**CSE 212 – Programming with Data Structures**

**W03 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Jonathan Starks |
| **Date:** | 5/11/2024 |
| **Teacher:** | Brother Comeau |

*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 adds letters to a stack from a string called text, then it will take letters off and add them to result until the stack is empty, then it returns result.

The use of a stack like this will reverse the text it is given.

**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 program goes through a stack, it determines things by spaces, it looks at a variable named item, if the item is a +, -, \*, or / it will check the length of the stack, if the length of the stack is less than 2 it will throw an error, it pops the top 2 things in the stack and sets them equal to op2 and op1 and makes a float named res, then depending on what the item is the program will either add, subtract, or multiply op1 and op2, if the value of op2 is zero then the program, will throw an error, but if it isn’t then the program will divide them, then the program will add res onto the stack. If the item is anything other than a +, -, \*, or / then it will try to change the item to the value type of float and add it to the stack. If the item is nothing then the program will do nothing, and in all other cases the program will throw an error. The program checks the length of the stack and if it isn’t 1 then it will throw an error. The program will then pop off and return the item in the stack.

If a list was used instead of a stack, then you would have to explicitly say what index you wanted to use every time you needed something.

**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 + \***
  + **Invalid Case 4! The length of the stack is not 1.**
  + **\*, +, 22**
* **What will the result be if the input parameter is: 6 2 + 5 3 - /**
  + **The code would break as “+” cannot be subtracted by 0.333333333.**
  + **/ - 3 5 + 0.33333333333**
* **What input would result in the display of “Invalid Case 1!”**
  + **If there are less than 2 numbers in the stack**
* **What input would result in the display of “Invalid Case 2!”**
  + **Any time than op2 = 0**
* **What input would result in the display of “Invalid Case 3!”**
  + **Any case where the item isn’t a +-\* or /**
* **What input would result in the display of “Invalid Case 4!”**
  + **Anytime the final list isn’t a length of 1**