Data Structures

1. **State**:
    - Represents a state in the search space with the following attributes:
        - board: The 3x3 grid representing the current configuration.
        - blank_row and blank_col: The position of the blank (zero) tile.
        - cost: The cost to reach this state from the initial state.
2. **PQNode**:
    - Represents a node in the priority queue with the following attributes:
        - state: Pointer to a State object.
        - priority: The priority value used in the priority queue.
3. **PriorityQueue**:
    - Represents a priority queue with the following attributes:
        - nodes: An array of pointers to PQNode objects.
        - size: The current size of the priority queue.

Functions

1. **initializeState**:
    - Initializes a state with the given board configuration.
    - Identifies the position of the blank tile (0).
2. **isGoalState**:
    - Checks if a given state is the goal state by verifying if the tiles are in ascending order, with the blank tile at the end.
3. **calculateHammingPriority**:
    - Calculates the Hamming priority, which counts the number of tiles not in their goal positions.
4. **calculateManhattanDistance**:
    - Calculates the Manhattan distance of a single tile from its goal position.
5. **calculateManhattanPriority**:
    - Calculates the Manhattan priority, which is the sum of the Manhattan distances of all tiles from their goal positions.
6. **printBoard**:
    - Prints the current configuration of the board.
7. **swap**:

- Swaps two elements in the board.

8. **createPriorityQueue**:
    - Creates and initializes an empty priority queue.
9. **push**:
    - Inserts a new state into the priority queue with the given priority.
10. **pop**:
    - Removes and returns the state with the highest priority from the priority queue.
11. **isEmpty**:
    - Checks if the priority queue is empty.
12. **aStarSearch**:
    - Implements the A* search algorithm to find the shortest path from the initial state to the goal state.
    - Uses the Manhattan priority as the heuristic function.
    - Generates successor states by moving the blank tile in all possible directions and pushes them into the priority queue with the calculated priority.
    - Stops when the goal state is reached or when the priority queue is empty.

Main Function

1. **main**:
    - Initializes the initial board configuration.
    - Checks if the initial state is the goal state.
    - Prints the initial state and its priorities (Hamming and Manhattan).
    - Calls the aStarSearch function to solve the puzzle.
    - Frees the allocated memory for the initial state.