

Олимпиада «Ломоносов» по информатике
2025-2026 учебный год. Заключительный этап
Работа участника с id заявки 1593510, логином inf26f_130

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
1278	3:56:19	inf26f_130	6	g++	Partial solution	1	0
801	2:56:26	inf26f_130	5	g++	OK	103	100
677	2:33:48	inf26f_130	4	g++	OK	103	100
600	2:20:55	inf26f_130	3	g++	Partial solution	16	65
188	0:56:34	inf26f_130	2	g++	OK	53	100
107	0:40:02	inf26f_130	1	g++	OK	23	100

465 (четыреста шестьдесят пять) технических баллов
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 <iostream>
[2] #include <vector>
[3] #include <algorithm>
[4] #include <cmath>
[5] #include <stack>
[6] #include <set>
[7] #include <map>
[8] #include <unordered_map>
[9] #include <unordered_set>
[10] #include <queue>
[11]
[12] using namespace std;
[13] using T = long long;
[14] using D = long double;
[15] using L = __int128_t;
[16] const int MAXK = 501, MAXS = 151;
[17] map<char, int> cnvrt = {{'N', 0}, {'I', 1}, {'V', 5}, {'X', 10}, {'L', 50}};
[18]
[19] int makeDigit(stack<char>& d) {
[20]     int buffer = 0, res = 0;
[21]     int prev = 50;
[22]     while (!d.empty()) {
[23]         int curr = cnvrt[d.top()];
[24]         d.pop();
[25]         if (curr > prev) {
[26]             res += curr - buffer;
[27]             buffer = 0;
[28]         } else if (curr == prev) {
[29]             buffer += curr;
[30]         } else {
[31]             res += buffer;
[32]             buffer = curr;
[33]         }
[34]
[35]         prev = curr;
[36]     }
[37]
[38]     res += buffer;
[39]     return res;
[40] }
[41]
[42] void proceed(string& s, vector<vector<int>>& nums, int indx) {
[43]     stack<char> digit;
[44]     int currNum = MAXS - 1;
[45]     for (int i = s.size() - 1; i >= 0; i--) {
[46]         if (s[i] == '.') {
[47]             nums[indx][currNum] = makeDigit(digit);
[48]             currNum--;
[49]             continue;
[50]         }
[51]
[52]         digit.push(s[i]);
[53]     }
[54]     nums[indx][currNum] = makeDigit(digit);
[55] }
[56]
[57] int main() {
[58]     ios_base::sync_with_stdio(false);
[59]     cin.tie(nullptr);
[60]     int k;
[61]     cin >> k;
[62]     vector<vector<int>> nums(k, vector<int>(MAXS));
[63]     string num;
[64]     for (int i = 0; i < k; ++i) {
[65]         cin >> num;
[66]         proceed(num, nums, i);
[67]     }
[68]
[69]     sort(nums.begin(), nums.end());
[70]     bool firstNum = false;
[71]     for (auto e1 : nums[0]) {
[72]         if (!firstNum && !e1) continue;
[73]
```

```
[74]     firstNum = true;
[75]     if (el >= 26) {
[76]         cout << (char) ('a' + (el - 26));
[77]     } else {
[78]         cout << (char) ('A' + el);
[79]     }
[80] }
[81] if (!firstNum) cout << 'A';
[82] cout << '\n';
[83]
[84] firstNum = false;
[85] for (auto el : nums[k - 1]) {
[86]     if (!firstNum && !el) continue;
[87]
[88]     firstNum = true;
[89]     if (el >= 26) {
[90]         cout << (char) ('a' + el - 26);
[91]     } else {
[92]         cout << (char) ('A' + el);
[93]     }
[94] }
[95] if (!firstNum) cout << 'A';
[96] cout << '\n';
[97] return 0;
[98] }
```

Посылка по задаче 2

