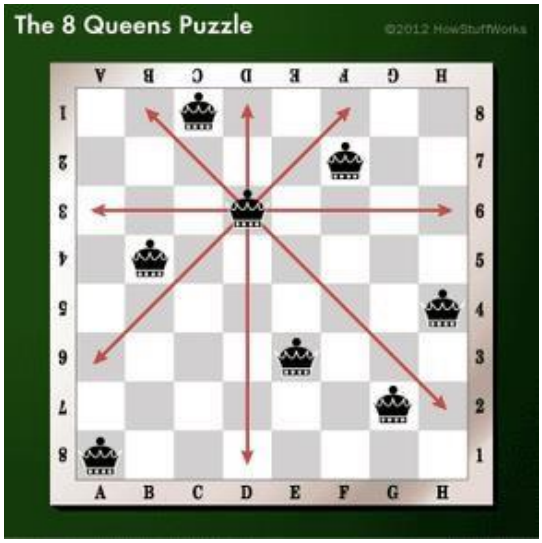
EX.NO: 01

8- QUEENS PROBLEM

AIM :

To implement an 8-Queesns problem using Python.

You are given an 8x8 board; find a way to place 8 queens such that no queen can attack any other queen on the chessboard. A queen can only be attacked if it lies on the same row, same column, or the same diagonal as any other queen. Print all the possible configurations.
To solve this problem, we will make use of the Backtracking algorithm. The backtracking algorithm, in general checks all possible configurations and test whether the required result is obtained or not. For the given problem, we will explore all possible positions the queens can be relatively placed at. The solution will be correct when the number of placed queens = 8.



CODE :

```
def isSafe(mat, r, c):
    # Check column
    for i in range(r):
        if mat[i][c] == 'Q':
            return False

    # Check upper left diagonal
    i, j = r, c
    while i >= 0 and j >= 0:
        if mat[i][j] == 'Q':
            return False
        i -= 1
        j -= 1
```

```
    # Check upper right diagonal
    i, j = r, c
    while i >= 0 and j < len(mat):
        if mat[i][j] == 'Q':
            return False
        i -= 1
        j += 1

    return True


def printSolution(mat):
    for r in mat:
        print(str(r).replace(',', '').replace('\'', ''))
    print()


def nQueen(mat, r):
    if r == len(mat):  # All queens are placed
        printSolution(mat)
        return

    # Try all columns for the current row 'r'
    for i in range(len(mat)):
        if isSafe(mat, r, i):  # Check if the queen can be placed
            mat[r][i] = 'Q'  # Place the queen
            nQueen(mat, r + 1)  # Recur for the next row
            mat[r][i] = '-'  # Backtrack


if __name__ == '__main__':
    N = int(input("Enter the number of Queens: "))
    mat = [['-' for _ in range(N)] for _ in range(N)]  # Create an empty board
    nQueen(mat, 0)  # Start placing queens from row 0
```

OUTPUT :

```
Enter the number of Queens: 4
[- Q - -]
[- - - Q]
[Q - - -]
[- - Q -]

[- - Q -]
[Q - - -]
[- - - Q]
[- Q - -]
```

RESULT :

Thus, the implementing of 8-Queens problem is successfully executed and the output is verified.