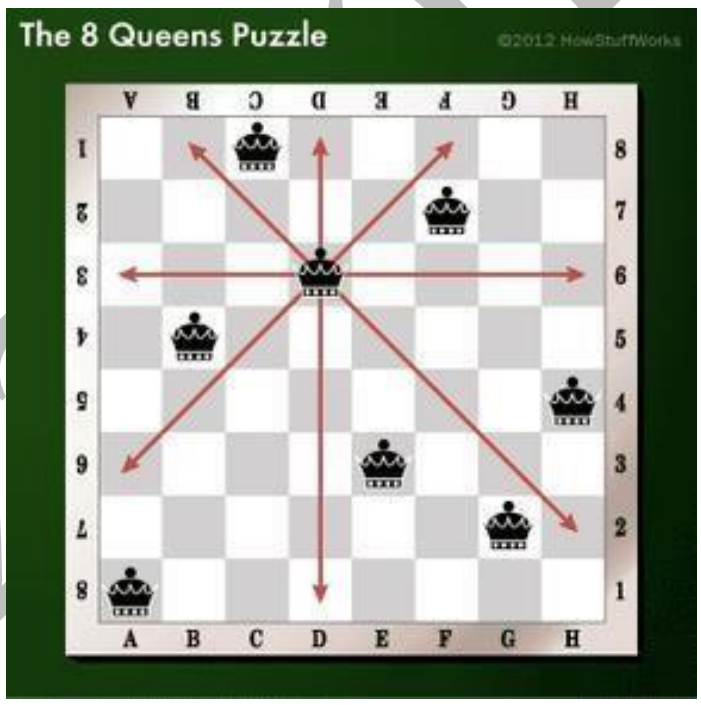
EX.NO:1 **DATE:6/9/2024**
Reg.no:220701060

8-QUEENS PROBLEM

AIM:
To implement an 8-Queesns problem using Python.

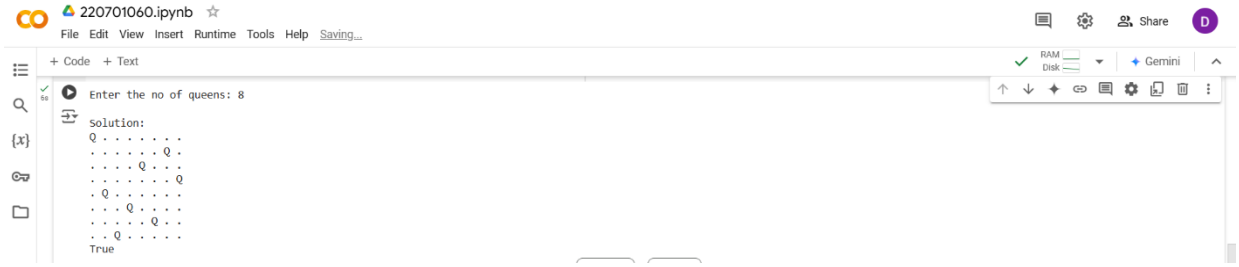
You are given an 8x8 board; find a way to place 8 queens such that no queen can attack any other queen on the chessboard. A queen can only be attacked if it lies on the same row, same column, or the same diagonal as any other queen. Print all the possible configurations.
To solve this problem, we will 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 solve_n_queens_util(board, col, n):
    if col >= n:
        return True
    for i in range(n):
        if is_safe(board, i, col, n):
            board[i][col] = 1
            if solve_n_queens_util(board, col + 1, n):
                return True
            board[i][col] = 0
    return False
def print_board(board, n):
    print("\nSolution:")
    for row in board:
        for cell in row:
            if cell == 1:
                print('Q', end=' ')
            else:
                print('.', end=' ')
        print()
def solve_n_queens(n):
    board = [[0 for _ in range(n)] for _ in range(n)]
    if not solve_n_queens_util(board, 0, n):
        print("No solution exists")
        return False
    print_board(board, n)
    return True
n = int(input("Enter the no of queens: "))
solve_n_queens(n)
```

OUTPUT:



RESULT:

Thus, the 8-Queens program has been implemented successfully.