**Algorithm:**

**Step 1** − If it is the first element, it is already sorted. So, the sub-list of size 1 is sorted.

**Step 2** − Pick next element

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

**Step 4** − Shift all the elements to right 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

**Required Output:**

|  |
| --- |
| **Test Case :** Array before Sorting: 8, 6, 11, 3, 15, 5  Sorted Array:  3, 5, 6, 8, 11, 15 |

**Insertion Sort Explanation:**

**12**, 11, 13, 5, 6

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

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

**11, 12**, 13, 5, 6

i = 2. 13 will remain at its position as all elements in A[0 … i-1] are smaller than 13

**11, 12, 13**, 5, 6

i = 3. 5 will move to the beginning and all other elements from 11 to 13 will move one position ahead of their current position.

**5, 11, 12, 13**, 6

i = 4. 6 will move to position after 5, and elements from 11 to 13 will move one position ahead of their current position.

**5, 6, 11, 12, 13**

