Exp No : 1

1. N- Queens Problem using Backtracking in python.

Aim :- Write a program to Solve N- Queens problem with python code.

Algorithm :

i) Start

ii) Placing empty space - Create an N×N board filled.

iii) Call check, if placing Queen is safe.
No Queen, is in the Same row.
No Other is in the Same diagonal.
No other queen is in the Same lower diagonal.

iv) If a Safe position is found, place the queen ('1') and move to next column. using recursion.

v) If all Queen are Successfully placed the algorithm, return 'True', indicating that the Solution is found.

vi) If placing the Queen is the current column does not lead to Solution Algorithm Backtracks to the previous column and tries next row.

Program:

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

    for ij in zip(range (row, n, -1) range
                  (col, -1, -1)):
        if board[i][j]== 1:
            return True

def solve NQ util (board, col, n):
    if col >=n:
        return True

    for i in range (n):
        if issafe (board, i, col, n):
            board [i] [col]=1
            if solve NQ util (board, col +1, n)== True
                return True
            board [i] [col]=0
    return False

def solve NQ (n):
    board = [[0]* n for _in range (n)]
    if solve NQ util (board, o, n)==false:
        print ("Solution does not exists")
    return false
```

```
for i in board :
    print [i]
return True
n = int (input ("enter n value : ")
Solve N8 (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 :