# Evidence 1

To represent the puzzle as a grid, a 4x4 2-dimensional array, called board, can be used, representing the rows and columns of the grid.

The three modules used are CreateBoard(), OutputBoard(board), and DisplayBoard(board).

To declare the puzzle, a function, CreatePuzzle(), can be run, which creates the 2-dimensional array with 4 rows and 4 columns, board, containing the initial 16 values as shown in the example grid and returns the 2-dimension array, board, created.

In order to return the string representation of the board, a function, OutputBoard, given a parameter board, returns board in a format such that columns are separated using space characters, and rows are separated using a newline character.

In order to display the string representation of a board onto the screen, a procedure, DisplayBoard, given a parameter board, prints board onto the screen in a format as specified by the OutputBoard function.

# Evidence 2

def CreateBoard():

board = [[4, 3, 2, 1], \

[1, 2, 4, 3], \

[3, 4, 1, 2], \

[2, 1, 3, 4]]

return board

def OutputBoard(board):

output = ""

for i in range(4): # for each row in the board

for j in range(4): # for each cell in the row

cell = str(board[i][j])

if j == 3 and i == 3: # last cell on the last row

output += cell

elif j == 3: # last cell in the row

output += cell + "\n"

else:

output += cell + " "

return output

def DisplayBoard(board):

print(OutputBoard(board))

# Evidence 3

# Evidence 4