

Олимпиада «Ломоносов» по информатике
2025-2026 учебный год. Заключительный этап
Работа участника с id заявки 1628656, логином inf26f_187

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
828	3:02:20	inf26f_187	1	g++	OK	23	100
427	1:43:27	inf26f_187	5	g++	OK	103	100
265	1:08:56	inf26f_187	4	g++	OK	103	100
219	1:00:47	inf26f_187	3	g++	OK	23	100
49	0:28:29	inf26f_187	2	g++	Partial solution	51	96

496 (четыреста девяносто шесть) технических баллов
71 (семьдесят один) итоговый балл



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<iostream>
[2] #include<vector>
[3] #include<set>
[4] #include<bits/stdc++.h>
[5] #include<cmath>
[6] #include<algorithm>
[7] #include<map>
[8] #include<unordered_map>
[9] #include<unordered_set>
[10] #include<queue>
[11]
[12]
[13] using namespace std;
[14]
[15] using ll = long long;
[16] using ld = long double;
[17] using vi = vector<int>;
[18] using vll = vector<ll>;
[19] ll mod = 1e9 + 7;
[20]
[21] bool isPrime(ll n) {
[22]     if(n == 0 || n == 1) {
[23]         return false;
[24]     }
[25]     for(ll i = 2; i * i <= n; i++) {
[26]         if(n % i == 0) {
[27]             return false;
[28]         }
[29]     }
[30]
[31]     return true;
[32] }
[33]
[34] ll gcd(ll a, ll b) {
[35]     return (b == 0 ? a : gcd(b, a % b));
[36] }
[37]
[38] ll lcm(ll a, ll b) {
[39]     return a / gcd(a, b) * b;
[40] }
[41]
[42] vll buildTree(vll& vec) {
[43]     int n = 1;
[44]     while(n < 2 * vec.size()) {
[45]         n *= 2;
[46]     }
[47]     n--;
[48]
[49]     vll tree(n, 0);
[50]     for(int i = 0, j = n / 2; i < vec.size(); i++, j++) {
[51]         tree[j] = vec[i];
[52]     }
[53]
[54]     for(int i = n / 2 - 1; i >= 0; i--) {
[55]         tree[i] = tree[2 * i + 1] + tree[2 * i + 2];
[56]     }
[57]
[58]     return tree;
[59] }
[60]
[61] ll getSum(vll& tree, int v, int l, int r, int from, int to) {
[62]     if(l > to || r < from) {
[63]         return 0;
[64]     }
[65]     if(from <= l && r <= to) {
[66]         return tree[v];
[67]     }
[68]
[69]     int m = (l + r) / 2;
[70]     return getSum(tree, 2 * v + 1, l, m, from, to) + getSum(tree, 2 * v + 2, m + 1, r, from, to);
[71] }
[72]
[73] void updateUp(vll& tree, int pos, ll val) {
[74]     pos += tree.size() / 2;
[75]     tree[pos] = val;
```

