Sudoku Skyscraper - Function Calls

This list shows the order in which the functions are called to solve the Sudoku. The process follows the recursive backtracking approach.

Chronological Order of Function Calls:

1. main
2. copy_matrix
3. solve
4. recursion
5. next_position
6. is_available
7. column_available
8. row_available
9. box_available
10. reverse_recursion
11. reverse_next_position
12. compare_solutions
13. print_sudoku

Functions and Their Explanations:

main:

This is the program's entry point. It checks the input and prepares the data for solving the Sudoku.
If the input is valid, it calls the 'solve' function.
The main function is the entry point of the program.
It verifies if the correct number of arguments is provided. Then, it copies the input into two separate 9x9 matrices (tab1, tab2) and calls the 'solve' function to solve the Sudoku.

copy_matrix:

This function copies the entered Sudoku data into a 9x9 matrix. It ensures that the Sudoku grid is read correctly.

This function copies the input data (coming from command-line arguments) into a 9x9 Sudoku matrix.
It also verifies if the input is valid (9 characters per row, only allowed characters like numbers or '.').

solve:

The main function to solve the Sudoku. It calls 'recursion' to attempt a forward solution.
Then, it calls 'reverse_recursion' to check if the solution is unique.

This function is responsible for solving the Sudoku. It first tries a normal solution using the 'recursion' function.
Then, it checks with 'reverse_recursion' if the solution is unique. If the solutions do not match, an error is displayed.

recursion:

This function tries to solve the Sudoku using backtracking. It searches for an empty position, tests numbers from 1 to 9, and calls itself recursively until a solution is found.

This function implements recursive backtracking to solve the Sudoku. It looks for the next free cell using 'next_position' and tests all numbers from '1' to '9'.
If a number fits ('is_available'), it is placed, and the function calls itself. If no solution is found, the cell is reset (backtracking).

next_position:

Finds the next empty position in the Sudoku (from top left to bottom right). Used by 'recursion'.

Finds the next empty position ('.') in the Sudoku by scanning from left to right, top to bottom. This function is essential for the recursive solving process.

is_available:

Checks if a number can be inserted at a specific position. Calls 'column_available', 'row_available',

and 'box_available' to verify.

This function checks whether a number is allowed at a specific position. It calls three helper functions: 'column_available' (column check), 'row_available' (row check), and 'box_available' (3x3 block check).

column_available:

Checks if a number already exists in the current column.

Checks if a specific number is already in the column. If so, the function returns '0' (not allowed); otherwise, it returns '1'. It scans the column from top to bottom.

row_available:

Checks if a number already exists in the current row.

Checks if a specific number is already in the row. If so, the function returns '0' (not allowed); otherwise, it returns '1'. It scans the row from left to right.

box_available:

Checks if a number already exists in the 3x3 block.

This function checks if a number is already present in the 3x3 block. The coordinates of the current cell are used to find the corresponding block, which is then scanned.

reverse_recursion:

Solves the Sudoku from bottom right to top left to check if a unique solution exists.

Works similarly to 'recursion', but processes the Sudoku in reverse order (from bottom right to top left). This helps determine if the Sudoku has a unique solution.

reverse_next_position:

Finds the next empty position from bottom right to top left.

Finds the next empty position ('.'), but in reverse order, scanning from bottom right to top left. This function is used in 'reverse_recursion'.

compare_solutions:

Compares the forward and backward solved Sudoku grids to see if they match.

Compares two Sudoku solutions (one solved forward and one solved backward). If both match, the solution is unique. If not, an error is displayed.

print_sudoku:

Prints the solved Sudoku in the correct format on the screen.

This function outputs the solved Sudoku to the console. It iterates through the 9x9 grid and prints it in a readable format.