**Program – 5**

**AIM – Write an algorithm and program to implement Insertion Sort**

**Algorithm –**

Step 1 − If it is the first element, it is already sorted. return 1;

Step 2 − Pick next element

Step 3 − Compare with all elements in the sorted sub-list

Step 4 − Shift all the elements in the sorted sub-list that is greater than the

value to be sorted

Step 5 − Insert the value

Step 6 − Repeat until list is sorted

**Using Arrays**

**Source Code –**

#include <bits/stdc++.h>

using namespace std;

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;

}

}

void printArray(int arr[], int n)

{

int i;

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

cout << arr[i] << " ";

cout << endl;

}

int main()

{

int arr[] = { 12, 11, 13, 5, 6 };

int n = sizeof(arr) / sizeof(arr[0]);

insertionSort(arr, n);

printArray(arr, n);

return 0;

}

**Using recursion**

**Source Code –**

#include <iostream>

using namespace std;

void insertionSortRecursive(int arr[], int n)

{

if (n <= 1)

return;

insertionSortRecursive( arr, n-1 );

int last = arr[n-1];

int j = n-2;

while (j >= 0 && arr[j] > last)

{

arr[j+1] = arr[j];

j--;

}

arr[j+1] = last;

}

void printArray(int arr[], int n)

{

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

cout << arr[i] <<" ";

}

int main()

{

int arr[] = {12, 11, 13, 5, 6};

int n = sizeof(arr)/sizeof(arr[0]);

insertionSortRecursive(arr, n);

printArray(arr, n);

return 0;

}