SEM 3\Exp9\main.c

```c
#include <stdio.h>

// Function declarations (you can also include headers for better organization)
void mergeSort(int arr[], int left, int right);
void radixSort(int arr[], int size);
void countingSort(int arr[], int size);
void bucketSort(int arr[], int size);
void heapSort(int arr[], int size);

// Function to display the array (add this in main.c)
void display(int arr[], int size)
{
    for (int i = 0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

int main()
{
    // Array for testing sorting algorithms
    int arr1[] = {74, 34, 25, 12, 22, 11, 90, 65, 32, 1};
    int size1 = sizeof(arr1) / sizeof(arr1[0]);

    // Merge Sort
    printf("Original array for Merge Sort: ");
    display(arr1, size1);
    mergeSort(arr1, 0, size1 - 1);
    printf("Sorted array using Merge Sort: ");
    display(arr1, size1);

    // Radix Sort
    int arr2[] = {74, 34, 25, 12, 22, 11, 90, 65, 32, 1};
    printf("\nOriginal array for Radix Sort: ");
    display(arr2, size1);
    radixSort(arr2, size1);
    printf("Sorted array using Radix Sort: ");
    display(arr2, size1);

    // Counting Sort
    int arr3[] = {74, 34, 25, 12, 22, 11, 90, 65, 32, 1};
    printf("\nOriginal array for Counting Sort: ");
    display(arr3, size1);
    countingSort(arr3, size1);
    printf("Sorted array using Counting Sort: ");
    display(arr3, size1);

    // Bucket Sort
    int arr4[] = {74, 34, 25, 12, 22, 11, 90, 65, 32, 1};
    int size4 = sizeof(arr4) / sizeof(arr4[0]);
    printf("\nOriginal array for Bucket Sort: ");
    display(arr4, size4);
```

```c
    bucketSort(arr4, size4);
    printf("Sorted array using Bucket Sort: ");
    display(arr4, size4);

    // Heap Sort
    int arr5[] = {74, 34, 25, 12, 22, 11, 90, 65, 32, 1};
    int size5 = sizeof(arr5) / sizeof(arr5[0]);
    printf("\nOriginal array for Heap Sort: ");
    display(arr5, size5);
    heapSort(arr5, size5);
    printf("Sorted array using Heap Sort: ");
    display(arr5, size5);

    return 0;
}

```