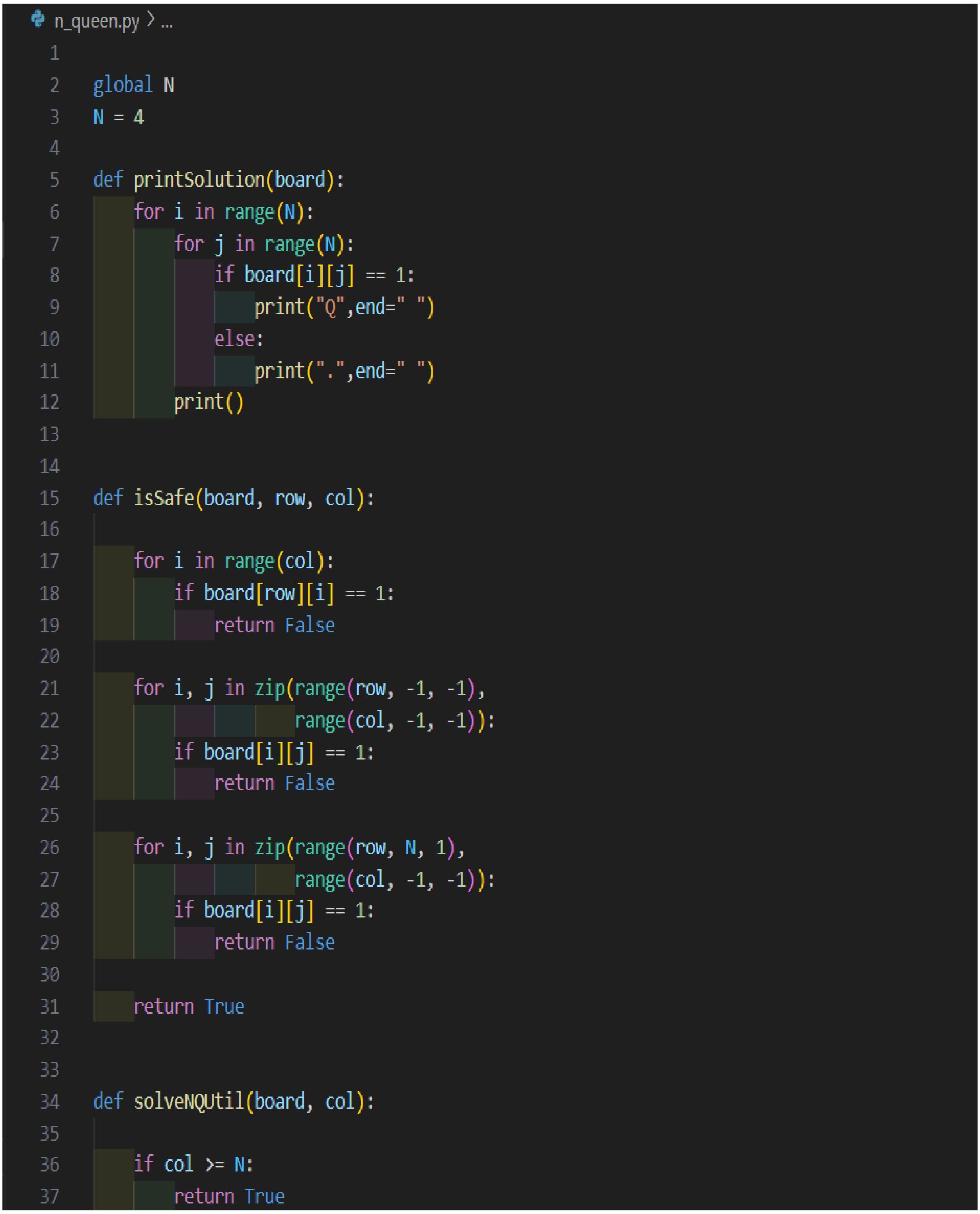
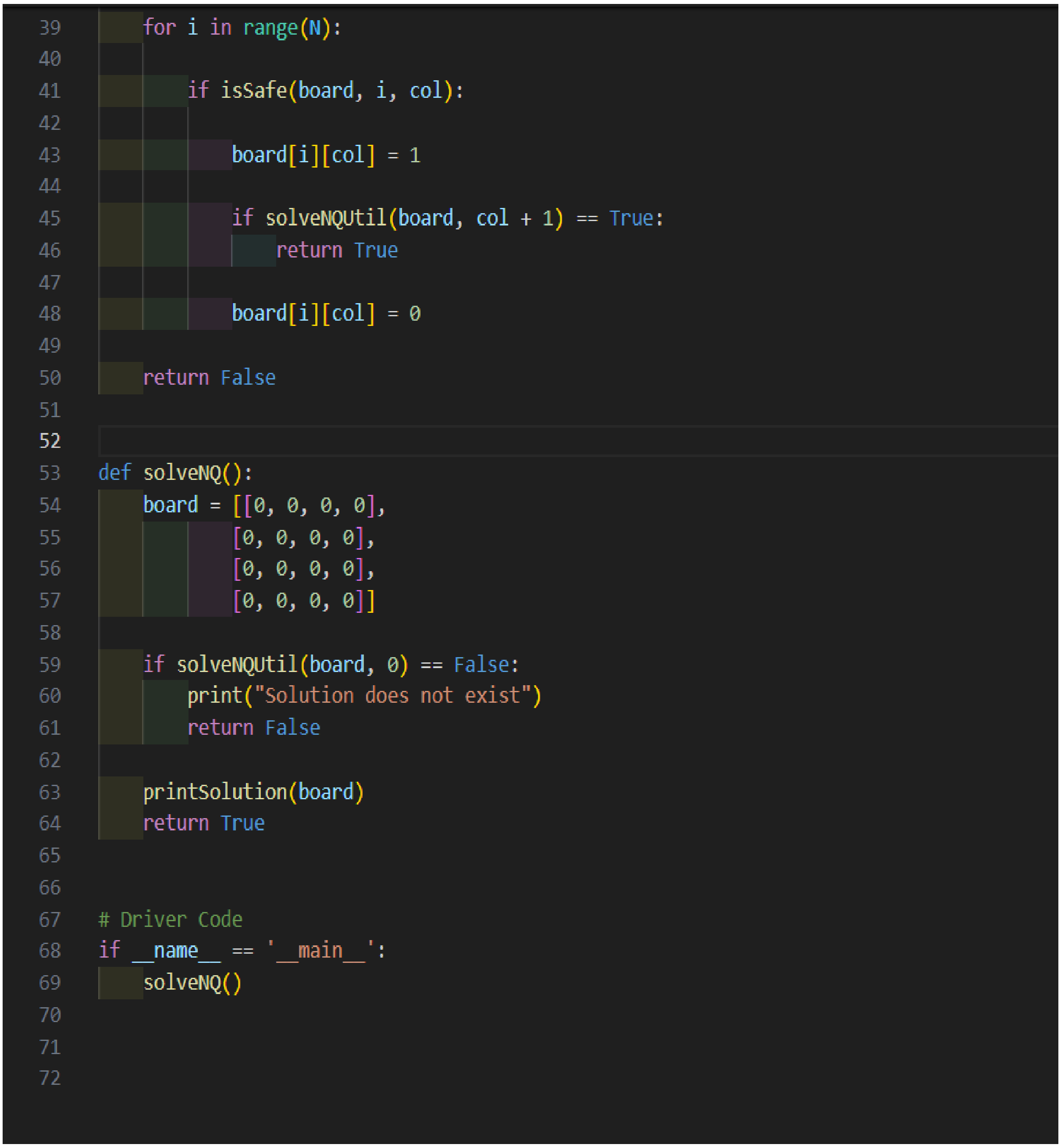
CSP: N - Queens Problem

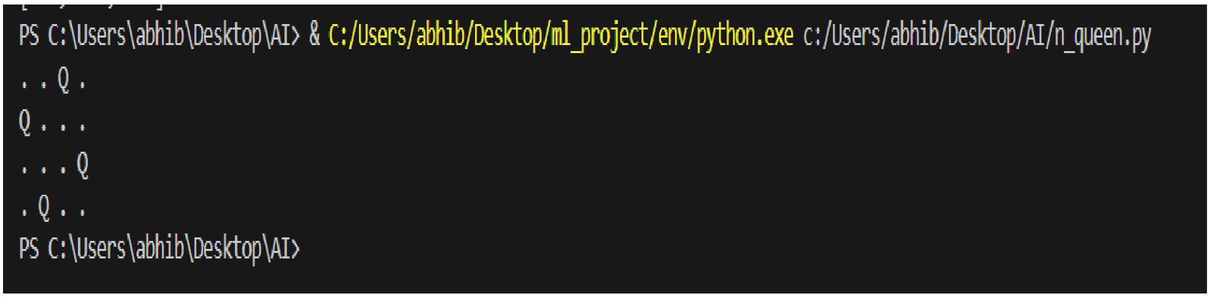


```python

global N
N = 4

def printSolution(board):
    for i in range(N):
        for j in range(N):
            if board[i][j] == 1:
                print("Q",end=" ")
            else:
                print(".",end=" ")
        print()


def isSafe(board, row, col):

    for i in range(col):
        if board[row][i] == 1:
            return False

    for i, j in zip(range(row, -1, -1),
                    range(col, -1, -1)):
        if board[i][j] == 1:
            return False

    for i, j in zip(range(row, N, 1),
                    range(col, -1, -1)):
        if board[i][j] == 1:
            return False

    return True


def solveNQUtil(board, col):

    if col >= N:
        return True
```



```python
    for i in range(N):

        if isSafe(board, i, col):

            board[i][col] = 1

            if solveNQUtil(board, col + 1) == True:
                return True

            board[i][col] = 0

    return False


def solveNQ():
    board = [[0, 0, 0, 0],
             [0, 0, 0, 0],
             [0, 0, 0, 0],
             [0, 0, 0, 0]]

    if solveNQUtil(board, 0) == False:
        print("Solution does not exist")
        return False

    printSolution(board)
    return True


# Driver Code
if __name__ == '__main__':
    solveNQ()
```

Output:



```
PS C:\Users\abhib\Desktop\AI> & C:/Users/abhib/Desktop/ml_project/env/python.exe c:/Users/abhib/Desktop/AI/n_queen.py
. . Q .
Q . . .
. . . Q
. Q . .
PS C:\Users\abhib\Desktop\AI>
```