**DAA PRACTICAL**

**Name: Virendra Kashinath Bagul Roll No.:05**

**MERGE SORT PROGRAM**

#include<iostream>

using namespace std;

#define max 100 // Macro definition for maximum size of array

void merge\_sort(int arr[],int low,int up);

void merge\_s(int arr[],int temp[],int low1,int up1,int low2,int up2);

void copy\_s(int arr[],int temp[],int low,int up);

int main()

{

int i,n,arr[max];

cout<<"enter the size of array:"<<endl;

cin>>n;

cout<<"enter array elements "<<endl;

for(i=0;i<n;i++)

{

cin>>arr[i];

}

merge\_sort(arr,0,n-1); // Sorting array using merge sort

cout<<"sorted list is "<<endl;

for(i=0;i<n;i++)

cout<<arr[i]<<" ";

return 0;

}

void merge\_sort(int arr[],int low,int up) // Merge sort function

{

int mid;

int temp[max]; // Temporary array for merging

if(low<up)

{

mid=(low+up)/2;

merge\_sort(arr,low,mid); //left sublist // Recursive calls to divide array into sublists

merge\_sort(arr,mid+1,up); //right sublist

merge\_s(arr,temp,low,mid,mid+1,up);

copy\_s(arr,temp,low,up); // Copying the merged sublist back to original array

}

}

void merge\_s(int arr[],int temp[],int low1,int up1,int low2,int up2)

{

int i=low1;

int j=low2;

int k=low1;

while((i<=up1)&&(j<=up2)) // Merging the sublists

{

if(arr[i]<=arr[j])

temp[k++]=arr[i++];

else

temp[k++]=arr[j++];

}

while(i<=up1) // Copying remaining elements of left sublist if any

temp[k++]=arr[i++];

while(j<=up2)

temp[k++]=arr[j++];

}

void copy\_s(int arr[],int temp[],int low,int up)

// Function to copy elements from temporary array to original array

{

int i;

for(i=low;i<=up;i++) // Copying elements from temp array to original array

arr[i]=temp[i];

}

//OUTPUT :

