**Experiment No 5:**

**A program for Insertion Sort**

**Aim:** Write a program to implement Insertion Sort.

**Theory:**

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

To sort an array of size N in ascending order iterate over the array and compare the current element (key) to its predecessor, if the key element is smaller than its predecessor, compare it to the elements before. Move the greater elements one position up to make space for the swapped element.

**\*Any Solved Example with all passes**

**Algorithm:**

**Algorithm Insertion sort:**

**Procedure 1**

(Insertion sort)INSERTION(A,N)

1.Set A[0]:=-∞.[Initialize sentinel element.]

2.Repeat Steps 3 to 5 for k=2,3,….,N:

3.Set TEMP:=A[K] and PTR:=K-1.

4.Repeat while TEMP<A[PTR]:

(a) Set A[PTR+1]:=A[PTR].[Moves element forward.]

(b) Set PTR:=PTR-1.

5.Set A[PTR+1]:=TEMP.[Inserts element in proper place.]

[End of Step 2 loop]

6.Return.

**PROGRAM: [Write program for Insertion sort-out put should be display in passes]**

#include<stdio.h>

void main()

{

    int n,k,i,temp,loc;

    printf("Enter number of the elements : ");

    scanf("%d",&n);

    int a[n];

    for (i=1;i<=n;i++)

    {

        printf("Enter %d element :",i);

        scanf("%d",&a[i]);

    }

    int ptr;

    a[0] = -9999;

    for (k=2;k<=n;k++)

    {

        temp = a[k];

        ptr = k-1;

        while(temp<a[ptr])

        {

            a[ptr+1] = a[ptr];

            ptr = ptr-1;

        }

        a[ptr+1] = temp;

        for (i=1;i<=n;i++)

        {

            printf("%d ,",a[i]);

        }

        printf("\n");

    }

    printf("Sorted Array : ");

    for (i=1;i<=n;i++)

    {

        printf("%d ,",a[i]);

    }

}

**OUTPUT**

**CONCLUSION:**