DAA – Assignment no: 05

Write a Python Program to Design n-Queens matrix having first Queen placed. Use backtracking to place remaining Queens to generate the final n-queen's matrix.

Input:

Assignment5.py

```python
def print_board(board):
    for row in board:
        print(' '.join(map(str, row)))
    print()

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

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

    return True

def solve_n_queens(board, row, n):
    if row == n:
        print_board(board)
        return

    for col in range(n):
        if is_safe(board, row, col, n):
            board[row][col] = 1
            solve_n_queens(board, row + 1, n)
            board[row][col] = 0  # backtrack

```

Assignment5.py

```python

def n_queens(n):
    # Initialize the chessboard with empty cells
    board = [[0 for _ in range(n)] for _ in range(n)]

    # Place the first queen in the first row
    board[0][2] = 1  # Placing the first queen in the third column (0-indexed)

    # Start placing queens from the second row
    solve_n_queens(board, 1, n)

# Example for 4-Queens
n_queens(4)

```

Output:

```
PS D:\Tanmay Mohadikar\Sem 7 Practicals\DAA\Code file> & D:/Python/python.exe "d:/Tanmay Mohadikar/Sem 7 Practicals/DAA/Code file/Assignment4.py"
Maximum value in Knapsack: 220
PS D:\Tanmay Mohadikar\Sem 7 Practicals\DAA\Code file> & D:/Python/python.exe "d:/Tanmay Mohadikar/Sem 7 Practicals/DAA/Code file/Assignment5.py"
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0

PS D:\Tanmay Mohadikar\Sem 7 Practicals\DAA\Code file>
```