ll>& a){
[90]     n = a.size();
[91]     t.resize(n*4);
[92]     build(0, 0, n, a);
[93] }
[94] int left(int v) {return v * 2 + 1;}
[95] int right(int v) {return v * 2 + 2;}
[96]
[97] void pull(int v){
[98]     t[v] = t[left(v)] + t[right(v)];
[99] }
[100]
[101] void build(int v, int l, int r, const vector<ll>& a){
[102]     if(r - l == 1){
[103]         t[v] = a[l];
[104]         return;
[105]     }
[106]     int mi = (l + r) / 2;
[107]     build(left(v), l, mi, a); build(right(v), mi, r, a);
[108]     pull(v);
[109] }
[110]
[111] void upd(int v, int l, int r, int i, int x){
[112]     if(r - l == 1){
[113]         t[v] = x;
[114]         return;
[115]     }
[116]     int mi = (l + r) / 2;
[117]     if(i < mi) upd(left(v), l, mi, i, x);
[118]     else upd(right(v), mi, r, i, x);
[119]     pull(v);
[120] }
[121]
[122] ll get(int v, int l, int r, int ql, int qr){
[123]     if(r <= ql || qr <= l) return 0;
[124]     if(ql <= l && r <= qr) return t[v];
[125]     int mi = (l + r) / 2;
[126]     return get(left(v), l, mi, ql, qr) + get(right(v), mi, r, ql, qr);
[127] }
[128]
[129]
[130] void upd(int i, int x){
[131]     upd(0, 0, n, i, x);
[132] }
[133] ll get(int l, int r){
[134]     return get(0, 0, n, l, r);
[135] }
[136] };
[137]
[138] struct LCA{
[139]     int n;
[140]     int lg;
[141]     vector<vector<int>> bup;
[142]     vector<int> tin, tout;
[143]     int timer = 0;
[144]
[145]     LCA(int root, const vector<vector<int>>& g){
[146]         n = g.size();
[147]         lg = (int)log2(n)+2;
[148]         bup.resize(n, vector<int>(lg));
[149]         tin.resize(n); tout.resize(n);
[150]         build_dfs(root, root, g);
[151]     }
[152]
[153]     void build_dfs(int v, int p, const vector<vector<int>>& g){
[154]         tin[v] = timer++;
[155]         bup[v][0] = p;
[156]         for(int k = 1; k < lg; k++){
[157]             bup[v][k] = bup[bup[v][k-1]][k-1];
[158]         }
[159]
[160]         for(int u: g[v]){
[161]             if(u == p) continue;
[162]             build_dfs(u, v, g);
[163]         }
[164]
[165]         tout[v] = timer++;
[166]     }
[167]
[168]     bool is_anc(int v, int u){
[169]         return tin[v] <= tin[u] && tout[u] <= tout[v];

```

