



PRACTICE

COMPETE

JOBS

LEADERBOARD

Search



swatantragoswam1 ▾

[All Contests](#) > [GCFL_3_year_6_sem](#) > [Castle on the Grid](#)

Castle on the Grid

locked

Problem

Submissions

Leaderboard

Discussions

Editorial

Submitted a few seconds ago • Score: 1.00

Status: **Accepted**

| | | | | | |
|---|---------------|---|---------------|---|---------------|
| ✓ | Test Case #0 | ✓ | Test Case #1 | ✓ | Test Case #2 |
| ✓ | Test Case #3 | ✓ | Test Case #4 | ✓ | Test Case #5 |
| ✓ | Test Case #6 | ✓ | Test Case #7 | ✓ | Test Case #8 |
| ✓ | Test Case #9 | ✓ | Test Case #10 | ✓ | Test Case #11 |
| ✓ | Test Case #12 | ✓ | Test Case #13 | | |

Submitted Code

Language: C++

Open in editor

[Privacy](#) - [Terms](#)

```
1 // Sample Input
2
3 // 3
4 // .X.
5 // .X.
6 // ...
7 // 0 0 0 2
8 // Sample Output
9
10 // 3
11
12
13
14
15 #include <cmath>
16 #include <cstdio>
17 #include <vector>
18 #include <iostream>
19 #include <algorithm>
20
21 #include <queue>
22 using namespace std;
23 char mark = 'X';
24
25 int main() {
26     int n;
27     cin >> n;
28
29     char c, m[n][n];
30     for (int i = 0; i < n; ++i) {
31         for (int j = 0; j < n; ++j) {
32             cin >> c;
33             m[i][j] = c;
34         }
```

```
35     }
36
37     int xs, ys, xd, yd;
38     cin >> xs >> ys >> xd >> yd;
39
40     queue<int> dist;
41     queue<pair<int, int>> q;
42     q.push(make_pair(xs, ys));
43     dist.push(0);
44     m[xs][ys] = mark;
45
46     vector<pair<int, int>> mv = { make_pair(-1, 0), make_pair(0, +1), make_pair(+1, 0), make_pair(0,
-1) };
47
48     while (!q.empty())
49     {
50         pair<int, int> cur = q.front();
51
52         if (cur.first == xd && cur.second == yd)
53         {
54             cout << dist.front() << endl;
55             break;
56         }
57
58         for (int i = 0; i < mv.size(); ++i)
59         {
60             int newX = cur.first + mv[i].first;
61             int newY = cur.second + mv[i].second;
62
63             while (newX >= 0 && newX < n && newY >= 0 && newY < n && m[newX][newY] != mark)
64             {
65                 q.push(make_pair(newX, newY));
66                 dist.push(dist.front() + 1);
67                 m[newX][newY] = mark;
68             }
```

```
69         newX += mv[i].first;
70         newY += mv[i].second;
71     }
72 }
73
74     q.pop();
75     dist.pop();
76 }
77
78     return 0;
79 }
```

[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)