ASSIGNMENT REPORT- 2

(*Task : to perform coding of Radix and Merge Sort*)

Submitted by

(Mogalraj Kushal Dath)

Registration Number : 12100559

Course Name : Advanced Data Structure and Algorithms Laboratory

Course Code : CSE505

Section Number : K21ML

Submitted to

(Gurasis Singh Sir)

School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY

Program 1 : Program to perform Radix Sort (code in c++)

```
# include <iostream>
using namespace std;

void print(int Arr[], int n) { for(int i=0;i<n;i++) { cout<<"["<<Arr[i]<<"]"<<ends; }
cout<<endl; }

void radix_sort(int array[],int n)
{
int max = array[0];

for (int i=1; i<n; i++) { if(array[i]>max) { max=array[i]; } }

for(int p=1; max/p>0; p*=10)
{
int output[n];
int count[10];

for (int i=0; i<10; i++) { count[i] = 0; } // initializing count values to zero

for (int i=0; i<n; i++){
count[(array[i] / p)%10]++; }     // Counting the elements and updating count values

for (int i=1; i<10; i++){ count[i] += count[i-1]; }  // Cumulating the count values

for (int i=n-1; i>=0; i--) {
output[count[(array[i] / p)%10] - 1] = array[i]; // Sorting and Placing Elements
count[(array[i] / p)%10]--;  }                  // according to its count value
for (int i=0; i<n; i++){
array[i] = output[i];}
}
}

int main()
{
	int n;
	cout<<"enter the size of the array : "; cin>>n;
	int arr[n];
	cout<<"enter the elements in the array : "; for(int i=0;i<n;i++) { cin>>arr[i]; }
	cout<<"Array Elements Before Radix Sort arr["<<n<<"] : "; print(arr,n);
	radix_sort(arr,n);
	cout<<"Array Elements After Radix Sort arr["<<n<<"] : "; print(arr,n);
}
```

Program 2 : Program to perform Merge Sort (code in c++)

```
# include <iostream>
using namespace std;

void Merge_Sub_Array(int a[], int low, int mid, int high)
{
	// We have low to mid and mid+1 to high already sorted
	int i = low, j = mid + 1, k = 0, temp[high-low+1];



	// Merge the two parts into temp[], Sorting at the same time.
	while (i<=mid && j<=high)
	{
        if (a[i]<a[j])
        {
                temp[k]=a[i];
                k++;
                i++;
        }
        else
        {
                temp[k]=a[j];
                k++;
                j++;
        }
}

// Insert all the remaining values from i->mid into temp[] Array
while(i<=mid)
{
        temp[k] = a[i];
        k++;
        i++;
}

// Insert all the remaining values from j->high into temp[] Array
while(j<=high)
{
        temp[k]=a[j];
        k++;
        j++;
}
```

```
    // Assigning sorted data which is stored in temp[] to a[]
    for (i=low;i<=high;i++)
    {
        a[i] = temp[i-low];
    }
}

void Merge_Sort(int Arr[], int low, int high)
{
    if(low >= high){ return; }
    int mid = low+(high-low)/2;
    Merge_Sort(Arr, low, mid);
    Merge_Sort(Arr, mid+1, high);
    Merge_Sub_Array(Arr, low, mid, high);
}

void print(int Arr[], int n) { for(int i=0;i<n;i++) { cout<<"["<<Arr[i]<<"]"<<ends; }
cout<<endl; }

int main()
{
    int n;
    cout<<"enter the size of the array : "; cin>>n;
    int Arr[n];
    cout<<"enter the elements in the array : "; for(int i=0;i<n;i++) { cin>>Arr[i]; }
    cout<<"\nArray Elements Before Merge Sort Arr["<<n<<"] = "; print(Arr,n);
    Merge_Sort(Arr,0,n-1);
    cout<<endl<<"Array Elements After Merge Sort Arr["<<n<<"] = "; print(Arr,n);
}
```