```

[76]     while(pos != 0) {
[77]         pos = (pos - 1) / 2;
[78]         tree[pos] = tree[2 * pos + 1] + tree[2 * pos + 2];
[79]     }
[80]
[81]     return;
[82] }
[83]
[84] string getMn(string a, string b) {
[85]     if(a.length() < b.length()) {
[86]         return a;
[87]     }
[88]     if(a.length() > b.length()) {
[89]         return b;
[90]     }
[91]     return min(a, b);
[92] }
[93]
[94] string getMx(string a, string b) {
[95]     if(a.length() > b.length()) {
[96]         return a;
[97]     }
[98]     if(a.length() < b.length()) {
[99]         return b;
[100]    }
[101]    return max(a, b);
[102] }
[103]
[104] int main() {
[105]     ios_base::sync_with_stdio(0); cin.tie(nullptr); cout.tie(0);
[106]     map<char, int> m;
[107]     m['I'] = 1;
[108]     m['N'] = 0;
[109]     m['V'] = 5;
[110]     m['X'] = 10;
[111]     m['L'] = 50;
[112]     int k;
[113]     cin >> k;
[114]     string mx = "";
[115]     string mn(151, 'z');
[116]     for(int _ = 0; _ < k; _++) {
[117]         string s;
[118]         cin >> s;
[119]         s.push_back('.');
[120]         string res;
[121]         while(find(s.begin(), s.end(), '.') != s.end()) {
[122]             string curr = s.substr(0, s.find('.'));
[123]             s = s.substr(s.find('.') + 1);
[124]             ll sum = 0;
[125]             for(int i = 0; i < curr.length(); i++) {
[126]                 if(curr[i] == 'I') {
[127]                     if(curr.find_last_of('X') > i && curr.find_last_of('X') < curr.length() ||
[128] curr.find_last_of('V') > i && curr.find_last_of('V') < curr.length()) {
[129]                         sum--;
[130]                     }
[131]                     else {
[132]                         sum++;
[133]                     }
[134]                 }
[135]                 if(curr[i] == 'X') {
[136]                     if(curr.find_last_of('L') > i && curr.find_last_of('L') < curr.length()) {
[137]                         sum -= 10;
[138]                     }
[139]                     else {
[140]                         sum += 10;
[141]                     }
[142]                 }
[143]                 if(curr[i] == 'V') {
[144]                     sum += 5;
[145]                 }
[146]                 if(curr[i] == 'L') {
[147]                     sum += 50;
[148]                 }
[149]             }
[150]             if(sum >= 26) {

```

```
[151]         }
[152]         else {
[153]             res.push_back('A' + sum);
[154]         }
[155]     }
[156]     mn = getMn(mn, res);
[157]     mx = getMx(mx, res);
[158] }
[159] cout << mn << endl << mx << endl;
[160] return 0;
[161] }
```

Посылка по задаче 2

```
[1] #include<iostream>
[2] #include<vector>
[3] #include<set>
[4] #include<bits/stdc++.h>
[5] #include<cmath>
[6] #include<algorithm>
[7] #include<map>
[8] #include<unordered_map>
[9] #include<unordered_set>
[10] #include<queue>
[11]
[12]
[13] using namespace std;
[14]
[15] using ll = long long;
[16] using ld = long double;
[17] using vi = vector<int>;
[18] using vll = vector<ll>;
[19] ll mod = 1e9 + 7;
[20]
[21] bool isPrime(ll n) {
[22]     if(n == 0 || n == 1) {
[23]         return false;
[24]     }
[25]     for(ll i = 2; i * i <= n; i++) {
[26]         if(n % i == 0) {
[27]             return false;
[28]         }
[29]     }
[30]
[31]     return true;
[32] }
[33]
[34] ll gcd(ll a, ll b) {
[35]     return (b == 0 ? a : gcd(b, a % b));
[36] }
[37]
[38] ll lcm(ll a, ll b) {
[39]     return a / gcd(a, b) * b;
[40] }
[41]
[42] vll buildTree(vll& vec) {
[43]     int n = 1;
[44]     while(n < 2 * vec.size()) {
[45]         n *= 2;
[46]     }
[47]     n--;
[48]
[49]     vll tree(n, 0);
[50]     for(int i = 0, j = n / 2; i < vec.size(); i++, j++) {
[51]         tree[j] = vec[i];
[52]     }
[53]
[54]     for(int i = n / 2 - 1; i >= 0; i--) {
[55]         tree[i] = tree[2 * i + 1] + tree[2 * i + 2];
[56]     }
[57]
[58]     return tree;
[59] }
[60]
[61] ll getSum(vll& tree, int v, int l, int r, int from, int to) {
[62]     if(l > to || r < from) {
[63]         return 0;
[64]     }
[65]     if(from <= l && r <= to) {
[66]         return tree[v];
[67]     }
[68]
[69]     int m = (l + r) / 2;
[70]     return getSum(tree, 2 * v + 1, l, m, from, to) + getSum(tree, 2 * v + 2, m + 1, r, from, to);
[71] }
[72]
[73] void updateUp(vll& tree, int pos, ll val) {
[74]     pos += tree.size() / 2;
[75]     tree[pos] = val;
```