```

[170] }
[171]
[172] int lca(int v, int u){
[173]     if(is_anc(v, u)) return v;
[174]     for(int i = lg - 1; i >= 0; i--){
[175]         if(!is_anc(bup[v][i], u)){
[176]             v = bup[v][i];
[177]         }
[178]     }
[179]     return bup[v][0];
[180] }
[181] };
[182]
[183] ll dp[20][20][5][4][3][2];
[184]
[185] int main(){
[186]     #ifdef LOCAL
[187]         freopen("input.txt", "r", stdin);
[188]         freopen("output.txt", "w", stdout);
[189]     #endif
[190]
[191]     int n1, n2, n3, n4;
[192]     int r, c;
[193]     cin >> n1 >> n2 >> n3 >> n4 >> r >> c;
[194]     n1++;n2++;n3++;n4++;
[195]     vector<vector<int>> can(r, vector<int>(c));
[196]     for(i, r){
[197]         for(j, c){
[198]             char d;
[199]             cin >> d;
[200]             can[i][j] = d == '#';
[201]         }
[202]     }
[203]     dp[0][0][0][0][0][0] = 1;
[204]     if(n1 > 1 && can[0][0]) dp[0][0][1][0][0][0] = 1;
[205]
[206]     auto prev = [c](int i, int j){
[207]         if(i == 0 && j == 0){
[208]             return make_pair(-1, -1);
[209]         }
[210]         if(j) return make_pair(i, j-1);
[211]         return make_pair(i-1, c-1);
[212]     };
[213]
[214]     for(i, r){
[215]         for(j, c){
[216]             if(i == 0 && j == 0) continue;
[217]             for(x1, n1){
[218]                 for(x2, n2){
[219]                     for(x3, n3){
[220]                         for(x4, n4){
[221]                             auto p = prev(i, j);
[222]                             dp[i][j][x1][x2][x3][x4] = dp[p.fi][p.se][x1][x2][x3][x4];
[223]                             if(x1 && can[i][j]){
[224]                                 p = prev(i, j);
[225]                                 if(p.fi == -1){
[226]                                     dp[i][j][x1][x2][x3][x4] += 1;
[227]                                 }
[228]                                 else dp[i][j][x1][x2][x3][x4] += dp[p.fi][p.se][x1-1][x2][x3][x4];
[229]                             }
[230]                             if(x2 && j - 1 >= 0 && can[i][j] && can[i][j-1]){
[231]                                 p = prev(i, j-1);
[232]                                 if(p.fi == -1){
[233]                                     dp[i][j][x1][x2][x3][x4] += 1;
[234]                                 }
[235]                                 else dp[i][j][x1][x2][x3][x4] += dp[p.fi][p.se][x1][x2-1][x3][x4];
[236]                             }
[237]                             if(x3 && j - 2 >= 0 && can[i][j] && can[i][j-1] && can[i][j-2]){
[238]                                 p = prev(i, j-2);
[239]                                 if(p.fi == -1){
[240]                                     dp[i][j][x1][x2][x3][x4] += 1;
[241]                                 }
[242]                                 else dp[i][j][x1][x2][x3][x4] += dp[p.fi][p.se][x1][x2][x3-1][x4];
[243]                             }
[244]                             if(x4 && j-3 >= 0 && can[i][j] && can[i][j-1] && can[i][j-2] && can[i][j-3]){
[245]                                 p = prev(i, j-3);
[246]                                 if(p.fi == -1){
[247]                                     dp[i][j][x1][x2][x3][x4] += 1;
[248]                                 }
[249]                                 else dp[i][j][x1][x2][x3][x4] += dp[p.fi][p.se][x1][x2][x3][x4-1];
[250]                             }
[251]                         }
[252]                     }
[253]                 }
[254]             }
[255]         }
[256]     }
[257]     cout << dp[r-1][c-1][n1-1][n2-1][n3-1][n4-1];
[258] }

```

## Посылка по задаче 7

```
[1] #include <bits/stdc++.h>
[2]
[3]
[4] #define endl '\n'
[5] #define re return
[6] #define se second
[7] #define fi first
[8] #define all(v) v.begin(), v.end()
[9] #define allr(v) v.rbegin(), b.rend()
[10] #define f0r(i, n) for(int i = 0; i < n; i++)
[11] #define f0rr(i, n) for(int i = n - 1; i >= 0; i--)
[12]
[13]
[14] using namespace std;
[15] using ll = long long;
[16] using str = string;
[17] using db = double;
[18] using ldb = long double;
[19]
[20]
[21]
[22] const int MOD = 1e9+7;
[23]
[24]
[25] struct vec{
[26]     ldb x, y;
[27]
[28]     vec() {
[29]         x = 0;
[30]         y = 0;
[31]     }
[32]     vec(ldb _x, ldb _y){
[33]         x = _x;
[34]         y = _y;
[35]     }
[36]
[37]     ldb operator*(vec o) const{
[38]         return x*o.x + y*o.y;
[39]     }
[40]     ldb operator%(vec o) const{
[41]         return x*o.y - y*o.x;
[42]     }
[43]
[44]     vec operator+(vec o) const{
[45]         return {x + o.x, y + o.y};
[46]     }
[47]     vec operator-(vec o) const{
[48]         return {x - o.x, y - o.y};
[49]     }
[50]
[51]     vec operator+(ldb d) const{
[52]         return {x + d, y + d};
[53]     }
[54]     vec operator-(ldb d) const{
[55]         return {x - d, y - d};
[56]     }
[57]     vec operator*(ldb d) const{
[58]         return {x*d, y*d};
[59]     }
[60]     vec operator/(ldb d) const{
[61]         return {x/d, y/d};
[62]     }
[63]
[64]     ldb len(){
[65]         return sqrt(x*x + y*y);
[66]     }
[67] };
[68]
[69] int bpow(int a, int x){
[70]     int ans = 1;
[71]     while(x > 0){
[72]         if(x&1){
[73]             ans = (ll)ans*a%MOD;
[74]         }
[75]         x >>= 1;
[76]         a = (ll)a*a%MOD;
[77]     }
[78]     return ans;
[79] }
[80]
```

