Design and Analysis of Algorithms – 20ISL57A

Program 2 - Implement and analyze merge sort algorithm

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
void simple_merge(int a[],int low, int mid, int high)
{
   int i=low, j=mid+1, k=low, c[10000];
   while(i<=mid && j<=high)
   {
         if(a[i]<a[j])
         {
            c[k]=a[i];
            i++;
            k++;
         }
         else
         {
            c[k]=a[j];
            j++;
            k++;
         }
   }
   while(i<=mid)
         c[k++]=a[i++];
   while(j<=high)
         c[k++]=a[j++];
```

```
    for(i=low;i<=high;i++)
            a[i]=c[i];
}
void merge_sort(int a[],int low,int high)
{
    int mid;
    if(low<high)
    {
            mid=(low+high)/2;
            merge_sort(a,low,mid);
            merge_sort(a,mid+1,high);
            simple_merge(a,low,mid,high);
    }
}
int main()
{
    int a[10000],i=0,n;
    clock_t st,end;
    printf("Enter the value of n\n");
    scanf("%d",&n);
    printf("Random numbers generated are\n");
    for(i=0;i<n;i++)
    {
            a[i]=rand()%100;
            printf("%d\t",a[i]);
    }
    st=clock();
    merge_sort(a,0,n-1);
    end=clock();
    printf("\nAfter Sorting\n");
    for(i=0;i<n;i++)
    printf("%d\t",a[i]);
    printf("\nTime required to sort given elements is %f",(float)(end-st)/CLOCKS_PER_SEC);
}
```