**NAME :Gullay Zahra**

**Roll number:44**

1. ****Queen Problem****

## Introduction

The N-Queen problem is a well-known challenge in computer science and mathematics. It requires placing N queens on an N×N chessboard in such a way that no two queens can attack each other. This means ensuring that no two queens share the same row, column, or diagonal. One of the most efficient approaches to solving this problem is backtracking, which systematically explores possible solutions while eliminating invalid placements.

## How the Program Works

The given Python program uses backtracking to solve the N-Queen problem. Below is an overview of its key components:

### 1. ****Checking for a Safe Placement (****is\_safe ****function)****

Before placing a queen in any position, this function checks:

* If there is already a queen in the same row.
* If there is a queen on the upper-left diagonal.
* If there is a queen on the lower-left diagonal. If any of these conditions are met, the placement is considered unsafe.

### 2. ****Placing Queens Recursively (****solve ****function)****

* The function places queens one column at a time.
* If a queen can be placed safely, it moves to the next column.
* If an invalid placement is found, the function backtracks and tries the next possible row.
* This process continues until all queens are successfully placed.

### 3. ****Main Execution (****Nqueen ****function)****

* Initializes an empty chessboard as a 2D list.
* Calls the recursive function to place queens.
* If a solution is found, it prints the chessboard.
* If no solution exists, it displays an appropriate message.

## Conclusion

The backtracking algorithm effectively finds a solution to the N-Queen problem by systematically exploring different placements and reverting (backtracking) when an invalid placement is encountered. This ensures that a valid arrangement is found for a given board size or confirms that no solution exists.

**Output:**

