Lab 04 – Recursive Palindromes

# Problem

Issued the task of writing a recursive method that that checks to see if a string entered by the user is a palindrome.

# Proposed Solution

Take input from the user, pass input to a recursive method that checks from outer to inner on the string to see if char = char. Move from index0-> length-1 while also moving from length-1 to index 0. Ensure to trim / ignore whitespace.

# Tests and Results

Relatively simple lab, especially with the provided hints. Successfully found palindromes with the input of “Taco Cat, taco cat, taco cat, ta co ca t, radar, r a d a r,” etc.

# Problems Encountered

I had some issues with white space not cooperating with me until I utilized the trim method. I was also returning true at the end of my recursive method instead of returning the method itself, so after fixing that it gave me no more issues.

# Conclusions and Discussions

Good practice with recursion. I had to look up the trim method in the java docs to better understand how that worked.

# Additional Questions

**Lab Report Questions:**

1. Using the recursive method, write out the string that is being processed at each step if the word “radar” was entered.  Do the same for “Straw arts”.

Input = radar

r = r CHECK

input = ada

a = a CHECK

d < length2 RETURN TRUE

Input = Straw arts

S = s CHECK

Input = traw art

T = t CHECK

Input = raw ar

R = r CHECK

Input = aw a

A = a CHECK

Input = w

W < length2 RETURN TRUE

1. Write some simple using the iterative method (using loops) to solve this same problem.

String word = input.NextLine();

Int indexStart=0, indexEnd = word.length()-1;

While ( indexStart != indexEnd)

{

If ( word.CharAt(indexStart) != word.CharAt(indexEnd)

{

S.o.pl(“Not a palindrome”);

Return;

}

Else

{

indexStart++;

indexEnd--;

}

}