**N-Queens Problem Solver Documentation**

**Overview**

This Python program provides a solution to the N-Queens problem using backtracking. The program allows users to generate and visualize solutions or count the number of valid arrangements for a given board size.

**Functions**

**1. n\_queens\_solver(board\_size)**

**Description:** Finds all valid arrangements of N queens on an N×N chessboard.

**Parameters:**

* board\_size (int): The size of the chessboard (N).

**Returns:**

* solutions (list of lists): A list containing all possible valid placements of queens, where each inner list represents a board configuration with column indices of queens for each row.

**Internal Functions:**

* is\_position\_safe(board, row, col): Checks if placing a queen at (row, col) is safe.
* explore\_solutions(board, col): Recursively explores all valid board configurations.

**2. show\_solutions(solutions, board\_size)**

**Description:** Prints the board representation of valid solutions.

**Parameters:**

* solutions (list of lists): The list of valid board configurations.
* board\_size (int): The size of the chessboard.

**Output:**

* Prints each board configuration with 'Q' representing a queen and '.' for empty spaces.

**3. count\_solutions(board\_size)**

**Description:** Counts the number of valid N-Queens arrangements without explicitly listing them.

**Parameters:**

* board\_size (int): The size of the chessboard.

**Returns:**

* count (int): The total number of valid arrangements.

**Internal Functions:**

* is\_safe(queens, row, col): Determines if a queen can be placed at (row, col) without conflicts.
* explore\_count(queens, row): Recursively explores possible placements to count solutions.

**4. get\_positive\_integer(prompt)**

**Description:** Prompts the user to enter a positive integer and handles input validation.

**Parameters:**

* prompt (str): The message displayed to the user.

**Returns:**

* num (int): A valid positive integer entered by the user.

**Main Execution**

When the script is run directly, it prompts the user to:

1. Enter the chessboard size (N).
2. Choose to either:
   * Display solutions (S)
   * Count solutions (C)
   * Do both (B)

Based on the user’s choice, the program executes the corresponding functions to solve and/or count solutions to the N-Queens problem.

**Example Usage**

Enter the chessboard size (N): 4

Do you want to see a solution (S), count solutions (C), or both (B)? B

Total possible arrangements for 4-Queens: 2

. Q . .

. . . Q

Q . . .

. . Q .

. . Q .

Q . . .

. . . Q

. Q . .

Number of solutions for 4-Queens problem: 2

