Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
8035437	18:13:33	Grid	✔ Accepted	0.01 s	C++
	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔ ✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔ ✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔ ✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔ ✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
grid.cpp	2106 bytes	8af9bcd760a938f3b2b29cbd50ec44cfc23757fa	download

Edit and resubmit this submission.

grid.cpp

```
#include <bits/stdc++.h>
int inf = 2 << 28;
class Solution {
public:
  bool inrange(int n, int m, int x, int y) {
      if(x < 0) {
          return false;
      }
      if(y < 0) {
          return false;
      }
      if(x >= n) {
          return false;
      }
      if(y >= m) {
          return false;
      }
      return true;
  }
};
int main(){
  std::ios_base::sync_with_stdio(false);
  std::cin.tie(NULL);
  Solution S1= Solution();
  [illegible] m;
  [illegible]cin >> n >> m;

    std::vector<std::vector<int>> v;
```

Help

```
29      v.resize(n, std::vector<int>(m));
30
31      for(int i = 0; i < n; i++) {
32          for(int j = 0; j < m; j++) {
33              char c;
34              std::cin >> c;
35              v[i][j] = c - '0';
36          }
37      }
38
39      std::vector<std::vector<int>> dist;
40      dist.resize(n, std::vector<int>(m, inf));
41
42      std::queue<std::pair<int, int>> q;
43      dist[0][0] = 0;
44      q.push({0,0});
45
46      while(!q.empty()) {
47          int x = q.front().first;
48          int y = q.front().second;
49          q.pop();
50
51          int range = v[x][y];
52          if(S1.inrange(n, m, x-range, y)) {
53              if(dist[x][y] + 1 < dist[x-range][y]) {
54                  dist[x-range][y] = dist[x][y] + 1;
55                  q.push({x-range,y});
56              }
57          }
58          if(S1.inrange(n, m, x+range, y)) {
59              if(dist[x][y] + 1 < dist[x+range][y]) {
60                  dist[x+range][y] = dist[x][y] + 1;
61                  q.push({x+range,y});
62              }
63          }
64          if(S1.inrange(n, m, x, y-range)) {
65              if(dist[x][y] + 1 < dist[x][y-range]) {
66                  dist[x][y-range] = dist[x][y] + 1;
67                  q.push({x,y-range});
68              }
69          }
70          if(S1.inrange(n, m, x, y+range)) {
71              if(dist[x][y] + 1 < dist[x][y+range]) {
72                  dist[x][y+range] = dist[x][y] + 1;
73                  q.push({x,y+range});
74              }
75          }
76      }
77
78      int d = dist[n-1][m-1];
79      if(d == inf) {
80          std::cout << "-1" << std::endl;
81      }
82      else {
83          std::cout << d << std::endl;
84      }
85    return 0;
86  }
```