**CSE 212 – Programming with Data Structures**

**W03 Prove – Response Document**

|  |  |
| --- | --- |
| **Name:** | Luis Roberto Cardenas Melgar |
| **Date:** | 5/7/2022 |
| **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 a word as the input, and it flips the order of the letters in the word. The first letter becomes the last one, and the last one, the first one. Inside the For loop, the program takes a word and creates a list with the letters from word. Then the program uses a While loop to remove the last letter from the list and add that letter to the new word (variable) called result every time it runs the code within the loop. It implements the stack by storing letters in the list and removing letters from the list. When we add the first character to the list, it’s stored in the front of the stack. When we add more characters to the list, they are stored in the back of the stack. When we remove a character from the list, it is removed from the back of the stack.

**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.**

Mystery Stack 2 code takes a string, splits the characters in the string, and makes a list with them (It doesn’t consider single whitespaces as characters). It loops through the list and stores the values that are numbers in another list called stack. If there are symbols like “+,” “-,“ “\*,” and “/” in the original list, the program uses them to indicate what operation to perform. It then performs operations like addition, subtraction, multiplication and division with the numbers in stack list. If there are extra whitespaces, it ignores them and check for invalid cases. At the end, it returns the results of all the operations performed. The stack allows the program to create this list called stack with the numerical values of a string, pop stored numbers from the back of the list to perform operations with them and store the result back in the list. As long as there are characters in the original list, the program keeps looping through the code, performing all the operations specified by the operators until there is only one value in the list. That value is the final result of all the operations.

**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!”**

The program would display “Invalid Case 1!” if the first or second element in the list created with the string characters were operators like “+,” “-,“ “\*,” or “/.” The reason is that there wouldn’t be numerical values to use to perform operations yet.

* **What input parameter would result in the display of “Invalid Case 2!”**

The program would display “Invalid Case 2!” if, in the list created with the string characters, there is a zero (0) before a division operator (/). The reason for this is to avoid dividing by zero because it’s undefined. If that happened, when the program got to the division operator, the program would first pop the number zero and then the number previous to zero in the list and divide the other number by zero.

* **What input parameter would result in the display of “Invalid Case 3!”**

The program would display “Invalid Case 3!” if there are other characters in the string besides numerical values or the before mentioned operators. This would stop the program from trying to perform operations with letters or any other no valid character.

* **What input parameter would result in the display of “Invalid Case 4!”**

The program would display “Invalid Case 4!” if there are only numerical values in the string input and not operators. If there are no operators, the program would only store all the numbers in the stack list, and it wouldn’t perform any operations with them.