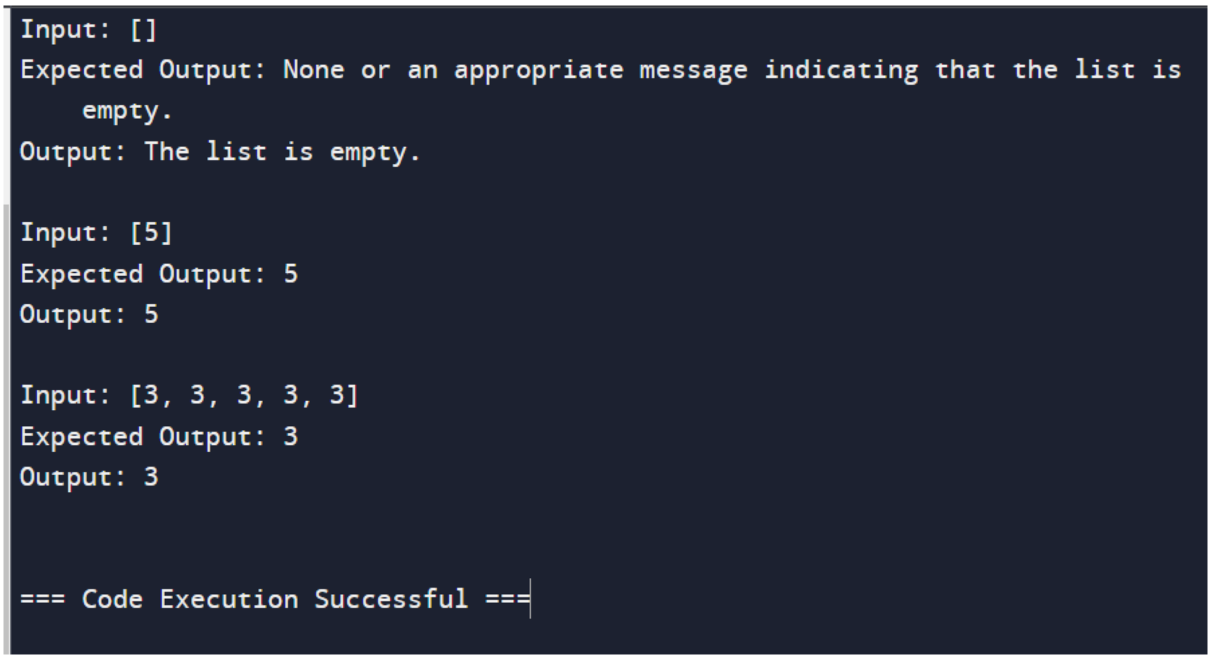
1. 130. You have an algorithm that process a list of numbers. It firsts sorts the list using an efficient sorting algorithm and then finds the maximum element in sorted list. Write the code for the same.
   Test Cases
   1. Empty List
      1. Input: []
      2. Expected Output: None or an appropriate message indicating that the list is empty.
   2. Single Element List
      1. Input: [5]
      2. Expected Output: 5
   3. All Elements are the Same
      1. Input: [3, 3, 3, 3, 3]
      2. Expected Output: 3

PROGRAM:-

```
def process_list(nums):

   if not nums:

      return "The list is empty."



   # Sorting the list using an efficient sorting algorithm (Timsort)

   sorted_nums = sorted(nums)



   # Returning the maximum element (the last element in the sorted list)

   return sorted_nums[-1]



# Test Case 1: Empty List

input1 = []

output1 = process_list(input1)

print(f"Input: {input1}\nExpected Output: None or an appropriate message indicating that the list is empty.\nOutput: {output1}\n")



# Test Case 2: Single Element List

input2 = [5]
```

```
output2 = process_list(input2)

print(f"Input: {input2}\nExpected Output: 5\nOutput: {output2}\n")


# Test Case 3: All Elements are the Same

input3 = [3, 3, 3, 3, 3]

output3 = process_list(input3)

print(f"Input: {input3}\nExpected Output: 3\nOutput: {output3}\n")
```

OUTPUT:-



```
Input: []
Expected Output: None or an appropriate message indicating that the list is
    empty.
Output: The list is empty.

Input: [5]
Expected Output: 5
Output: 5

Input: [3, 3, 3, 3, 3]
Expected Output: 3
Output: 3


=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(nlogn)