```
# Name: Shantanu Rohile
# Roll No: 53

# ================================================
# AI Practical 03
# Problem Statement:
# Write a program to implement various Sorting Algorithms
# (such as Bubble Sort, Selection Sort, Merge Sort) using
# appropriate Knowledge Representation (like lists, arrays, rules)
# and Reasoning Techniques (like if-else logic, decision making).
# ================================================

# ---------------------------------------------------------------------
# Sorting Algorithms using Procedural Knowledge Representation
# ---------------------------------------------------------------------

# 1. Bubble Sort Algorithm
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n - i - 1):
            if arr[j] > arr[j + 1]:
                # Swapping if elements are in the wrong order
                arr[j], arr[j + 1] = arr[j + 1], arr[j]

# 2. Selection Sort Algorithm
def selection_sort(arr):
    n = len(arr)
    for i in range(n):
        min_index = i
        for j in range(i + 1, n):
            if arr[j] < arr[min_index]:
                min_index = j
        # Swap with the smallest element found
        arr[i], arr[min_index] = arr[min_index], arr[i]

# 3. Merge Sort Algorithm
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr) // 2  # Finding the middle
        left_half = arr[:mid]
        right_half = arr[mid:]

        # Recursive calls to sort both halves
        merge_sort(left_half)
        merge_sort(right_half)

        # Merge the sorted halves
        i = j = k = 0

        # Compare elements from both halves and merge
        while i < len(left_half) and j < len(right_half):
            if left_half[i] < right_half[j]:
                arr[k] = left_half[i]
                i += 1
            else:
```

```
            arr[k] = right_half[j]
            j += 1
        k += 1

    # Copy any remaining elements
    while i < len(left_half):
        arr[k] = left_half[i]
        i += 1
        k += 1

    while j < len(right_half):
        arr[k] = right_half[j]
        j += 1
        k += 1

# ------------------------------------------------------------------
# Example Usage
# ------------------------------------------------------------------
arr = [9, 3, 7, 1, 5]
print("Original Array:", arr)

# User input to choose sorting method
sorting_method = input("Enter sorting method (bubble / selection / merge): ").lower()

# Conditional reasoning to apply correct algorithm
if sorting_method == 'bubble':
    sorted_arr = list(arr)
    bubble_sort(sorted_arr)
    print("Bubble Sorted Array:", sorted_arr)

elif sorting_method == 'selection':
    sorted_arr = list(arr)
    selection_sort(sorted_arr)
    print("Selection Sorted Array:", sorted_arr)

elif sorting_method == 'merge':
    sorted_arr = list(arr)
    merge_sort(sorted_arr)
    print("Merge Sorted Array:", sorted_arr)

else:
    print("Invalid sorting method. Please enter 'bubble', 'selection', or 'merge'.")

# ------------------------------------------------------------------
# Sample Output:
# Original Array: [9, 3, 7, 1, 5]
# Enter sorting method (bubble / selection / merge): merge
# Merge Sorted Array: [1, 3, 5, 7, 9]
# ------------------------------------------------------------------
```