Marwadi University	**Marwadi University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: DAA (01CT0512)	**AIM:** Insertion Sort	
Experiment No: 01	**Date: 18/07/2023**	**Enrolment No:**

Insertion Sort:

Insertion Sort is a simple sorting algorithm that works by repeatedly building a sorted portion of the array on the left side and inserting elements from the unsorted portion into their correct positions within the sorted part. It is an efficient algorithm for small datasets or partially sorted datasets.

Algorithm:

1. Start with the second element (index 1) of the array. The first element (index 0) is considered as the initial sorted portion.
2. Compare the second element with the elements in the sorted portion from right to left until a smaller element is found or the beginning of the array is reached.
3. Insert the second element into the correct position in the sorted portion, shifting larger elements one position to the right.
4. Move on to the third element (index 2) and repeat steps 2 and 3, inserting it into the correct position in the sorted portion.
5. Continue this process for the remaining elements until the entire array is sorted.

Code:

```
#include<iostream>

using namespace std;


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

```
        cout<<arr[i]<<" ";
    }
}
int main(){
    int arr[9] = {1,5,7,6,9,4,3,2,8};
    insertionSort(arr,9);
    return 0;
}
```

OUTPUT:

```
-o InsertionSort } ; if ($?) { .\InsertionSort }
1  2  3  4  5  6  7  8  9
```

Space complexity: _____

Justification:_____

_____

Time complexity:

Best case time complexity: _____

Justification:_____

_____

Worst case time complexity: _____

Justification:_____

_____