**Palindrome Checker**

**Project Description & Goal**

The goal of this project is to develop an application that will verify if the user’s input is a palindrome or not. A palindrome is a phrase that is spelt the same way forward as it is backwards. For example, “racecar” and “anna” are palindromes. Even more complex, we have phrases like, “A Santa dog lived as a devil god at NASA” which are palindromes. A palindrome can be checked in multiple ways. One of which is to compare both ends of the string character by character until there is only one or zero characters left to check. Another way is to store the string in another variable backwards and compare the two.

**For the purpose of this assignment, you can consider spaces to be characters that count. Generally, palindromes don’t consider spaces. However, you will find much more interesting results if you exclude strings.**

**Project Learning Objectives**

* Learn how to treat a string as an array to iteratively decipher information about it.
* Learn how to algorithmically compare data, and why some methods are more effective at doing so.

**Project Demonstrated Competencies**

1. Palindrome is correctly checked and verified.
2. Student understands the 2 described methods of checking and understands why one is more efficient than the other.
3. Able to parse through a very large string to find all palindromes that exist within it.

**Rubric**

|  |  |  |
| --- | --- | --- |
|  | **Description of perfect implementation** | **Score** |
| Competency #1 | Algorithm returns correct results based on the user’s input | \_\_\_  50 |
| Competency #2 | Include in your submission a document (.docx, .doc, .txt) that describes which of the two methods is more efficient (space wise, and complexity wise). | \_\_\_  50 |
| Competency #3 | Alter your program to find every palindrome in a string. Find strings that are very long, for example a chapter of a book. | \_\_\_  +25 |