

Lap No. 1

Topic:

N-Queens Problem

Aim:

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

Algorithm:

Step 1: Start

Step 2: Create a  $n \times n$  chessboard with all cells set to 0.

Step 3: Ensure no queen is in the same row, upper diagonal or lower diagonal.

Step 4: Try placing a queen in each row of the current column.

Step 5: Move to next column if placing a queen works.

Step 6: If queen all placed in all columns return success.

~~Step 7: Display the board.~~

Step 8: If no solution, print "solution does not exist".

Program:

```
def isafe (b, a, c, n):
    for i in range(c):
        if b[a][i] == 1:
            return False
    for i, j in zip(range(a, -1, -1),
                    range(c, -1, -1)):
        if b[i][j] == 1:
            return False
    for i, j in zip(range(a, 1, -1), range(c, -1, -1)):
        if b[i][j] == 1:
            return False
    return True
```

```
def solve Nautil (b, c, n):
    if c >= n:
        return True
    for i in range(n):
        if isafe (b, i, c, n):
            b[i][c] = 1
            if solve Nautil (b, c+1, n) == True:
                return True
            b[i][c] = 0
```

return False

```
def solve Na(n):
```

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

```
    if solve Nautil (b, 0, n) == False:
```

```
        print ("sol. does not exist")
```

```
        return
```

```
    return False
```

```
    for i in board
```

```
        print(i)
```



```

return true
n = int(input("Enter value n").
solve W2(n)

```

Output:

```

Enter n value : 5
[1, 0, 0, 0, 0]
[0, 0, 0, 1, 0]
[0, 1, 0, 0, 0]
[0, 0, 0, 0, 1]
[0, 0, 1, 0, 0]

```

Result:

Thus the program is successfully executed & verified.