# 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]

*#*