NAME : RASAVE PRALHAD MAROTI
ROLL NO : 55

ASS NO : 4

```
import java.util.Arrays;


public class NQueensProblem {
    private int[] solution;
    private int boardSize;
    private boolean[] attackedColumns;
    private boolean[] attackedDiagonals1;
    private boolean[] attackedDiagonals2;
    private int solutionsCount;

    public NQueensProblem(int n) {
        boardSize = n;
        solution = new int[boardSize];
        attackedColumns = new boolean[boardSize];
        attackedDiagonals1 = new boolean[2 * boardSize - 1];
        attackedDiagonals2 = new boolean[2 * boardSize - 1];
        solutionsCount = 0;
    }

    public void solve() {
        placeQueen(0);
        System.out.println("Total solutions: " + solutionsCount);
    }

    private void placeQueen(int row) {
        if (row == boardSize) {
            solutionsCount++;
```

```
            System.out.println("Solution " + solutionsCount + ": " + Arrays.toString(solution));
            return;
        }

        for (int col = 0; col < boardSize; col++) {
            if (!isAttacked(row, col)) {
                placeQueenAt(row, col);
                placeQueen(row + 1);
                removeQueenAt(row, col);
            }
        }
    }

    private boolean isAttacked(int row, int col) {
        return attackedColumns[col] || attackedDiagonals1[row + col] || attackedDiagonals2[row - col + boardSize - 1];
    }

    private void placeQueenAt(int row, int col) {
        solution[row] = col;
        attackedColumns[col] = true;
        attackedDiagonals1[row + col] = true;
        attackedDiagonals2[row - col + boardSize - 1] = true;
    }

    private void removeQueenAt(int row, int col) {
        attackedColumns[col] = false;
        attackedDiagonals1[row + col] = false;
        attackedDiagonals2[row - col + boardSize - 1] = false;
    }
```

```
    public static void main(String[] args) {
        int n = 8; // Change this to the desired board size
        NQueensProblem nQueens = new NQueensProblem(n);
        nQueens.solve();
    }
}
```

OUTPUT :

Solution 1: [0, 4, 7, 5, 2, 6, 1, 3]

Solution 2: [0, 5, 7, 2, 6, 3, 1, 4]

Solution 3: [0, 6, 3, 5, 7, 1, 4, 2]

Solution 4: [0, 6, 4, 7, 1, 3, 5, 2]

Solution 5: [1, 3, 5, 7, 2, 0, 6, 4]

Solution 6: [1, 4, 6, 0, 2, 7, 5, 3]

Solution 7: [1, 4, 6, 3, 0, 7, 5, 2]

Solution 8: [1, 5, 0, 6, 3, 7, 2, 4]

Solution 9: [1, 5, 7, 2, 0, 3, 6, 4]

Solution 10: [1, 6, 2, 5, 7, 4, 0, 3]

Solution 11: [1, 6, 4, 7, 0, 3, 5, 2]

Solution 12: [1, 7, 5, 0, 2, 4, 6, 3]

Solution 13: [2, 0, 6, 4, 7, 1, 3, 5]

Solution 14: [2, 4, 1, 7, 0, 6, 3, 5]

Solution 15: [2, 4, 1, 7, 5, 3, 6, 0]

Solution 16: [2, 4, 6, 0, 3, 1, 7, 5]

Solution 17: [2, 4, 7, 3, 0, 6, 1, 5]

Solution 18: [2, 5, 1, 4, 7, 0, 6, 3]

Solution 19: [2, 5, 1, 6, 0, 3, 7, 4]

Solution 20: [2, 5, 1, 6, 4, 0, 7, 3]

Solution 21: [2, 5, 3, 0, 7, 4, 6, 1]

Solution 22: [2, 5, 3, 1, 7, 4, 6, 0]

Solution 23: [2, 5, 7, 0, 3, 6, 4, 1]

Solution 24: [2, 5, 7, 0, 4, 6, 1, 3]

Solution 25: [2, 5, 7, 1, 3, 0, 6, 4]

Solution 26: [2, 6, 1, 7, 4, 0, 3, 5]

Solution 27: [2, 6, 1, 7, 5, 3, 0, 4]

Solution 28: [2, 7, 3, 6, 0, 5, 1, 4]

Solution 29: [3, 0, 4, 7, 1, 6, 2, 5]

Solution 30: [3, 0, 4, 7, 5, 2, 6, 1]

Solution 31: [3, 1, 4, 7, 5, 0, 2, 6]

Solution 32: [3, 1, 6, 2, 5, 7, 0, 4]

Solution 33: [3, 1, 6, 2, 5, 7, 4, 0]

