**CSE 212 – Programming with Data Structures**

**W03 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Joshua Ludwig |
| **Date:** | 5/6/2023 |
| **Teacher:** | 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 mystery Stack 1 code takes each letter from the end of the text and adds that letter into the stack list. Then while the stack list is above 0, the result will copy the stack and delete the last index of the stack. In the end the result will be the reverse of the text input. The stack list helps hold all the letters in the text.

**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.**

The mystery Stack 2 code creates a stack for numerical items to be stored in a list for the operators to be used on. There are multiple Invalid Cases in which either the stack holds on too many items or not enough and if the text variable cannot be expressed in a mathematical expression. The text variable is used to calculate a math expression and operates it in an unconventional way and returns it.

**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!”**
  + The result will display Invalid Case 1 if there are not enough items in the stack for the operators. This is because the operators will not have enough items in the stack to operate on and complete the expression.
* **What input parameter would result in the display of “Invalid Case 2!”**
  + The result will display Invalid Case 2 if one of the items is being divided by zero. This is because you cannot divide by zero without causing an error.
* **What input parameter would result in the display of “Invalid Case 3!”**
  + The result will display Invalid Case 3 when the text cannot be converted into a mathematical expression. Either because there are no numbers or because there are no spaces in the text.
* **What input parameter would result in the display of “Invalid Case 4!”**
  + The result will display Invalid Case 4 if there is not 1 item in the stack. This is because every operator in the expression needs two numbers and if the stack has zero or more than one number then the result will display Invalid Case 4.