**LETSUPGRADE- DATA STRUCTURES AND ALGORITHMS- ASSIGNMENT DAY 5**

**(Srishti Rakesh Pandey)**

**Question 1**

Write the function for insertion sort

**Answer**

Insertion sort is a simple sorting algorithm. It takes a value from the array and places it such that the value to the left of it is lesser than it and values to the right is greater than it. It does not matter the left and right sides of the chosen element must be sorted. In this way, every value is placed.

**Insertion Sort** is basically insertion of an element from a random set of numbers, to its correct position where it should actually be, by shifting the other elements if required.

1. We will store the random set of numbers in an array.  
2. We will traverse this array and insert each element of this array, to its correct position where it should actually be, by shifting the other elements if required.  
3. The first element in the array is considered as sorted, even if it is an unsorted array. The array is sub-divided into two parts, the first part holds the first element of the array which is considered to be sorted and second part contains all the remaining elements of array.  
4. With each iteration one element from the second part is picked and inserted into the first part of array at its correct position by shifting the existing elements if required.  
5. This goes until the last element in second part of array is placed in correct position in the output array.  
6. Now, we have the array in sorted order.

**Program Code:**

/\* C Program to sort an array in ascending order using Insertion Sort \*/

#include <stdio.h>

int main()

{

int n, i, j, temp;

int arr[64];

printf("Enter number of elements\n");

scanf("%d", &n);

printf("Enter %d integers\n", n);

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

{

scanf("%d", &arr[i]);

}

for (i = 1 ; i <= n - 1; i++)

{

j = i;

while ( j > 0 && arr[j-1] > arr[j])

{

temp = arr[j];

arr[j] = arr[j-1];

arr[j-1] = temp;

j--;

}

}

printf("Sorted list in ascending order:\n");

for (i = 0; i <= n - 1; i++)

{

printf("%d\n", arr[i]);

}