**INSERTION SORT**

**DESCRIPTION:-**

Insertion sort is a simple sorting algorithm. The array is virtually split into a sorted part and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted array.

**ALGORITHM:-**

To sort an array of size n in ascending order.

1. Iterate from arr[1] to arr[n] over the array.

2. Compare the current element (key) to its predecessor.

3. If the key element is smaller than its predecessor, compare it to the element before. Move the greater elements one position up to make space for the swapped elements.

**PROGRAM THEORY:-**

Array elements are 12,11,13,5,6

Let us loop for i=1 that is second element of the array to last element of the array(i=4).

For i=1.Since 11 is smaller than 12, move 12 and insert 11 before 12.

11,12,13,5,6.

For i=2.13 will remain at its position as all elements in the array are smaller than 13.

For i=3.5 will move to the beginning and all other elements from 11 to 13 will move one position of their current position.

5, 11, 12, 13, 6

For i=4.6 will move to position after 5, and the elements from 11 to 13 will move one position ahead of their current position.

5, 6, 11, 12, 13

The sorted array is 5, 6, 11, 12, 13.

**PROGRAM:-**

#include <stdio.h>

void insertionsort(int a[],int n);

int a[20];

int main()

{

int num,i=0;

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

scanf("%d",&num);

printf("enter the elements:\n");

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

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

insertionsort(a,num);

printf("the elements after sorting:");

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

printf("%d ",a[i]);

return 0;

}

void insertionsort(int a[], int n)

{

int j,temp,p;

for(p = 1; p < n; p++)

{

temp = a[p];

for(j = p; j > 0 && a[j-1] > temp; j--)

a[j] = a[j-1];

a[j] = temp;

}

}