```

[76]     while(pos != 0) {
[77]         pos = (pos - 1) / 2;
[78]         tree[pos] = tree[2 * pos + 1] + tree[2 * pos + 2];
[79]     }
[80]
[81]     return;
[82] }
[83]
[84] int main() {
[85]     ios_base::sync_with_stdio(0); cin.tie(nullptr); cout.tie(0);
[86]
[87]     vector<bool> primes(160001, true);
[88]     primes[0] = primes[1] = false;
[89]     for (ll i = 2; i < 160001; i++) {
[90]         if(primes[i]) {
[91]             for(ll j = i * i; j < 160001; j += i) {
[92]                 primes[j] = false;
[93]             }
[94]         }
[95]     }
[96]     set<ll> been;
[97]     for(ll i = 2; i < 160001; i++) {
[98]         ll sum = 0;
[99]         ll j = i;
[100]        while(sum < 160001) {
[101]            been.insert(sum);
[102]            sum += j;
[103]            j *= i;
[104]        }
[105]    }
[106]    int n;
[107]    cin >> n;
[108]    vll a(n);
[109]    map<ll, ll> m;
[110]    for (int i = 0; i < n; i++) {
[111]        cin >> a[i];
[112]        if(primes[a[i]]) {
[113]            m[a[i]]++;
[114]        }
[115]    }
[116]    ll mx = 0;
[117]    ll num = 0;
[118]    for(pair<ll, ll> p : m) {
[119]        if(p.second == mx) {
[120]            num = max(num, p.first);
[121]        }
[122]        if(p.second > mx) {
[123]            mx = p.second;
[124]            num = p.first;
[125]        }
[126]    }
[127]    cout << num;
[128]    cout << endl;
[129]
[130]    return 0;
[131] }

```

Посылка по задаче 3

```
[1] #include<iostream>
[2] #include<vector>
[3] #include<set>
[4] #include<bits/stdc++.h>
[5] #include<cmath>
[6] #include<algorithm>
[7] #include<map>
[8] #include<unordered_map>
[9] #include<unordered_set>
[10] #include<queue>
[11]
[12]
[13] using namespace std;
[14]
[15] using ll = long long;
[16] using ld = long double;
[17] using vi = vector<int>;
[18] using vll = vector<ll>;
[19] ll mod = 1e9 + 7;
[20]
[21] bool isPrime(ll n) {
[22]     if(n == 0 || n == 1) {
[23]         return false;
[24]     }
[25]     for(ll i = 2; i * i <= n; i++) {
[26]         if(n % i == 0) {
[27]             return false;
[28]         }
[29]     }
[30]
[31]     return true;
[32] }
[33]
[34] ll gcd(ll a, ll b) {
[35]     return (b == 0 ? a : gcd(b, a % b));
[36] }
[37]
[38] ll lcm(ll a, ll b) {
[39]     return a / gcd(a, b) * b;
[40] }
[41]
[42] vll buildTree(vll& vec) {
[43]     int n = 1;
[44]     while(n < 2 * vec.size()) {
[45]         n *= 2;
[46]     }
[47]     n--;
[48]
[49]     vll tree(n, 0);
[50]     for(int i = 0, j = n / 2; i < vec.size(); i++, j++) {
[51]         tree[j] = vec[i];
[52]     }
[53]
[54]     for(int i = n / 2 - 1; i >= 0; i--) {
[55]         tree[i] = tree[2 * i + 1] + tree[2 * i + 2];
[56]     }
[57]
[58]     return tree;
[59] }
[60]
[61] ll getSum(vll& tree, int v, int l, int r, int from, int to) {
[62]     if(l > to || r < from) {
[63]         return 0;
[64]     }
[65]     if(from <= l && r <= to) {
[66]         return tree[v];
[67]     }
[68]
[69]     int m = (l + r) / 2;
[70]     return getSum(tree, 2 * v + 1, l, m, from, to) + getSum(tree, 2 * v + 2, m + 1, r, from, to);
[71] }
[72]
[73] void updateUp(vll& tree, int pos, ll val) {
[74]     pos += tree.size() / 2;
[75]     tree[pos] = val;
```

