#### Lab 3

Sort a given set of elements using Merge sort. Time analysis to be done by varying the input size and pattern.Turn in with the program, output screenshots and the observation table.

Code :

#include<iostream>

using namespace std;

void merge(int a[],int l,int m,int h){

    int temp[h-l+1];

    int i=l,j=m+1,k=0;

    while(i<=m && j<=h){

        if(a[i]<=a[j])

        {

            temp[k] = a[i];

            i++;

            k++;

        }

        else{

            temp[k] = a[j];

            j++;

            k++;

        }

    }

    while(i<=m){

        temp[k] = a[i];

        i++;

        k++;

    }

    while(j<=h){

        temp[k] = a[j];

        j++;

        k++;

    }

    int xy = 0;

    for(int x=l;x<=h;x++)

    {

        a[x] = temp[xy];

        xy++;

    }

}

void mergeSort(int a[],int l,int h){

  if(l<h)

  {

      int mid = (l+h)/2;

      mergeSort(a,l,mid);

      mergeSort(a,mid+1,h);

      merge(a,l,mid,h);

  }

}

int main(){

    int n;

    cout<<"Enter number of elements\n";

    cin>>n;

    int a[n];

    int i;

    cout<<"Enter "<<n<<" elements\n";

    for(i =0;i<n;i++){

        cin>>a[i];

    }

    cout<<"Entered arry :\n";

    for(i =0;i<n;i++){

        cout<<a[i]<<" ";

    }

// int y = sizeof(a)/sizeof(a[0]);

    mergeSort(a,0,n);

    cout<<"Sorted arry :\n";

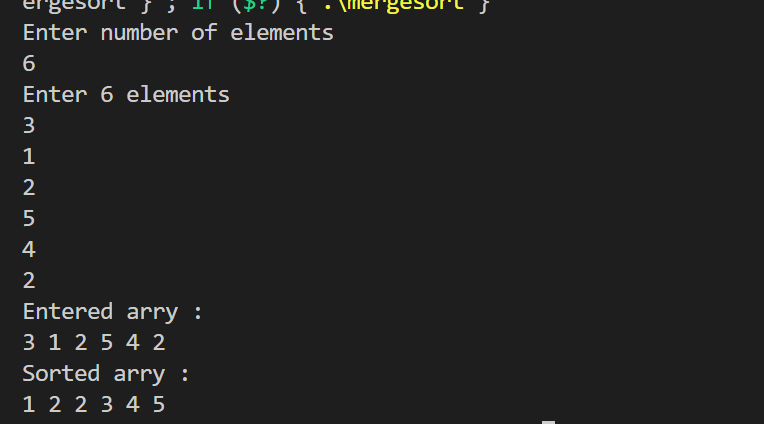
    for(i =0;i<n;i++){

        cout<<a[i]<<" ";

    }

}

###### Output :



Analysis :

