1. Includes and definitions

The required header files, string.h, stdlib.h, and stdio.h, are included in the code.

GRID_SIZE 3 defines the puzzle grid's dimensions.

The variables MOVE_UP 0, MOVE_DOWN 1, MOVE_LEFT 2, and MOVE_RIGHT 3 specify the potential motions (up, down, left, and right).

2. Heuristic function

The role of heuristics The Manhattan distance between the present state and the desired state is determined by the heuristic.

The function returns the heuristic value after receiving an array of integer states as input.

The absolute disparities between each tile's x and y coordinates in the current and target states are added up to determine the heuristic value.

3. A* search algorithm

The a_star_search function defines the A* search algorithm.

Initial_state and goal_state are two arrays of numbers that are entered into the function.

The function stores nodes to be investigated in a priority queue.

After creation, the first node is put into the priority queue.

4. Node exploration

Until the desired state is attained, the function investigates nodes.

From the priority queue, the node with the lowest f-score (cost + heuristic) gets removed.

By comparing the current node's state with the goal state, the function determines if the goal state has been attained.

The function traces back the parent nodes to recreate the solution route if the target state is attained.

5. Successor node generation

By applying the possible movements (up, down, left, and right) to the current node, the function creates successor nodes.

The function determines the heuristic value for each successor node and adds it to the priority queue.

6. Main function

The puzzle's starting point and end state are specified in the main function.

In order to use A* search to solve the problem, the function invokes the a_star_search function.

The console displays the path of the solution.