```

[76]     while(pos != 0) {
[77]         pos = (pos - 1) / 2;
[78]         tree[pos] = tree[2 * pos + 1] + tree[2 * pos + 2];
[79]     }
[80]
[81]     return;
[82] }
[83]
[84] int main() {
[85]     ios_base::sync_with_stdio(0); cin.tie(nullptr); cout.tie(0);
[86]     ll n, i, j;
[87]     cin >> n >> j >> i;
[88]     ll dist = min({i, j, n - i + 1, n - j + 1});
[89]     ll curr = 0;
[90]     for(ll i = 1; i < dist; i++) {
[91]         ll a = n - 2 * (i - 1);
[92]         curr += 4 * (a - 2) + 4;
[93]     }
[94]
[95]     ll x = dist;
[96]     ll y = dist;
[97]     ll a = n - 2 * (dist - 1);
[98]     if(y == j && i >= x) {
[99]         cout << curr + i - x;
[100]         return 0;
[101]     }
[102]     x += a - 1;
[103]     curr += a - 1;
[104]
[105]     if(x == i && j >= y) {
[106]         cout << curr + j - y;
[107]         return 0;
[108]     }
[109]
[110]     y += a - 1;
[111]     curr += a - 1;
[112]
[113]     if(y == j && i <= x) {
[114]         cout << curr + x - i;
[115]         return 0;
[116]     }
[117]
[118]     x -= a - 1;
[119]     curr += a - 1;
[120]     if(x == i && j <= y) {
[121]         cout << curr + y - j;
[122]         return 0;
[123]     }
[124]     return 0;
[125] }

```

Посылка по задаче 4

```
[1] #include<iostream>
[2] #include<vector>
[3] #include<set>
[4] #include<bits/stdc++.h>
[5] #include<cmath>
[6] #include<algorithm>
[7] #include<map>
[8] #include<unordered_map>
[9] #include<unordered_set>
[10] #include<queue>
[11]
[12]
[13] using namespace std;
[14]
[15] using ll = long long;
[16] using ld = long double;
[17] using vi = vector<int>;
[18] using vll = vector<ll>;
[19] ll mod = 1e9 + 7;
[20]
[21] bool isPrime(ll n) {
[22]     if(n == 0 || n == 1) {
[23]         return false;
[24]     }
[25]     for(ll i = 2; i * i <= n; i++) {
[26]         if(n % i == 0) {
[27]             return false;
[28]         }
[29]     }
[30]
[31]     return true;
[32] }
[33]
[34] ll gcd(ll a, ll b) {
[35]     return (b == 0 ? a : gcd(b, a % b));
[36] }
[37]
[38] ll lcm(ll a, ll b) {
[39]     return a / gcd(a, b) * b;
[40] }
[41]
[42] vll buildTree(vll& vec) {
[43]     int n = 1;
[44]     while(n < 2 * vec.size()) {
[45]         n *= 2;
[46]     }
[47]     n--;
[48]
[49]     vll tree(n, 0);
[50]     for(int i = 0, j = n / 2; i < vec.size(); i++, j++) {
[51]         tree[j] = vec[i];
[52]     }
[53]
[54]     for(int i = n / 2 - 1; i >= 0; i--) {
[55]         tree[i] = tree[2 * i + 1] + tree[2 * i + 2];
[56]     }
[57]
[58]     return tree;
[59] }
[60]
[61] ll getSum(vll& tree, int v, int l, int r, int from, int to) {
[62]     if(l > to || r < from) {
[63]         return 0;
[64]     }
[65]     if(from <= l && r <= to) {
[66]         return tree[v];
[67]     }
[68]
[69]     int m = (l + r) / 2;
[70]     return getSum(tree, 2 * v + 1, l, m, from, to) + getSum(tree, 2 * v + 2, m + 1, r, from, to);
[71] }
[72]
[73] void updateUp(vll& tree, int pos, ll val) {
[74]     pos += tree.size() / 2;
[75]     tree[pos] = val;
```

```

[76]     while(pos != 0) {
[77]         pos = (pos - 1) / 2;
[78]         tree[pos] = tree[2 * pos + 1] + tree[2 * pos + 2];
[79]     }
[80]
[81]     return;
[82] }
[83]
[84] int main() {
[85]     ios_base::sync_with_stdio(0); cin.tie(nullptr); cout.tie(0);
[86]     ll n;
[87]     cin >> n;
[88]
[89]     vll h(n);
[90]     for(int i = 0; i < n; i++) {
[91]         cin >> h[i];
[92]     }
[93]
[94]     vll dp(n, 1);
[95]     for(int i = 1; i < n; i++) {
[96]         for(int j = 0; j < i; j++) {
[97]             if(h[j] >= h[i]) {
[98]                 dp[i] = max(dp[i], dp[j] + 1);
[99]             }
[100]        }
[101]    }
[102]
[103]    cout << n - *max_element(dp.begin(), dp.end());
[104]    return 0;
[105] }

```

Посылка по задаче 5

