1. I didn’t know what palindrome was and I had to first understand and look for more definition and examples of palindrome words. Once I understood and thought about the palindrome problem, I realized that the problem could be answered both stack and queues but for this assignment it is done in stack data structure in order to solve it. Since a palindrome is a word or phrase that is the same as forward and backwards, I decided to push all the characters of the input string onto a stack. I then popped all of the characters off the stack and put them back into the string. Finally, I checked to see if the input string was the same as the string in reverse. If so, I pointed out that the input string was a palindrome. Otherwise, I pointed out that the input string was not a palindrome.

Text, letter

Description automatically generated

A white paper with writing on it

Description automatically generated with low confidence

5. Thinking in terms of the material regarding computational complexity that we just finished covering answer the following in the document portion of your homework:

a. (5 Points) What is the basic operation in your code?

*The basic operation in the code is to push all characters of the input string onto a stack and then pop all characters off the stack and put them back in the string. Finally, the code checks whether the input string is the same as the string in reverse. If so, it prints out that the input string is a palindrome. Otherwise, it prints out that the input string is not a palindrome.*

b. (5 Points) What input determines how many time the basic operation takes place ?

*What input determines how many times the basic operation takes place  
The input that determines how many times the basic operation takes place is the length of the input string.*

c. (5 Points) Express the number of times the basic operation occurs in terms of n. What does n represent?

*The number of times the basic operation occurs is n, where n is the length of the input string*

d. (5 Points) What is the computational complexity of the code in terms of Big O? Explain why.

*The computational complexity of the code is O(n), where n is the length of the input string. The computational complexity of the code is O(n) because the code loops through the input string n times to push all characters onto the stack and then pop all characters of the stack. The number of times the code loops through the input string is directly proportional to the length of the input string, so the computational complexity is O(n).*

5. Explain how a stack data structure differs from a bag data structure. Explain how you used the stack in your code to solve the palindrome problem

*Bag data structure contains unordered collection of values that may have duplicates whereas stack are order collection of values such that the first value goes at the bottom and every other value sits on top of each other. When removing a value a bag, you can remove any value but stack is a linear collection whose elements are added in a last-in-first-out. I used stack: I created the class palindrome, I then created a scanner instance. After that I created stack and a method that pushes all characters of the input string onto a stack. I then created another method that pops all characters off the stack and puts them back in the string. Finally, the program checks whether the input string is the same as the string in reverse. If so, it prints out that the input string is a palindrome. Otherwise, it prints out that the input string is not a palindrome*.