**CSE 212 – Programming with Data Structures**

**W03 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Daen Antule |
| **Date:** | May 7 2023 |
| **Teacher:** | Bro. Kunz |

*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.**

It puts the letters of the word on the stack, one by one. Then, it takes them off one by one, starting from the top, and puts them in a new word. The stack helps the reverse the word easily, by keeping the letters in the opposite order they were originally in. The function uses stacks to take the last values of the stack out and return them backwards, the words being input as test returns backwards.

**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 code uses a stack to determine what characters are inside of a given string and returns the corresponding result. The function first checks if each item in the input text is an operator (+, -, \*, /) or a number. If it is an operator, the function pops the last two operands from the stack, performs the operation, and pushes the result back onto the stack. If it is a number, the function converts it to a float and pushes it onto the stack. If the item is an empty string or an invalid character, the function ignores 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**
* **What will the result be if the input parameter is: 6 2 + 5 3 - /**
  + **4**
* **What input parameter would result in the display of “Invalid Case 1!”**
  + **To trigger "Invalid Case 1!", provide an input parameter `text` with only one operator, for example: `"+"`.**
* **What input parameter would result in the display of “Invalid Case 2!”**
  + **To trigger "Invalid Case 2!", provide an input parameter `text` with the second operand equal to 0 for a division operation, for example: `"5 0 /"`.**
* **What input parameter would result in the display of “Invalid Case 3!”**
  + **To trigger "Invalid Case 3!", provide an input parameter `text` with a non-float and non-operator item, for example: `"5 3 A +"`.**
* **What input parameter would result in the display of “Invalid Case 4!”**
  + **To trigger "Invalid Case 4!", provide an input parameter `text` with no operators or an incomplete expression, for example: `"5 3"`.**