Russell Harvey

Professor Steele

CSCI-141

4 October 2018

Homework 6 – Debugging

Caesar:

1. print("Expected Caesar: 0 | Actual Caesar: ", caesar("very", "very"))  
   print("Expected Caesar: 1 | Actual Caesar: ", caesar("very", "wfsz"))

Expected Caesar: 0 | Actual Caesar: 0

Expected Caesar: 1 | Actual Caesar: 1

1. For the first print statement, since the “shift” variable is 0, the program runs as it is supposed to because every ASCII value in the original and the codeword is the same. As for the second, since every character in the string is shifted up by one, the “shift” variable is properly assigned to 1 and the function will return 1. These work because in the first two working codeword/original pairs for the word very, none of the codeword characters’ ASCII values return to a lower ASCII value than the original’s.
2. When the program encounters a character with a lower ASCII in the codeword than the original. The statement that assigns “shift” a value gets tossed out of whack. While shift is assigned the correct shift value, when the loop moves to a character that does not have a lower ASCII value than the original the comparison will not be equivalent to shift and will result in the function returning -1 instead of the proper shift value. To fix this, I implemented a logic system where if the ASCII value of the original is greater than the codeword, it would take the difference of the original minus the codeword instead of the opposite. It takes this difference and subtracts it from 26, assigning it to shift giving shift the correct value for the loop to work. I also added an additional few statements in the for loop that checks for this same issue. If it is present, the loop performs the same operation stated above within an if statement so that the shift value will match the shift value at every index of the string.

Longest Consecutive Matching Substring: