

Exp. No: 1

N-Queens Problems.

Date:

Aim:

To solve the N-Queens problem where the goal is to place the N queens on a $n \times n$ chessboard such that no two queens attack each other.

Algorithm:

Step 01: Start

Step 02: Create a $n \times n$ chessboard with all cells set to 0, representing no queens placed.

Step 03: Ensure no queen is the same row, upper diagonal or lower diagonal for a given position.

Step 04: Try placing a queen in each row of current column & if it is safe using is safe

Step 05: Move to the next column if placing a queen works, else backtrack by removing queen

Step 06: If queens are placed in all columns return success

Step 07: Display the board.

Step 08: If no solution exists, print solution does not exist

Program:

```
def isSafe(board, row, col, n):  
    for i in range(col):  
        if board[row][i] == 1:  
            return False  
    for i, j in zip(range(row-1, -1, -1), range(col-1, -1, -1)):  
        if board[i][j] == 1:  
            return False  
    for i, j in zip(range(row+1, n, 1), range(col-1, -1, -1)):  
        if board[i][j] == 1:  
            return False  
    return True
```



```
def solveNQueens(board, col, n):
```

```
    if col >= n:
```

```
        return True
```

```
    for i in range(n):
```

```
        if isSafe(board, i, col, n):
```

```
            board[i][col] = 1
```

```
            if solveNQueens(board, col+1, n) == True:
```

```
                return True
```

```
            board[i][col] = 0
```

```
    return False
```

```
def solveNQueens(n):
```

```
    board = [0]*n for i in range(n):
```

```
        if solveNQueens(board, 0, n) == False:
```

```
            print("Solution does not exist")
```

```
        return False
```

```
    for i in board:
```

```
        print(i)
```

```
    return True
```

```
n = int(input("Enter n value: "))
```

```
solveNQueens(n)
```

enter n value: 5

[1,0,0,0,0] : (1,0,0,0,0) board

[0,0,0,1,0] : (0,0,0,1,0) board

[0,1,0,0,0] : (0,1,0,0,0) board

[0,0,0,0,1] : (0,0,0,0,1) board

Result: [0,0,0,1,0] board

Result:

Thus the n-queens problem's program is executed & the output is verified successfully.