```
[1] #include <iostream>
[2] #include <vector>
[3] #include <algorithm>
[4] #include <cmath>
[5] #include <stack>
[6] #include <set>
[7] #include <map>
[8] #include <unordered_map>
[9] #include <unordered_set>
[10] #include <queue>
[11]
[12] using namespace std;
[13] using T = long long;
[14] using D = long double;
[15] using L = __int128_t;
[16]
[17] bool isBrazil(int num) {
[18]     bool win = false;
[19]     for (int i = 2; i * i <= num; ++i) {
[20]         if (num % i == 0) return false;
[21]         if (num % i != 1 || win) continue;
[22]
[23]         int tmp = num;
[24]         win = true;
[25]         while (tmp) {
[26]             if (tmp % i != 1) {
[27]                 win = false;
[28]                 break;
[29]             }
[30]
[31]             tmp /= i;
[32]         }
[33]     }
[34]
[35]     return win;
[36] }
[37]
[38] int main() {
[39]     ios_base::sync_with_stdio(false);
[40]     cin.tie(nullptr);
[41]     int n;
[42]     cin >> n;
[43]     map<int, int> cnt;
[44]     for (int i = 0, num; i < n; ++i) {
[45]         cin >> num;
[46]         if (isBrazil(num)) cnt[num]++;
[47]     }
[48]
[49]     pair<int, int> res = {0, 0};
[50]
[51]     for (auto el : cnt) {
[52]         res = max(res, {el.second, el.first});
[53]     }
[54]
[55]     cout << res.second << '\n';
[56]     return 0;
[57] }
```

Посылка по задаче 3

```
[1] #include <iostream>
[2] #include <vector>
[3] #include <algorithm>
[4] #include <cmath>
[5] #include <stack>
[6] #include <set>
[7] #include <map>
[8] #include <unordered_map>
[9] #include <unordered_set>
[10] #include <queue>
[11]
[12] using namespace std;
[13] using T = long long;
[14] using D = long double;
[15] using L = __int128_t;
[16]
[17] int main() {
[18]     ios_base::sync_with_stdio(false);
[19]     cin.tie(nullptr);
[20]     T n, a, b;
[21]     cin >> n >> a >> b;
[22]     swap(a, b);
[23]     a--;
[24]     b--;
[25]     if (n == 1) {
[26]         cout << "0\n";
[27]         return 0;
[28]     } else if (n == 2 && a == 2 && b == 2) {
[29]         cout << "2/n";
[30]         return 0;
[31]     }
[32]
[33]     T sq = min(min(n - a, a), min(n - b, b));
[34]     if (b == sq) {
[35]         cout << 411 * sq * (n - sq) + (a - sq) << '\n';
[36]     } else if (b == n - sq - 1) {
[37]         cout << 411 * sq * (n - sq) + 311 * (n - 211 * sq - 1) - (a - sq) << '\n';
[38]     } else if (a == sq) {
[39]         cout << 411 * sq * (n - sq) + 411 * (n - 211 * sq - 1) - (b - sq) << '\n';
[40]     } else {
[41]         cout << 411 * sq * (n - sq) + (n - 211 * sq - 1) + (b - sq) << '\n';
[42]     }
[43]     return 0;
[44] }
```

Посылка по задаче 4

```
[1] #include <iostream>
[2] #include <vector>
[3] #include <algorithm>
[4] #include <cmath>
[5] #include <stack>
[6] #include <set>
[7] #include <map>
[8] #include <unordered_map>
[9] #include <unordered_set>
[10] #include <queue>
[11]
[12] using namespace std;
[13] using T = long long;
[14] using D = long double;
[15] using L = __int128_t;
[16] const int MAXN = 1e4 + 10;
[17]
[18] int sz[MAXN];
[19]
[20] int main() {
[21]     ios_base::sync_with_stdio(false);
[22]     cin.tie(nullptr);
[23]     int n;
[24]     cin >> n;
[25]     fill(sz, sz + n, -1);
[26]     int ans = 0;
[27]     for (int i = 0, curr; i < n; ++i) {
[28]         cin >> curr;
[29]         int l = -1, r = n, mid;
[30]         while (r - l > 1) {
[31]             mid = (r + l) >> 1;
[32]             if (curr > sz[mid]) {
[33]                 r = mid;
[34]             } else {
[35]                 l = mid;
[36]             }
[37]         }
[38]
[39]         sz[r] = curr;
[40]         ans = max(ans, r + 1);
[41]     }
[42]
[43]     cout << n - ans << '\n';
[44]     return 0;
[45] }
```

Посылка по задаче 5