```
[1] #include<iostream>
[2] #include<vector>
[3] #include<set>
[4] #include<bits/stdc++.h>
[5] #include<cmath>
[6] #include<algorithm>
[7] #include<map>
[8] #include<unordered_map>
[9] #include<unordered_set>
[10] #include<queue>
[11]
[12]
[13] using namespace std;
[14]
[15] using ll = long long;
[16] using ld = long double;
[17] using vi = vector<int>;
[18] using vll = vector<ll>;
[19] ll mod = 1e9 + 7;
[20]
[21] bool isPrime(ll n) {
[22]     if(n == 0 || n == 1) {
[23]         return false;
[24]     }
[25]     for(ll i = 2; i * i <= n; i++) {
[26]         if(n % i == 0) {
[27]             return false;
[28]         }
[29]     }
[30]
[31]     return true;
[32] }
[33]
[34] ll gcd(ll a, ll b) {
[35]     return (b == 0 ? a : gcd(b, a % b));
[36] }
[37]
[38] ll lcm(ll a, ll b) {
[39]     return a / gcd(a, b) * b;
[40] }
[41]
[42] vll buildTree(vll& vec) {
[43]     int n = 1;
[44]     while(n < 2 * vec.size()) {
[45]         n *= 2;
[46]     }
[47]     n--;
[48]
[49]     vll tree(n, 0);
[50]     for(int i = 0, j = n / 2; i < vec.size(); i++, j++) {
[51]         tree[j] = vec[i];
[52]     }
[53]
[54]     for(int i = n / 2 - 1; i >= 0; i--) {
[55]         tree[i] = tree[2 * i + 1] + tree[2 * i + 2];
[56]     }
[57]
[58]     return tree;
[59] }
[60]
[61] ll getSum(vll& tree, int v, int l, int r, int from, int to) {
[62]     if(l > to || r < from) {
[63]         return 0;
[64]     }
[65]     if(from <= l && r <= to) {
[66]         return tree[v];
[67]     }
[68]
[69]     int m = (l + r) / 2;
[70]     return getSum(tree, 2 * v + 1, l, m, from, to) + getSum(tree, 2 * v + 2, m + 1, r, from, to);
[71] }
[72]
[73] void updateUp(vll& tree, int pos, ll val) {
[74]     pos += tree.size() / 2;
[75]     tree[pos] = val;
```

```

[76]     while(pos != 0) {
[77]         pos = (pos - 1) / 2;
[78]         tree[pos] = tree[2 * pos + 1] + tree[2 * pos + 2];
[79]     }
[80]
[81]     return;
[82] }
[83]
[84] int main() {
[85]     ios_base::sync_with_stdio(0); cin.tie(nullptr); cout.tie(0);
[86]     int n;
[87]     cin >> n;
[88]
[89]     vector<vll> v(n, vll(3));
[90]     for(int i = 0; i < n; i++) {
[91]         cin >> v[i][0] >> v[i][1] >> v[i][2];
[92]     }
[93]     sort(v.begin(), v.end());
[94]
[95]     vll dp(n, 1);
[96]     for(int i = 0; i < n; i++) {
[97]         for(int j = 0; j < i; j++) {
[98]             if(v[j][0] < v[i][0] && v[j][1] < v[i][1] && v[j][2] > v[i][2]) {
[99]                 dp[i] = max(dp[i], dp[j] + 1);
[100]            }
[101]        }
[102]    }
[103]
[104]    cout << *max_element(dp.begin(), dp.end());
[105]
[106]    return 0;
[107] }

```

Посылка по задаче 6

Посылок по задаче 6 участником не было отправлено.

Посылка по задаче 7

Посылок по задаче 7 участником не было отправлено.