**CSE 212 – Programming with Data Structures**

**W03 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Chris Bagwell |
| **Date:** | 9/26/2022 |
| **Teacher:** | Bro Zachariah Alvey |

*It is a violation of BYU-Idaho Honor Code to post or share this document with others or to post it online. Storage into a personal and private repository (e.g. private GitHub repository, unshared Google Drive folder) is acceptable.*

**Question 1: From Part 1, describe what the Mystery Stack 1 code does and how the use of a stack helps in the implementation.** The stack uses the append method to take each letter from the original string and put it into the new stack. We then go through that stack one element a time and pop of each item from the top(or back of the string) and append that to a new string (result) we then return that result to show a backwards version of the original string.

**Question 2: From Part 1, what are the three outputs from the Mystery Stack 1 code for the following three different inputs?**

* **Racecar:** racecar
* **Stressed:** desserts
* **a nut for a jar of tuna :** anut fo raj a rof tun a

**Question 3: From Part 2, describe what the Mystery Stack 2 code does and how the use of a stack helps in the implementation.** This function takes in a string and does some tests to see if a calculation can be done. It uses a stack to store the numbers and operations that need to happen on those numbers then returns the only item left in the stack as the result. There are several checks to make sure that the items passed into the function are valid such as more than one item in the list, division by 0, invalid characters, and if any thing is passed on or just spaces. The stack stores numbers and if there is an operator it applies that to the numbers in the stack. If there is only one number left in the stack that is the result of all the calculations.

**Question 4: From Part 2, answer the following regarding what the Mystery Stack 2 code does:**

* **What will the result be if the input parameter is: 5 3 7 + \*:** 50.0
* **What will the result be if the input parameter is: 6 2 + 5 3 - / :** 4.0
* **What input parameter would result in the display of “Invalid Case 1!”** only one operator such as mystery\_2(‘+’), mystery\_2(‘-’), mystery\_2(‘\*’), mystery\_2(‘/’)
* **What input parameter would result in the display of “Invalid Case 2!” :** if a division by 0 occurs such as mystery\_2(‘4 2 2 - /’)
* **What input parameter would result in the display of “Invalid Case 3!”:** Invalid charactesr such as letters or no spaces between numbers and operators such as mystery\_2(‘a 5 5+’) or mystery\_2(‘537+\*’)
* **What input parameter would result in the display of “Invalid Case 4!”:** empty string or all spaces such as mystery\_2(‘’) or mystery\_2(‘ ’)