Date: 21/7/25 to 25/7/25

Experiment No: 01

I. **Aim:** Implement Merge sort for the given list of integer values and find space and time complexity/.

II. Theory:

III. Program:

```
#include <stdio.h>
#include <stdlib.h> // for malloc and free

void merge(int arr[], int left, int mid, int right) {
    int i, j, k;
    int n1 = mid - left + 1;
    int n2 = right - mid;

    int *L = (int *)malloc(n1 * sizeof(int));
    int *R = (int *)malloc(n2 * sizeof(int));

    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];

    i = 0; j = 0; k = left;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j])
            arr[k++] = L[i++];
        else
            arr[k++] = R[j++];
    }

    while (i < n1)
        arr[k++] = L[i++];
    while (j < n2)
        arr[k++] = R[j++];

        free(L);
    free(R);
}

void mergeSort(int arr[], int left, int right) {
    if (left < right) {
```

```
                int mid = (left + right) / 2;

                mergeSort(arr, left, mid);
                mergeSort(arr, mid + 1, right);
                merge(arr, left, mid, right);
            }
        }

        int main() {
            int i;
            int arr[10];
            printf("Enter 10 numbers:\n");
            for (i = 0; i < 10; i++)
                scanf("%d", &arr[i]);

            mergeSort(arr, 0, 9);

            printf("Sorted array:\n");
            for (i = 0; i < 10; i++)
                printf("%d ", arr[i]);
            printf("\n");

            return 0;
}
```

IV. Output:

V. Complexity Analysis: