시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	192 MB	147262	64127	41166	42.253%

문제

N×M크기의 배열로 표현되는 미로가 있다.

1	0	1	1	1	1
1	0	1	0	1	0
1	0	1	0	1	1
1	1	1	0	1	1

미로에서 1은 이동할 수 있는 칸을 나타내고, 0은 이동할 수 없는 칸을 나타낸다. 이러한 미로가 주어졌을 때, (1, 1)에서 출발하여 (N, M)의 위치로 이동할 때 지나야 하는 최소의 칸 수를 구하는 프로그램을 작성하시오. 한 칸에서 다른 칸으로 이동할 때, 서로 인접한 칸으로만 이동할 수 있다.

위의 예에서는 15칸을 지나야 (N, M)의 위치로 이동할 수 있다. 칸을 셀 때에는 시작 위치와 도착 위치도 포함한다.

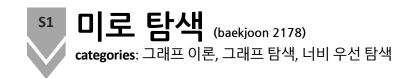
입력

첫째 줄에 두 정수 N, M($2 \le N$, M ≤ 100)이 주어진다. 다음 N개의 줄에는 M개의 정수로 미로가 주어진다. 각각의 수들은 **붙어서** 입력으로 주어진다.

출력

첫째 줄에 지나야 하는 최소의 칸 수를 출력한다. 항상 도착위치로 이동할 수 있는 경우만 입력으로 주어진다.





입/출력 예시 1

4 6 101111 101010 101011 111011

입/출력 예시 2

 4 6

 110110

 110110

 111111

 111101

입/출력 예시 3

38

입/출력 예시 4

 13

Answer(1/1)

```
1
    N, M = 0, 0
 2
 3
    def check_valid_pos(x_pos, y_pos):
 4
         if 0 \le x_pos < N and 0 \le y_pos < M:
 5
             return True
 6
 7
         return False
8
9
10
    delta = [(-1, 0), (1, 0), (0, -1), (0, 1)] # up, down, left, right
11
12
13
    def bfs(graph, start_node):
14
         queue = [start_node]
15
         visited = [[False] * M for _ in range(N)]
16
        while queue:
17
18
             cur = queue.pop(0)
19
20
             if not visited[cur[0]][cur[1]]:
                 visited[cur[0]][cur[1]] = True
21
22
23
                 for k in range(len(delta)):
                     next_x = cur[0] + delta[k][0]
24
25
                     next_y = cur[1] + delta[k][1]
26
27
                     # check next pos valid
28
                     if check_valid_pos(next_x, next_y):
29
                         if graph[next_x][next_y] != 0 and not visited[next_x][next_y]:
30
                              graph[next_x][next_y] = graph[cur[0]][cur[1]] + 1
31
                              queue.append((next_x, next_y))
32
         return visited
33
34
35
    N, M = map(int, input().split(" "))
36
    graph = []
37
38
    for _ in range(N):
39
        graph.append(list(map(int, input())))
40
    bfs(graph, (0, 0))
41
    print(graph[N-1][M-1])
42
43
```