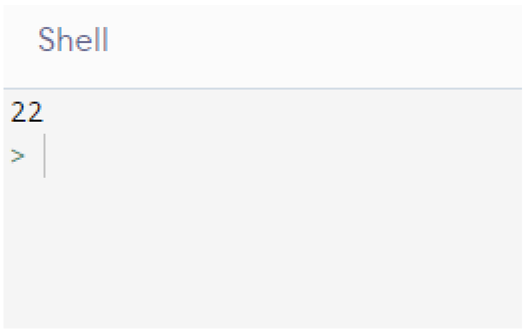
ASSIGNMENT-4

AIM: Implement 8- Puzzle problem.

CODE:

```
class Puzzle:
  def solve(self, board):
    dict = {}
    flatten = []
    for a in range(len(board)):
      flatten += board[a]
    flatten = tuple(flatten)

    dict[flatten] = 0

    if flatten == (0, 1, 2, 3, 4, 5, 6, 7, 8):
      return 0

    return self.get_paths(dict)

  def get_paths(self, dict):
    cnt = 0
    while True:
      current_nodes = [x for x in dict if dict[x] == cnt]
      if len(current_nodes) == 0:
        return -1

      for node in current_nodes:
        next_moves = self.find_next(node)
        for move in next_moves:
          if move not in dict:
            dict[move] = cnt + 1
          if move == (0, 1, 2, 3, 4, 5, 6, 7, 8):
            return cnt + 1
      cnt += 1

  def find_next(self, node):
    moves = {
      0: [1, 2],
      1: [0, 3, 4],
      2: [1, 5],
      3: [0, 4, 6],
      4: [1, 3, 4, 5],
      5: [3, 6, 8],
      6: [2, 7],
      7: [4, 7, 8],
      8: [5, 6],
    }

    results = []
    pos_0 = node.index(0)
    for move in moves[pos_0]:
      new_node = list(node)
      new_node[move], new_node[pos_0] = new_node[pos_0], new_node[move]
      results.append(tuple(new_node))

    return results
ob = Puzzle()
matrix = [
  [0, 5, 6],
  [8, 4, 7],
  [1, 3, 2]
]
print(ob.solve(matrix))
```

OUTPUT:



```
22
>
```