**Camelcase:**

Time Complexity O(n):

The function iterates through each character of the input string “s” using a “for” loop. The loop runs through each character of the string, which is O(n) where `n` is the length of the input string.

Space Complexity O(1):

1. The space complexity of the function is O(1) because it uses a single integer variable `count` to store the count of camel case words. The space used by this variable is constant and does not depend on the size of the input string.

**Correctness and the Loop invariant :**

1. Time Complexity (O(n^2)):

In the worst-case scenario, when the array is in reverse order, it will take the maximum number of swaps. In each outer loop, it may need to compare and swap elements up to the current position, In the inner while loop again it has to go through every element. Therefore, the time complexity of this program is O(n^2), where 'n' is the number of elements in the array.

2. Space Complexity (O(n)):

The space complexity of the program is determined by the space used to store the input array 'ar'. The input array is created with a size of 'n', where 'n' is the number of elements provided as input. Hence, the space complexity is O(n) because the program uses additional memory to store the input data.