Algorithm for Program 2:

Step 1: Start bubble sort using an array like BubbleSort(arr)

Step 2: A for loop for taking all the array elements from the user.

Step 3: Another for loop for passing thorough a outer loop.

Step 4: Taken another for loop as inner loop.

Step 5: checking if arr[j+1] < arr[j]

Step 6: Swapping (arr[j], arr[j+1])

Step 7: Using a for loop for checking array values for per pass.

Step 8: Using for loop for printing the output

Step 9: End Bubble Sort

Algorithm for Program 4:

Selection Sort Algorithm

```
    START
        Step 1 :[INITIALIZE ] Array
        Step 2: [INITIALIZE ] Size of Array
        Step 3: [CALL] SELECTION SORT(arr, n)

        Step 4: Repeat Steps 2 and 3 for i = 0 to n-1
Step 5: CALL SMALLEST(arr, i, n, pos)
Step 6: SWAP arr[i] with arr[pos]
[END OF LOOP]
Step 7: EXIT

        SMALLEST (arr, i, n, pos)
Step 8: [INITIALIZE] SET SMALL = arr[i]
Step 9: [INITIALIZE] SET pos = i
Step 10: Repeat for j = i+1 to n
if (SMALL > arr[j])
    SET SMALL = arr[j]
SET pos = j
END OF if]
[END OF LOOP]
Step 4: RETURN pos
           [RETURN TO MAIN]
           PRINT POSITION
           END
```

Algorithm for Program 5:

Step 1: Start of Program

Step 2: Enter the array elements

Step 3: Create a function for insertion sort pass the array and size

Step 4: Inside the Function

Step 5: If the element is the first one, it is already sorted.

Step 6: Move to the next element

Step 7: Compare the current element with all the elements in the sorted List

Step 8: If the element in the sorted array is smaller than the current element, iterate to the next element. Otherwise, shift all the greater element in the array by one position towards the right

Step 9: Insert Value at correct position

Step 10: Repeat until list is completed

Step 11: Print the array after each pass

Step 12: Print the Sorted array

Step 13: End of Program