Lab-5

1) Sort a given set of N integer elements using the Merge Sort technique and compute its time taken. Run the program for different values of N and record the time taken to sort.

```
#include<stdio.h>

void mergesort(int a[],int i,int j);
void merge(int a[],int i1,int j1,int i2,int j2);

int main()
{
   int a[30],n,i;
   printf("Enter no of elements:");
   scanf("%d",&n);
   printf("Enter array elements:");
   for(i=0;i<n;i++)
   scanf("%d",&a[i]);
   mergesort(a,0,n-1);
   printf("\nSorted array is :");
   for(i=0;i<n;i++)
   printf("%d ",a[i]);
   return 0;
}

void mergesort(int a[],int i,int j)
{
   int mid;
   if(i<j)
   {
      mid=(i+j)/2;
      mergesort(a,i,mid);
      mergesort(a,mid+1,j);
      merge(a,i,mid,mid+1,j);
   }
}

void merge(int a[],int i1,int j1,int i2,int j2)
{
   int temp[50];
   int i,j,k;
   i=i1;
   j=i2;
   k=0;
   while(i<=j1 && j<=j2)
   {
      if(a[i]<a[j])
```

```
        temp[k++]=a[i++];
        else
        temp[k++]=a[j++];
    }
    while(i<=j1)
    temp[k++]=a[i++];
    while(j<=j2)
    temp[k++]=a[j++];
    for(i=i1,j=0;i<=j2;i++,j++)
    a[i]=temp[j];
}
```

OUTPUT

```
Enter no of elements:7
Enter array elements:23 45 21 23 98 67 56

Sorted array is :21 23 23 45 56 67 98
Process returned 0 (0x0)   execution time : 11.775 s
Press any key to continue.
```

```
Enter no of elements:15
Enter array elements:23 87 34 56 12 34 65 78 54 67 15 23 63 20
29

Sorted array is :12 15 20 23 23 29 34 34 54 56 63 65 67 78 87
Process returned 0 (0x0)   execution time : 43.156 s
```

2) Sort a given set of N integer elements using the Quick Sort technique and compute its time taken.

```
#include<stdio.h>

void qsort(int a[], int low, int high)
{
    int mid;
    if(low<high)
    {
        mid=partition(a,low,high);
        qsort(a,low,mid-1);
        qsort(a,mid+1, high);
    }
}
int partition(int a[],int low, int high)
{
    int i,j,temp, pivot;
    pivot=a[low];
    i=low+1;
    j=high;
```

```
    while(i<=j)
    {
        while(a[i]<=pivot)
            i++;
        while(a[j]>pivot)
            j--;
        if(i<j)
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
    }
    temp=a[low];
    a[low]=a[j];
    a[j]=temp;
    return j;
}

int main()
{
    int a[30],n,i;
    printf("Enter no of elements:");
    scanf("%d",&n);
    printf("Enter array elements:");
    for(i=0;i<n;i++)
    scanf("%d",&a[i]);
    qsort(a,0,n-1);
    printf("\nSorted array is :");
    for(i=0;i<n;i++)
    printf("%d ",a[i]);
    return 0;
}
```

OUTPUT

```
Enter no of elements:5
Enter array elements:1 8 3 2 10

Sorted array is :1 2 3 8 10
Process returned 0 (0x0)   execution time : 13.919 s
```

```
Enter no of elements:10
Enter array elements:12 54 17 23 97 90 17 27 45 22

Sorted array is :12 17 17 22 23 27 45 54 90 97
Process returned 0 (0x0)   execution time : 18.328 s
```