ADA-Test-1

1BM19CS087
Mithun.R.K

- **Program:** Sort a given set of N integer elements using Insertion Sort technique & compute its time taken.

code:

```
#include <math.h>
#include <stdio.h>
#include <time.h>
void insertionSort(int arr[], int n){
    int i, key, j;
    for (i=1; i<n; i++){
        key = arr[i];
        j = i-1;
        while (j>=0 && arr[j] > key){
            arr[j+1] = arr[j];
            j = j-1;
        }
    arr[j+1] = key;
    }
}
int main(){
    int n;
    clock_t start, end;
    double cpu_time_taken;
    printf("enter no. of elements\n");
    scanf("%d", &n);
    int arr[n];
    printf("chosen array is:");
    for(int i=0; i<n; i++){
        arr[i] = rand()%100;
        printf("%d\t", arr[i]);
    }
```

```
start = clock();
insertionSort(arr, n);
for (int c=1; c<=8000; c++)
   for (int d=1; d<=8000; d++) { }
end = clock();
cpu_time_taken = (double)(end-start)/CLOCKS_PER_SEC;
printf("\n Sorted array is:\n");
for (int i=0; i<n; i++).
    printf("%d\t", arr[i]);
printf("\n");
printf("\n time Spent : %f sec\n", cpu_time_taken);
}.
```

- Modification : Apply insertion sort technique starting from the last element in descending order.

```
void insertionSort (int arr[], int n) {.
    int i, key, j;
    for (i=n; i>0; i--) {
        key = arr[i];
        j = i-1;
        while (j<=0 && arr[j] <key) {
            arr[j+1] = arr[j];
            j = j-1;
        }
        arr[j+1]=key;
    }
}.
```