DSA LAB

Lab Assignment number 18

Name: Aamir Ansari **Batch:** A **Roll no:** 01

Aim: Implement Merge Sort and Quick Sort

Program:

```
#include<stdio.h>

/*Array to store the list*/
int array[1000];

/*Merge Sort*/
void merge(int first, int last)
{
   int mid = (first+last)/2;
   int i = first;
   int j = mid+1;
   int k = first;

   int temp[100];

   while(i<=mid && j<=last)
   {
      if(array[i] < array[j])
      {
         temp[k++] = array[i++];
      }
      else
      {
         temp[k++] = array[j++];
      }
   }

   while(i<=mid)
   {
      temp[k++] = array[i++];
   }

   while(j<=last)
   {
      temp[k++]  = array[j++];
   }

   /*Copy all element to original array*/
   for(int i=first;i<=last;i++)
   {
      array[i] = temp[i];
```

```
    }
}

void merge_sort(int first,int last)
{
    /*Base case - 1 or 0 elements*/
    if(first>=last)
    {
        return;
    }
    int mid = (first+last)/2; /*Divide*/

    /*Recursively the arrays - first,mid and mid+1,last*/
    merge_sort(first,mid);
    merge_sort(mid+1,last);

    /*Merge the two parts*/
    merge(first,last);

}

/*Quick Sort*/
void quick_sort(int first,int last)
{
    int i, j, pivot, temp;

    if(first<last)
    {
        pivot=first;
        i=first;
        j=last;

        while(i<j)
        {
            while(array[i]<=array[pivot]&&i<last)
            {
                i++;
            }
            while(array[j]>array[pivot])
            {
                j--;
            }
            if(i<j)
            {
                temp=array[i];
                array[i]=array[j];
                array[j]=temp;
            }
        }
        temp=array[pivot];
        array[pivot]=array[j];
        array[j]=temp;
```

```
        quick_sort(first,j-1);
        quick_sort(j+1,last);
     }
}

/*Print a sorted array*/
void print_sorted_array(int n)
{
   int i;
   printf("Sorted Array:");
   for(i=0;i<n;i++)
   {
      printf("%d ",array[i]);
   }
}

void main()
{
   int n,i, choice;

   printf("Enter number of elements in the List : ");
   scanf("%d", &n);
   printf("Enter %d integers\n", n);
   for (i = 0; i < n; i++)
   {
      scanf("%d", &array[i]);
   }

   printf("Type of sort to perform:\n1Merge Sort\n2.Quick Sort\n3.Exit");
   printf("\nEnter the choice to be performed: ");
   scanf("%d",&choice);

   switch(choice)
   {
      case 1:
         merge_sort(0,n-1);
         print_sorted_array(n);
         break;

      case 2:
         quick_sort(0,n-1);
         print_sorted_array(n);
         break;

      case 3:
      default:
         printf("Thank You!!");
   }
}
```

Output:

```
Enter number of elements in the List : 6
Enter 6 integers
1
30
75
16
55
28
Type of sort to perform:
1Merge Sort
2.Quick Sort
3.Exit
Enter the choice to be performed: 1
Sorted Array:1 16 28 30 55 75
```

```
Enter number of elements in the List : 5
Enter 5 integers
34
98
65
11
3
Type of sort to perform:
1Merge Sort
2.Quick Sort
3.Exit
Enter the choice to be performed: 2
Sorted Array:3 11 34 65 98
```