**CSE 212 – Programming with Data Structures**

**W03 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Daniel Cross |
| **Date:** | 10/2/2024 |
| **Teacher:** | Brother 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 Mystery Stack 1 code flips the characters in a string to the opposite direction. It does this by taking a string and adding each character to the stack individually. It then goes through and pops out the last character in the stack and adds it the new variable, then returns it.

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

* **racecar**
* **stressed**
* **a nut for a jar of tuna**

1. racecar
2. desserts
3. 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.**

Mystery Stack 2 does math for you in a specific way. It requires one more number than it does an operand. It uses numbers from right to left, but operators from left to right. The use of a stack keeps it in this order and verifies the correct ratio of numbers and operators.

**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**
* **What will the result be if the input parameter is: 6 2 + 5 3 - /**
  + **4**
* **What input would result in the display of “Invalid Case 1!”**
  + **This verifies there are not too few numbers in the stack (ex. Input being “5 +”)**
* **What input would result in the display of “Invalid Case 2!”**
  + **This stops a DivideByZero error by checking the denominator (ex. “5 0 /”)**
* **What input would result in the display of “Invalid Case 3!”**
  + **This stops other symbols from getting in (ex. “: { &”)**
* **What input would result in the display of “Invalid Case 4!”**
  + **This verifies there are not too many numbers in the stack (ex. “5 6 7 +”)**