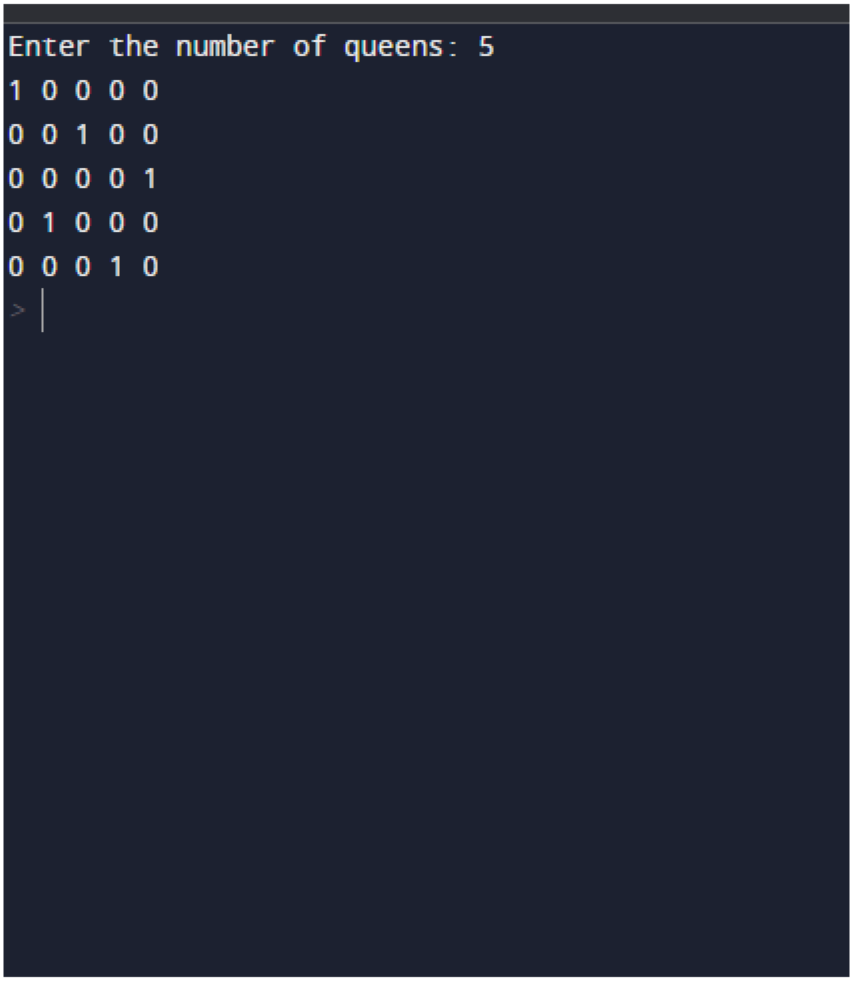
Assignment 4 : N Queens Problem

```
def is_safe(board, row, col, n):
    # Check if there is a queen in the same column
    for i in range(row):
        if board[i][col] == 1:
            return False

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

    # Check upper-right diagonal
    for i, j in zip(range(row, -1, -1), range(col, n)):
        if board[i][j] == 1:
            return False

    return True

def solve_n_queens_util(board, row, n):
    if row == n:
        # All queens are placed, solution found
        print_board(board, n)
        return True

    for col in range(n):
        if is_safe(board, row, col, n):
            board[row][col] = 1
            if solve_n_queens_util(board, row + 1, n):
                return True
            board[row][col] = 0

    return False

def print_board(board, n):
    for i in range(n):
        for j in range(n):
            print(board[i][j], end=" ")
        print()

def solve_n_queens(n):
    board = [[0] * n for _ in range(n)]
    if not solve_n_queens_util(board, 0, n):
        print("No solution exists")

if __name__ == "__main__":
    try:
        n = int(input("Enter the number of queens: "))
        solve_n_queens(n)
    except ValueError:
        print("Invalid input. Please enter a valid number.")
```



```
Enter the number of queens: 5
1 0 0 0 0
0 0 1 0 0
0 0 0 0 1
0 1 0 0 0
0 0 0 1 0
>
```