The Superior University

Name: Fatima Nadeem	Roll No:076	Course: PAI Lab
Semester:4th	Section:4B	Department:SE
Submitted To: Sir Rasikh	Total Marks: 10	Date:

Lab Task 04

N-Queen Problem

The N-Queens Problem (Dynamic Version):

Visualize a chessboard of any dimension (not only 8×8) and you have N queens to put on it. The task? No two queens can shoot at each other. That is:

Rules:

- No two queens in the same column
- No two queens in the same row
- No two queens in the same diagonal

Dynamic:

- Pick any board size (e.g., 4×4, 6×6, 10×10, etc.)
- Halt the game at any point (rather than solving the entire problem)

This makes the traditional N-Queens Problem an interactive game in which the player chooses when to proceed or stop. It's similar to solving a puzzle, step by step, instead of being compelled to discover all the possible solutions simultaneously.

Code:

```
def board_style(board):
    for row in board:
        print(" ".join(row))
    print()


def is_valid(board,row,col,N):
    for i in range(N):
        if board[i][col]=='q':
            return False


    for j in range(N):
        if board[row][j]=='q':
            return False


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

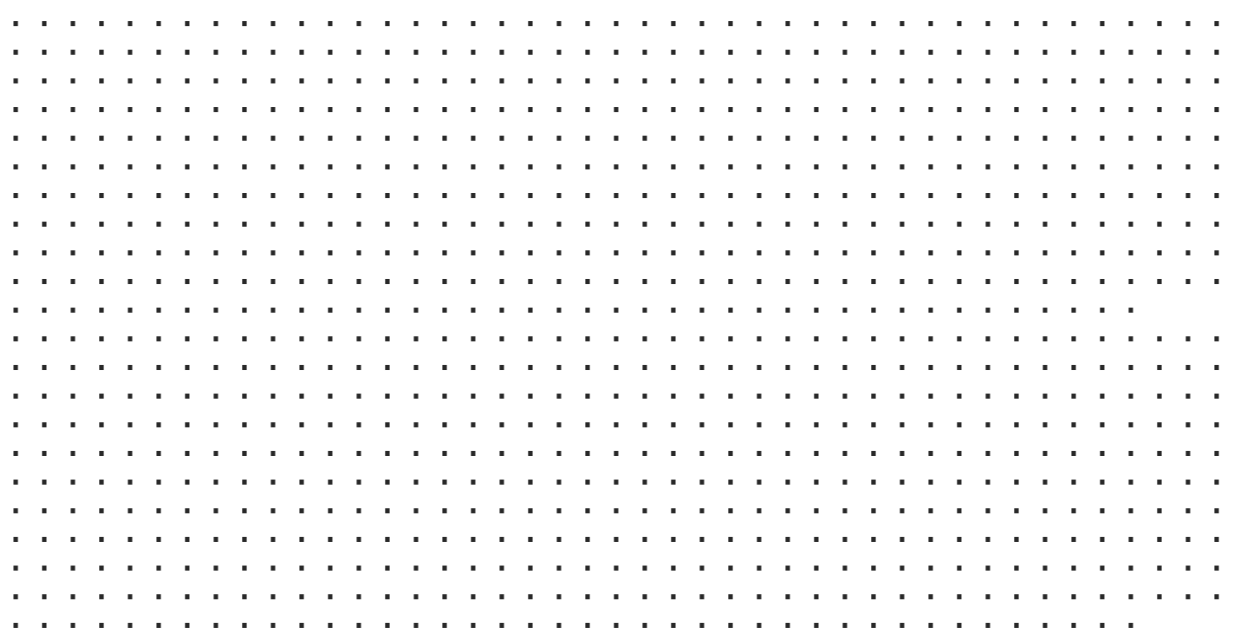
```
    for i , j in zip(range(row ,-1,-1),range(col,N)):
        if board[i][j]=='q':
            return False

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

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

def place_queen(board,N):
    for row in range(N):
        for col in range(N):
            if board[row][col]=='.' and is_valid(board,row,col,N):
                return True
    return False

def queen_problem(N):
    board=[['.' for  _ in range(N)] for _ in range(N)]
    queen=0
    print("Welcome to the Game")
    board_style(board)

    while queen<N:
```

```
        if not place_queen(board,N):
            print("No more moves left... sorry!!!! restarting the game")
            option=input("do you want to change size of
chessboard..?(yes/no/quit):").strip().lower()
            if option=='yes':
                N=int(input("enter the size of chessboard(N): "))
                return queen_problem(N)
            elif option=='quit':
                print("thank you for playing.....")
                return
            else:
                return queen_problem(N)


        try:
            while True:


                row = input(f"Enter row (0-{N-1}) for queen {queen + 1} (or type 'exit'
to quit): ").strip().lower()
                if row == "exit":
                    print("Game ended by user.")
                    return
                col = input(f"Enter col (0-{N-1}) for queen {queen + 1} (or type 'exit'
to quit): ").strip().lower()
                if col == "exit":
                    print("Game ended by user.")
                    return



                row, col = int(row), int(col)
```

```
                if 0 <= row < N and 0 <= col < N:
                    if board[row][col] == '.' and is_valid(board, row, col, N):
                        board[row][col] = 'q'
                        queen += 1
                        board_style(board)
                        break
                    else:
                        print("Invalid move! Queen cannot be placed there. Try again.")
                else:
                    print("Invalid input! Out of bounds. Try again.")
            except ValueError:
                print("Invalid input! Enter numbers only.")

    print("Congratulations! You placed all queens correctly.....You Won!!!!!!!!!")
    play_again()

def play_again():
    choice = input("Do you want to play again? (y/n): ").strip().lower()
    if choice == 'y':
        N = int(input("Enter the size of the chessboard (N): "))
        queen_problem(N)
    else:
        print("Thanks for playing! Goodbye.")
        exit()
N=int(input("enter the size of chessboard(N): "))
queen_problem(N)
```

```
(venv) (base) fajarfatima@Fatima-MacBook PAI_LAB % /Users/fajarfatima/Desktop/PAI_
/bin/python /Users/fajarfatima/Desktop/PAI_LAB/venv/lib/Queen_Problem.py
enter the size of chessboard(N): 4
Welcome to the Game
. . . .
. . . .
. . . .
. . . .

Enter row (0-3) for queen 1 (or type 'exit' to quit): 0
Enter col (0-3) for queen 1 (or type 'exit' to quit): 1
. q . .
. . . .
. . . .
. . . .

Enter row (0-3) for queen 2 (or type 'exit' to quit): 1
Enter col (0-3) for queen 2 (or type 'exit' to quit): 3
. q . .
. . . q
. . . .
. . . .


Enter row (0-3) for queen 3 (or type 'exit' to quit): 2
Enter col (0-3) for queen 3 (or type 'exit' to quit): 0
. q . .
. . . q
q . . .
. . . .

Enter row (0-3) for queen 4 (or type 'exit' to quit): 3
Enter col (0-3) for queen 4 (or type 'exit' to quit): 2
. q . .
. . . q
q . . .
. . q .

Congratulations! You placed all queens correctly.....You Won!!!!!!!!!
Do you want to play again? (y/n):
```

```
Do you want to play again? (y/n): y
Enter the size of the chessboard (N): 500
```



```
Enter row (0-499) for queen 1 (or type 'exit' to quit): exit
Game ended by user.
(venv) (base) fajarfatima@fatima-MacBook PAI_LAB %
```