PAI LAB
BS in Artificial Intelligence

Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Sr.#	Reg.#	Student Name
1.	SU92-BSAIM-F23-107	Ayesha Faisal

Table of Contents

Table of Figures

N-Queen Problem:

1 INTRODUCTION:

Python Code Solves the N-Queens problem for any given board size n.

- Places n queens on an n x n chessboard so that no two queens attack each other.
- Uses a backtracking algorithm to explore valid queen placements row by row.
- Checks for conflicts efficiently using column and diagonal tracking.
- Returns one valid solution showing the column positions of queens in each row.
- If no solution exists, returns -1.

2 CODE IMPLEMENTATION:

```python
def isSafe(row,col, n,cols, diag1,diag2):
    return not cols[col] and not diag1[row + col] and not diag2[row -col + n - 1]

def placeQueens(row, mat, n, cols, diag1, diag2):
    if row == n:
        return True

    for col in range(n):
        if isSafe(row, col, n,cols, diag1,diag2):
            mat[row][col] = 1
            cols[col] =True
            diag1[row+ col] = True
            diag2[row -col + n - 1] = True

            if placeQueens(row + 1, mat, n, cols, diag1,diag2):
                return True

            mat[row][col] = 0
            cols[col] =False
            diag1[row+ col] = False
            diag2[row -col + n - 1]  =False

    return False

def nQueen(n):
    mat =[[0]*n for _ in range(n)]
    cols= [False]*n
    diag1 = [False]*(2*n - 1)
    diag2 =[False]*(2*n - 1)
```


Figure a code-1

```
    diag2 =[False]*(2*n - 1)

    if placeQueens(0, mat, n,cols, diag1,diag2):
        ans =[]
        for i in range(n):
            for j in range(n):
                if mat[i][j]:
                    ans.append(j + 1)
        return ans
    else:
        return [-1]

if __name__== "__main__":
    n = int(input("Enter the size of the board (number of queens): "))
    ans= nQueen(n)
    print(" ".join(map(str, ans)))
```

✓ 0.0s

```
1 3 5 2 4
```

Figure b code-2