EXPERIMENT 2

Aim:

To Solve a Problem using Depth First Search(DFS) in Python.

Code:

```
maze_size = 6
obstacles = [(0,1),(1,1),(3,2),(3,3),(3,4),(3,5),(0,4),(4,1),(4,2),(4,3)]
start = (0,0)
goal = (0,5)


def is_valid(x,y):
  return 0 <= x < maze_size and 0 <= y < maze_size and (x,y) not
in obstacles


def dfs (current, visited, path):
  x, y = current
  if current == goal:
    path.append(current)
    return True
  visited.add(current)
  moves = [(x-1,y), (x+1, y), (x, y-1), (x, y+1)]
  for move in moves:
    if is_valid(*move) and move not in visited:
      if dfs(move, visited, path):
        path.append(current)
        return True
  return False

visited = set()
path = []
if dfs(start, visited, path):
  path.reverse()
  print("Path found:")
  for position in path:
    print(position)
else:
  print("No path found!")
```

Output:

```
Path found:
(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
(2, 1)
(2, 2)
(1, 2)
(0, 2)
(0, 3)
```

```
(1, 3)

(2, 3)

(2, 4)

(1, 4)

(1, 5)

(0, 5)
```

Result:
Thus the Problem has been Solved in DFS.