**Task: Write an algorithm to sort a list of numbers using the Bubble Sort method.**Bubble sort is a sorting algorithm that compares adjacent elements to check if they are in ascending order. If not, these elements are swapped, and the pattern continues until all the elements in the array are sorted from light to heavy. **Q. Define the problem and identify the inputs and outputs.**

**Problem:**The elements are unsorted or placed randomly within the list. We will use the bubble sort algorithm to sort the array. In other words, we will keep shuffling the elements until they are all in the correct order.

Example:  
*input = [7, 12, 9, 11]  
output = [7, 9, 11, 12]*

**Q. Write the algorithm in pseudocode.**

1. The iteration starts from the beginning of the unsorted array.  
2. Compare each pair of adjacent elements.  
3. If the element on the left is greater than the element on the right, swap them.

4. If the elements are already in the correct order, no swap is performed.

5. Once the iteration reaches the end of the list, start a new pass from the beginning.

6. The swapping continues over and over, until all the numbers are sorted.

Illustration:  
[7,12,9, 11] – unsorted array  
[7,12,9,11] – no swap needed  
[7,12,9, 11] – swap required

[7,9,12,11] – swapped

[7,9,12,11] – swap required

[7,9,11,12] – swapped

**Q. Implement the algorithm in a programming language of your choice.**

# Using the function for this algorithim

def bubble\_sort(unsorted\_list):

print('Input: ', unsorted\_list)

# Unsorted\_list variable as the input

indexing\_length = len(unsorted\_list) -1

# Using indexing linked for the comparision

# -1 because we cannot perfom comparision on the last element

sorted = False

# The sorted variable helps us to break the loop once all the elements are in order

# Performing iteration with while loop

while not sorted:

# As long as the sorted is False, continue the loop

sorted = True

for i in range(0, indexing\_length):

if unsorted\_list[i] > unsorted\_list[i+1]:

# Comparing the elements on left and right

sorted = False

# Flip the elements if they are in wrong order

unsorted\_list[i], unsorted\_list[i+1] = unsorted\_list[i+1], unsorted\_list[i]

# Return the new sorted list

return unsorted\_list

# Call the bubble sort function and pass the values

print('Output:',bubble\_sort([5,7,1,3,9]))

**Q. Test the algorithm with different lists to ensure it sorts correctly.  
Test 1:**

*Input: [5, 7, 1, 3, 9]*

*Output: [1, 3, 5, 7, 9]*

**Test 2:**  
*Input: [7, 12, 9, 11]*

*Output: [7, 9, 11, 12]*

**Test 3:**

*Input: [5, 0, 6, 1, 2, 7]*

*Output: [0, 1, 2, 5, 6, 7]*The bubble sort algorithm sorts correctly as expected.