

**The Superior University**

***Department of Software Engineering***

***Faculty of Computer Science & Information Technology***

***The Superior University, Lahore***

***Course: Programming For Artificial Intelligence***

***Course Instructor:Sir Rasikh ali***

***Semester 4 BSAI(4A)***

***Fall 2025***

***Name:Muhammad Reshail***

***Roll no:053***

***LAB TASK 4***



This function prints the current state of the chessboard. Each row is joined into a string with spaces between the elements, and an empty line is printed after the board for readability.



This part of the is\_safe function checks the column above the current position (row, col) to ensure no queen is already placed in the same column.

This checks the upper-left diagonal for any queens. It iterates from the current position (row, col) upwards and to the left.

This checks the upper-right diagonal for any queens. It iterates from the current position (row, col) upwards and to the right. If no conflicts are found, the position is safe, and the function returns True.



This is the base case for the recursion. If all rows have been processed (row == n), it means a valid solution has been found, and the board is printed.

This loop iterates through each column in the current row. If placing a queen at (row, col) is safe, the queen is placed ('Q').

This function initializes an N x N chessboard with empty spaces ('.') and starts the solving process by calling solve\_n\_queens with the first row (row = 0).The user is prompted to input the number of queens (N). The n\_queens function is called with this value to solve the N-Queens problem.