Solution 34: [3, 1, 6, 4, 0, 7, 5, 2]

Solution 35: [3, 1, 7, 4, 6, 0, 2, 5]

Solution 36: [3, 1, 7, 5, 0, 2, 4, 6]

Solution 37: [3, 5, 0, 4, 1, 7, 2, 6]

Solution 38: [3, 5, 7, 1, 6, 0, 2, 4]

Solution 39: [3, 5, 7, 2, 0, 6, 4, 1]

Solution 40: [3, 6, 0, 7, 4, 1, 5, 2]

Solution 41: [3, 6, 2, 7, 1, 4, 0, 5]

Solution 42: [3, 6, 4, 1, 5, 0, 2, 7]

Solution 43: [3, 6, 4, 2, 0, 5, 7, 1]

Solution 44: [3, 7, 0, 2, 5, 1, 6, 4]

Solution 45: [3, 7, 0, 4, 6, 1, 5, 2]

Solution 46: [3, 7, 4, 2, 0, 6, 1, 5]

Solution 47: [4, 0, 3, 5, 7, 1, 6, 2]

Solution 48: [4, 0, 7, 3, 1, 6, 2, 5]

Solution 49: [4, 0, 7, 5, 2, 6, 1, 3]

Solution 50: [4, 1, 3, 5, 7, 2, 0, 6]

Solution 51: [4, 1, 3, 6, 2, 7, 5, 0]

Solution 52: [4, 1, 5, 0, 6, 3, 7, 2]

Solution 53: [4, 1, 7, 0, 3, 6, 2, 5]

Solution 54: [4, 2, 0, 5, 7, 1, 3, 6]

Solution 55: [4, 2, 0, 6, 1, 7, 5, 3]

Solution 56: [4, 2, 7, 3, 6, 0, 5, 1]

Solution 57: [4, 6, 0, 2, 7, 5, 3, 1]

Solution 58: [4, 6, 0, 3, 1, 7, 5, 2]

Solution 59: [4, 6, 1, 3, 7, 0, 2, 5]

Solution 60: [4, 6, 1, 5, 2, 0, 3, 7]

Solution 61: [4, 6, 1, 5, 2, 0, 7, 3]

Solution 62: [4, 6, 3, 0, 2, 7, 5, 1]

Solution 63: [4, 7, 3, 0, 2, 5, 1, 6]

Solution 64: [4, 7, 3, 0, 6, 1, 5, 2]

Solution 65: [5, 0, 4, 1, 7, 2, 6, 3]

Solution 66: [5, 1, 6, 0, 2, 4, 7, 3]

Solution 67: [5, 1, 6, 0, 3, 7, 4, 2]

Solution 68: [5, 2, 0, 6, 4, 7, 1, 3]

Solution 69: [5, 2, 0, 7, 3, 1, 6, 4]

Solution 70: [5, 2, 0, 7, 4, 1, 3, 6]

Solution 71: [5, 2, 4, 6, 0, 3, 1, 7]

Solution 72: [5, 2, 4, 7, 0, 3, 1, 6]

Solution 73: [5, 2, 6, 1, 3, 7, 0, 4]

Solution 74: [5, 2, 6, 1, 7, 4, 0, 3]

Solution 75: [5, 2, 6, 3, 0, 7, 1, 4]

Solution 76: [5, 3, 0, 4, 7, 1, 6, 2]

Solution 77: [5, 3, 1, 7, 4, 6, 0, 2]

Solution 78: [5, 3, 6, 0, 2, 4, 1, 7]

Solution 79: [5, 3, 6, 0, 7, 1, 4, 2]

Solution 80: [5, 7, 1, 3, 0, 6, 4, 2]

Solution 81: [6, 0, 2, 7, 5, 3, 1, 4]

Solution 82: [6, 1, 3, 0, 7, 4, 2, 5]

Solution 83: [6, 1, 5, 2, 0, 3, 7, 4]

Solution 84: [6, 2, 0, 5, 7, 4, 1, 3]

Solution 85: [6, 2, 7, 1, 4, 0, 5, 3]

Solution 86: [6, 3, 1, 4, 7, 0, 2, 5]

Solution 87: [6, 3, 1, 7, 5, 0, 2, 4]

Solution 88: [6, 4, 2, 0, 5, 7, 1, 3]

Solution 89: [7, 1, 3, 0, 6, 4, 2, 5]

Solution 90: [7, 1, 4, 2, 0, 6, 3, 5]

Solution 91: [7, 2, 0, 5, 1, 4, 6, 3]

Solution 92: [7, 3, 0, 2, 5, 1, 6, 4]

Total solutions: 92

=== Code Execution Successful ===