Technical Question 2:

For this problem, I have created a for loop to traverse the length of the string – 1. The loop cycles through the entire string beginning at the 1 index. Which each loop, we assign a low and high variable depending on which while loop we are visiting.

The first while loop’s base condition is that the low variable has to be greater than 0 (which is the starting of the string), the high is less than the length of the string and two letters match.

If this condition is met, we record that low variable value in the start variable and then expand the range of letters to be examined by 1 in each direction. We remain in this while loop until the base condition is no longer met.

We then move on to our second while condition. For this comparison, we look at a higher range of characters because we are adding one to the high position. If the base condition is met (low being greater than 0, high being less than the string length and the two letters match, we capture the position of the low variable in start, store its length and then move the low down 1 and the high up 1. We remain in this while condition until our base condition here is no longer met.

Finally, after moving through the length of the string, if the length of the found palindrome is greater than 1 character, we return the substring beginning at the position recorded in start of length start + length.

Our time constraint is O(E) since we need to visit each character of the string. Our space constraint is at most O(E).