```
[1] #include <iostream>
[2] #include <vector>
[3] #include <algorithm>
[4] #include <cmath>
[5] #include <stack>
[6] #include <set>
[7] #include <map>
[8] #include <unordered_map>
[9] #include <unordered_set>
[10] #include <queue>
[11]
[12] using namespace std;
[13] using T = long long;
[14] using D = long double;
[15] using L = __int128_t;
[16] const int MAXN = 5e3 + 10;
[17]
[18] vector<int> graph[MAXN];
[19] int path[MAXN];
[20]
[21] void dfs(int curr) {
[22]     if (path[curr]) return;
[23]     path[curr] = 1;
[24]
[25]     for (auto el : graph[curr]) {
[26]         if (!path[el]) dfs(el);
[27]         path[curr] = max(path[curr], path[el] + 1);
[28]     }
[29] }
[30]
[31]
[32] int x[MAXN], y[MAXN], h[MAXN];
[33]
[34] int main() {
[35]     ios_base::sync_with_stdio(false);
[36]     cin.tie(nullptr);
[37]     int n;
[38]     cin >> n;
[39]     for (int i = 0; i < n; ++i) {
[40]         cin >> x[i] >> y[i] >> h[i];
[41]         path[i] = 0;
[42]         for (int j = 0; j < i; ++j) {
[43]             if (x[i] > x[j] && y[i] > y[j] && h[i] < h[j]) {
[44]                 graph[i].push_back(j);
[45]             } else if (x[j] > x[i] && y[j] > y[i] && h[j] < h[i]) {
[46]                 graph[j].push_back(i);
[47]             }
[48]         }
[49]     }
[50]
[51]     int ans = 0;
[52]     for (int i = 0; i < n; ++i) {
[53]         dfs(i);
[54]         ans = max(ans, path[i]);
[55]     }
[56]
[57]     cout << ans << '\n';
[58]     return 0;
[59] }
```

Посылка по задаче 6

```
[1] #include <iostream>
[2] #include <vector>
[3] #include <algorithm>
[4] #include <cmath>
[5] #include <stack>
[6] #include <set>
[7] #include <map>
[8] #include <unordered_map>
[9] #include <unordered_set>
[10] #include <queue>
[11]
[12] using namespace std;
[13] using T = long long;
[14] using D = long double;
[15] using L = __int128_t;
[16] const int MAXR = 21;
[17]
[18] bool field[MAXR][MAXR];
[19]
[20] struct Position {
[21]     vector<vector<int>> n1, n2, n3, n4;
[22]     Position(int a, int b, int d, int e) {
[23]         n1.resize(a, vector<int>(2));
[24]         n2.resize(b, vector<int>(3));
[25]         n3.resize(d, vector<int>(3));
[26]         n4.resize(e, vector<int>(3));
[27]     }
[28]
[29]     void srt() {
[30]         sort(n1.begin(), n1.end());
[31]         sort(n2.begin(), n2.end());
[32]         sort(n3.begin(), n3.end());
[33]         sort(n4.begin(), n4.end());
[34]     }
[35]
[36]     bool operator<(const Position& a) const {
[37]         if (a.n1 != n1 || a.n2 != n2 || a.n3 != n3 || a.n4 != n4) return true;
[38]         return false;
[39]     }
[40]
[41]     bool operator==(const Position& a) const {
[42]         if (a.n1 != n1 || a.n2 != n2 || a.n3 != n3 || a.n4 != n4) return false;
[43]         return true;
[44]     }
[45] };
[46]
[47] set<Position> uniq;
[48] int r, c;
[49]
[50] void rec(vector<vector<int>>& water, Position& curr, int n1, int n2, int n3, int n4) {
[51]     if (n1) {
[52]         for (int i = 0; i < r; ++i) {
[53]             for (int j = 0; j < c; ++j) {
[54]                 if (field[i][j]) {
[55]                     water[i][j]++;
[56]                     curr.n1[n1 - 1] = {i, j};
[57]                     rec(water, curr, n1 - 1, n2, n3, n4);
[58]                     water[i][j]--;
[59]                 }
[60]             }
[61]         }
[62]     } else if (n2) {
[63]         for (int i = 0; i < r; ++i) {
[64]             for (int j = 0; j < c; ++j) {
[65]                 if (field[i][j]) {
[66]                     water[i][j]++;
[67]                     if (i + 1 < r && field[i + 1][j]) {
[68]                         water[i + 1][j]++;
[69]                         curr.n2[n2 - 1] = {i, j, 0};
[70]                         rec(water, curr, n1, n2 - 1, n3, n4);
[71]                         water[i + 1][j]--;
[72]                     }
[73]                     if (j + 1 < c && field[i][j + 1]) {
[74]                         water[i][j + 1]++;
[75]                         curr.n2[n2 - 1] = {i, j, 1};
[76]                         rec(water, curr, n1, n2 - 1, n3, n4);
```

```

[77]         water[i][j + 1]--;
[78]     }
[79]     water[i][j]--;
[80] }
[81] }
[82] }
[83] } else if (n3) {
[84]     for (int i = 0; i < r; ++i) {
[85]         for (int j = 0; j < c; ++j) {
[86]             if (field[i][j]) {
[87]                 water[i][j]++;
[88]                 if (i + 2 < r && field[i + 1][j] && field[i + 2][j]) {
[89]                     water[i + 1][j]++;
[90]                     water[i + 2][j]++;
[91]                     curr.n3[n3 - 1] = {i, j, 0};
[92]                     rec(water, curr, n1, n2, n3 - 1, n4);
[93]                     water[i + 1][j]--;
[94]                     water[i + 2][j]--;
[95]                 }
[96]                 if (j + 2 < c && field[i][j + 1] && field[i][j + 2]) {
[97]                     water[i][j + 1]++;
[98]                     water[i][j + 2]++;
[99]                     curr.n3[n3 - 1] = {i, j, 1};
[100]                    rec(water, curr, n1, n2, n3 - 1, n4);
[101]                    water[i][j + 1]--;
[102]                    water[i][j + 2]--;
[103]                }
[104]                water[i][j]--;
[105]            }
[106]        }
[107]    }
[108] } else if (n4) {
[109]     for (int i = 0; i < r; ++i) {
[110]         for (int j = 0; j < c; ++j) {
[111]             if (field[i][j]) {
[112]                 water[i][j]++;
[113]                 if (i + 3 < r && field[i + 1][j] && field[i + 2][j] && field[i + 3][j]) {
[114]                     water[i + 1][j]++;
[115]                     water[i + 2][j]++;
[116]                     water[i + 3][j]++;
[117]                     curr.n4[n4 - 1] = {i, j, 0};
[118]                     rec(water, curr, n1, n2, n3, n4 - 1);
[119]                     water[i + 1][j]--;
[120]                     water[i + 2][j]--;
[121]                     water[i + 3][j]--;
[122]                 }
[123]                 if (j + 3 < c && field[i][j + 1] && field[i][j + 2] && field[i][j + 3]) {
[124]                     water[i][j + 1]++;
[125]                     water[i][j + 2]++;
[126]                     water[i][j + 3]++;
[127]                     curr.n4[n4 - 1] = {i, j, 1};
[128]                     rec(water, curr, n1, n2, n3, n4 - 1);
[129]                     water[i][j + 1]--;
[130]                     water[i][j + 2]--;
[131]                     water[i][j + 3]--;
[132]                 }
[133]                water[i][j]--;
[134]            }
[135]        }
[136]    }
[137] } else {
[138]     for (int i = 0; i < r; ++i) {
[139]         for (int j = 0; j < c; ++j) {
[140]             if (field[i][j] && water[i][j] <= 0) return;
[141]         }
[142]     }
[143]
[144]     curr.srt();
[145]     uniq.insert(curr);
[146] }
[147] }
[148]
[149] int main() {
[150]     ios_base::sync_with_stdio(false);
[151]     cin.tie(nullptr);
[152]     int n1, n2, n3, n4;
[153]     cin >> n1 >> n2 >> n3 >> n4 >> r >> c;

```

```
[154]     for (int i = 0; i < r; ++i) {
[155]         for (int j = 0; j < c; ++j) {
[156]             char type;
[157]             cin >> type;
[158]             field[i][j] = (type == '#');
[159]         }
[160]     }
[161]     Position curr(n1, n2, n3, n4);
[162]     vector<vector<int>> water(r, vector<int>(c, 0));
[163]     rec(water, curr, n1, n2, n3, n4);
[164]     cout << uniq.size() - 1 << '\n';
[165]     return 0;
[166] }
```

Посылка по задаче 7

Посылок по задаче 7 участником не было отправлено.