Task 1:

The following are the critique from the scripts.

1. **The function definition was missing a colon.**

In Python, a colon is used to indicate the start of a new block of code, such as in a function definition, conditional statement, or loop. The absence of a colon after the function definition in the original code is a syntax error.

2. **The while loop condition lacked a colon.**

Similar to the function definition, a colon is required after the condition of a 'while' loop in Python. This signals the beginning of the code block associated with the 'while' loop.

3. **The indentation of the code was inconsistent.**

Python relies on indentation to define blocks of code. Consistent indentation is crucial for the interpreter to understand the structure of the code. In the script, the indentation was inconsistent, leading to a potential indentation error.

4. **The range in the first for loop was incorrect.**

The range in the first 'for' loop should go up to 'len(L)-1' because the loop accesses elements up to the second-to-last element of the list. Using 'len(L)' would cause an 'IndexError' when trying to access 'L[i + 1]' at the last iteration.

5. **The range in the second for loop was incorrect.**

In the second for loop, which is intended to iterate backward, the range should start from 'len(L)-1' and go down to '1 (not 0)'. This ensures that the loop covers elements from the last to the second element. The step -1 indicates the backward iteration.

6. **The comparison in the if statements was incorrect**

The purpose of the code is to swap elements when the current element is greater than the next one. It should be 'L[i] > L[i + 1]'.

The corrected script

```python
def whatdoido(L):
    s = True
    while s:
        s = False

        # Forward pass (from the beginning to the end)
        for i in range(len(L) - 1):
            if L[i] > L[i + 1]:
                # Swap values L[i] and L[i+1]
                L[i], L[i + 1] = L[i + 1], L[i]
                s = True

        # Backward pass (from the end to the beginning)
        for i in range(len(L) - 1, 0, -1):
            if L[i] < L[i - 1]:
                # Swap values L[i] and L[i-1]
                L[i], L[i - 1] = L[i - 1], L[i]
                s = True

# Example usage:
to_sort = [5, 2, 9, 1, 5, 6]
whatdoido(to_sort)
print(to_sort)  # Sorted list
```

Task 1 – Part B:

```
def whatdoido(L, include_backward=True):
    #counting the number of comparison from forward pass
    comparisons_forward = 0
    #counting the number of swaps from forward pass
    swaps_forward = 0
    #counting the number of comparison from backward pass
    comparisons_backward = 0
    #counting the number of swaps from backward pass
    swaps_backward = 0

    s = True
    while s:
        s = False

        # Forward pass (from the beginning to the end)
        for i in range(len(L) - 1):
            comparisons_forward += 1  # Increment comparisons counter
            if L[i] > L[i + 1]:
                # Swap values L[i] and L[i+1]
                L[i], L[i + 1] = L[i + 1], L[i]
                swaps_forward += 1  # Increment swaps counter
                s = True

        # Backward pass (from the end to the beginning)
        for i in range(len(L) - 1, 0, -1):
            comparisons_backward += 1  # Increment comparisons counter
            if L[i] < L[i - 1]:
                # Swap values L[i] and L[i-1]
                L[i], L[i - 1] = L[i - 1], L[i]
                swaps_backward += 1  # Increment swaps counter
                s = True
```

```
print("After sorting:")
    print(L)

    print("Forward pass:")
    print(f"Comparisons: {comparisons_forward}, Swaps: {swaps_forward}")

    if include_backward:
        print("Backward pass:")
        print(f"Comparisons: {comparisons_backward}, Swaps: {swaps_backward}")
        print(f"Total comparisons: {comparisons_forward +
comparisons_backward}")
        print(f"Total swaps: {swaps_forward + swaps_backward}")
    else:
        print(f"Total comparisons: {comparisons_forward}")
        print(f"Total swaps: {swaps_forward}")

# Example usage:
to_sort = [5, 2, 9, 1, 5, 6]
whatdoido(to_sort, include_backward=True)
print("----------------------------------------------------------------------")
print("The total comparison is 20 and total swaps is 6.")
print("However, upon removing the backward comparison, the total was 6 with 4
swaps")
```

Output:

```
After sorting:
[1, 2, 5, 5, 6, 9]
Forward pass:
Comparisons: 10, Swaps: 4
Backward pass:
Comparisons: 10, Swaps: 2
Total comparisons: 20
Total swaps: 6
----------------------------------------------------------------------
The total comparison is 20 and total swaps is 6.
However, upon removing the backward comparison, the total was 6 with 4 swaps
```