```

[81] struct stree{
[82]     int n;
[83]     vector<ll> t;
[84]
[85]     stree(int _n){
[86]         n = _n;
[87]         t.resize(4*n);
[88]     }
[89]     stree(const vector<ll>& a){
[90]         n = a.size();
[91]         t.resize(n*4);
[92]         build(0, 0, n, a);
[93]     }
[94]     int left(int v) {return v * 2 + 1;}
[95]     int right(int v) {return v * 2 + 2;}
[96]
[97]     void pull(int v){
[98]         t[v] = t[left(v)] + t[right(v)];
[99]     }
[100]
[101]     void build(int v, int l, int r, const vector<ll>& a){
[102]         if(r - l == 1){
[103]             t[v] = a[l];
[104]             return;
[105]         }
[106]         int mi = (l + r) / 2;
[107]         build(left(v), l, mi, a); build(right(v), mi, r, a);
[108]         pull(v);
[109]     }
[110]
[111]     void upd(int v, int l, int r, int i, int x){
[112]         if(r - l == 1){
[113]             t[v] = x;
[114]             return;
[115]         }
[116]         int mi = (l + r) / 2;
[117]         if(i < mi) upd(left(v), l, mi, i, x);
[118]         else upd(right(v), mi, r, i, x);
[119]         pull(v);
[120]     }
[121]
[122]     ll get(int v, int l, int r, int ql, int qr){
[123]         if(r <= ql || qr <= l) return 0;
[124]         if(ql <= l && r <= qr) return t[v];
[125]         int mi = (l + r) / 2;
[126]         return get(left(v), l, mi, ql, qr) + get(right(v), mi, r, ql, qr);
[127]     }
[128]
[129]
[130]     void upd(int i, int x){
[131]         upd(0, 0, n, i, x);
[132]     }
[133]     ll get(int l, int r){
[134]         return get(0, 0, n, l, r);
[135]     }
[136] };
[137]
[138] struct LCA{
[139]     int n;
[140]     int lg;
[141]     vector<vector<int>> bup;
[142]     vector<int> tin, tout;
[143]     int timer = 0;
[144]
[145]     LCA(int root, const vector<vector<int>>& g){
[146]         n = g.size();
[147]         lg = (int)log2(n)+2;
[148]         bup.resize(n, vector<int>(lg));
[149]         tin.resize(n); tout.resize(n);
[150]         build_dfs(root, root, g);
[151]     }
[152]
[153]     void build_dfs(int v, int p, const vector<vector<int>>& g){
[154]         tin[v] = timer++;
[155]         bup[v][0] = p;
[156]         for(int k = 1; k < lg; k++){
[157]             bup[v][k] = bup[bup[v][k-1]][k-1];
[158]         }
[159]         for(int u: g[v]){
[160]

```

```

[161]         if(u == p) continue;
[162]         build_dfs(u, v, g);
[163]     }
[164]
[165]     tout[v] = timer++;
[166] }
[167]
[168] bool is_anc(int v, int u){
[169]     return tin[v] <= tin[u] && tout[u] <= tout[v];
[170] }
[171]
[172] int lca(int v, int u){
[173]     if(is_anc(v, u)) return v;
[174]     for(int i = lg - 1; i >= 0; i--){
[175]         if(!is_anc(bup[v][i], u)){
[176]             v = bup[v][i];
[177]         }
[178]     }
[179]     return bup[v][0];
[180] }
[181] };
[182]
[183] const ldb PI = acosl(-1);
[184]
[185] ldb torad(ldb ang){
[186]     return ang/180*PI;
[187] }
[188]
[189] struct node{
[190]     ldb x, y, z;
[191] };
[192]
[193] int main(){
[194]     #ifdef LOCAL
[195]         freopen("input.txt", "r", stdin);
[196]         freopen("output.txt", "w", stdout);
[197]     #endif
[198]
[199]     map<str, node> ma;
[200]     while(true){
[201]         str ch;
[202]         cin >> ch;
[203]         if(ch[0] == '-') break;
[204]         str s;
[205]         cin >> s;
[206]         ldb a = 0;
[207]         str now = "0";
[208]         int i = 0;
[209]         int fl = 1;
[210]         if(s[i] == '-') {
[211]             fl = -1;
[212]             i++;
[213]         }
[214]         while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[215]             now.push_back(s[i]);
[216]             i++;
[217]         }
[218]         a += stold(now);
[219]         now = "0";
[220]         while(!isdigit(s[i])) i++;
[221]         while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[222]             now.push_back(s[i]);
[223]             i++;
[224]         }
[225]         a += (ldb)stold(now)/60;
[226]         now = "0";
[227]         while(!isdigit(s[i])) i++;
[228]         while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[229]             now.push_back(s[i]);
[230]             i++;
[231]         }
[232]         a += (ldb)stold(now)/360;
[233]         now = "0";
[234]         if(fl == -1){
[235]             a = 360 - a;
[236]         }
[237]         if(a > 180){
[238]             a = -(360-a);
[239]         }
[240]         a = torad(a);

```

