**미로 찾기**

**다음 그림과 같은 미로가 있다. 한번도 가보지 않은 길은 0으로 벽은 1로 표시가 되어있다. 출구는 오른쪽 맨 아래칸이라 할 때, 길을 찾다가 막혀 있는 길이라면 3으로 표시를 하고, 막혀있지 않은 출구까지의 길이라면(즉, 미로 찾기의 정답 경로이면) 2로 표시를 한다고 할 때, 주어진 미로 maze1, maze2, maze3, maze4에 출구까지의 길이 있는지 확인하는 함수 findPath(maze,x,y)를 작성하시오.(주어진 미로에 출구까지의 길이 있다면 findPath는 True를 리턴하고, 그렇지 않으면 False를 반환한다.)**

**0 : UNVISITED\_WAY**

**1 : WALL**

**2 : POSSIBLE\_WAY**

**3 : BLOCKED**

**START**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 3 | 2 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

**FINISH**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 3 | 3 | 3 | 3 | 3 | 3 | 1 |
| 2 | 1 | 1 | 3 | 1 | 1 | 3 | 1 |
| 2 | 3 | 3 | 1 | 3 | 3 | 3 | 1 |
| 2 | 1 | 3 | 3 | 1 | 1 | 3 | 3 |
| 2 | 1 | 1 | 1 | 3 | 3 | 1 | 1 |
| 2 | 1 | 2 | 2 | 2 | 1 | 3 | 1 |
| 2 | 2 | 2 | 1 | 2 | 2 | 2 | 1 |
| 3 | 1 | 1 | 1 | 3 | 1 | 2 | 2 |

def findPath(maze,x,y):

// fill here!  
  
  
maze1 = [[0,0,0,0,0,0,0,1],  
 [0,1,1,0,1,1,0,1],  
 [0,0,0,1,0,0,0,1],  
 [0,1,0,0,1,1,0,0],  
 [0,1,1,1,0,0,1,1],  
 [0,1,0,0,0,1,0,1],  
 [0,0,0,1,0,0,0,1],  
 [0,1,1,1,0,1,0,0]]  
print(findPath(maze1,0,0))

maze2 = [[0,1,0,0,0,0,0,1],  
 [0,1,1,0,1,1,0,1],  
 [0,0,0,1,0,0,0,1],  
 [0,1,0,0,1,1,0,0],  
 [0,1,1,1,0,0,1,1],  
 [0,1,0,0,0,1,0,1],  
 [0,0,0,1,0,0,0,1],  
 [0,1,1,1,0,1,0,0]]  
print(findPath(maze2,0,0))

maze3 = [[0,0,0,0,0,0,0,1],  
 [1,1,1,0,1,1,0,1],  
 [1,0,0,1,0,0,0,1],  
 [1,1,0,0,1,1,0,0],  
 [1,1,1,1,0,0,1,1],  
 [0,1,0,0,0,1,0,1],  
 [0,0,0,1,0,0,0,1],  
 [0,1,1,1,0,1,0,0]]  
print(findPath(maze3,0,0))  
  
maze4 = [[0,0,0,0,0,0,0,1],  
 [1,1,1,0,1,1,0,1],  
 [1,0,0,0,0,0,1,1],  
 [1,0,1,0,1,1,0,0],  
 [1,0,0,1,0,0,0,1],  
 [0,1,0,0,0,1,1,1],  
 [0,0,0,1,1,1,0,1],  
 [0,1,0,0,0,0,0,0]]  
print(findPath(maze4,0,0))

**HINT**

**1)** True or False = True