```

[241]
[242]     ldb b = 0;
[243]     cin >> s;
[244]     i = 0;
[245]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[246]         now.push_back(s[i]);
[247]         i++;
[248]     }
[249]     b += stold(now)*60*60;
[250]     now = "0";
[251]     while(!isdigit(s[i])) i++;
[252]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[253]         now.push_back(s[i]);
[254]         i++;
[255]     }
[256]     b += (ldb)stold(now)*60;
[257]     now = "0";
[258]     while(!isdigit(s[i]))i++;
[259]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[260]         now.push_back(s[i]);
[261]         i++;
[262]     }
[263]     b += (ldb)stold(now);
[264]     now = "0";
[265]     b = b*360/60/60/12;
[266]     if(b > 180){
[267]         b = -(360-b);
[268]     }
[269]     b = torad(b);
[270]
[271]     ldb l;
[272]     cin >> l;
[273]
[274]     swap(a, b);
[275]     node res;
[276]     res.z = l*sinl(b);
[277]
[278]     ldb p1 = l*cosl(b);
[279]     res.x = p1*cosl(a);
[280]     res.y = p1*sinl(a);
[281]
[282]     ma[ch]=res;
[283] }
[284]
[285] bool cn = 0;
[286] ldb ans = 0;
[287] str ch;
[288] while(cin >> ch){
[289]     if(ch[0] == '-') break;
[290]     str s;
[291]     cin >> s;
[292]     ldb a = 0;
[293]     str now = "0";
[294]     int i = 0;
[295]     int fl = 1;
[296]     if(s[i] == '-') {
[297]         fl = -1;
[298]         i++;
[299]     }
[300]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[301]         now.push_back(s[i]);
[302]         i++;
[303]     }
[304]     a += stold(now);
[305]     now = "0";
[306]     while(!isdigit(s[i])) i++;
[307]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[308]         now.push_back(s[i]);
[309]         i++;
[310]     }
[311]     a += (ldb)stold(now)/60;
[312]     now = "0";
[313]     while(!isdigit(s[i]))i++;
[314]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[315]         now.push_back(s[i]);
[316]         i++;
[317]     }
[318]     a += (ldb)stold(now)/360;
[319]     now = "0";
[320]     a *= fl;

```

```

[321]     a = torad(a);
[322]
[323]     ldb b = 0;
[324]     cin >> s;
[325]     i = 0;
[326]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[327]         now.push_back(s[i]);
[328]         i++;
[329]     }
[330]     b += stold(now)*60*60;
[331]     now = "0";
[332]     while(!isdigit(s[i])) i++;
[333]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[334]         now.push_back(s[i]);
[335]         i++;
[336]     }
[337]     b += (ldb)stold(now)*60;
[338]     now = "0";
[339]     while(!isdigit(s[i]))i++;
[340]     while(i < s.size() && (isdigit(s[i]) || s[i] == '.')){
[341]         now.push_back(s[i]);
[342]         i++;
[343]     }
[344]     b += (ldb)stold(now);
[345]     now = "0";
[346]     b = b*360/60/60/12;
[347]     if(b > 180){
[348]         b = -(360-b);
[349]     }
[350]     b = torad(b);
[351]
[352]     ldb l;
[353]     cin >> l;
[354]
[355]     swap(a, b);
[356]     node res;
[357]     res.z = l*sinl(b);
[358]
[359]     ldb pl = l*cosl(b);
[360]     res.x = pl*cosl(a);
[361]     res.y = pl*sinl(a);
[362]
[363]     if(ma.count(ch)){
[364]         cn = 1;
[365]         node res1 = ma[ch];
[366]         ldb x = res.x - res1.x, y = res.y - res1.y, z = res.z - res1.z;
[367]         ldb l1 = sqrtl(x*x + y*y);
[368]         l1 = sqrtl(l1*l1 + z*z);
[369]         ans = max(ans, l1);
[370]     }
[371] }
[372]
[373] if(!cn){
[374]     cout << -1;
[375]     return 0;
[376] }
[377] cout << fixed << setprecision(10) << ans;
[378] cout << endl << 2*(sqrt(3));
[379] }

```