Brian Ross

Dr. Prudence

CSC 201

2/26/16

Program 2

Objective:

The objective of this program is to take a user’s input in the form of a string, dissect the string, and interpret its components. In this program the user is instructed to enter a string of characters within a set of parentheses.

Variables:

memory: is an empty set that store’s the user’s inputs

k: is the user’s initial input

para\_arg1: is the index of the “(“ character in the user’s string

para\_arg2: is the index of the “)” character in the user’s string

left: is the mathematical argument in the user’s string

sign: is the index of the character that follows “(“

num1: assigns an integer value to the character(s) between the “(“ and the argument

num2: assigns an integer value to the character(S) between the argument and the “)”.

mem1: is the computation of the user’s input appended to an empty list

inputs: asks the user if they would like to recall previous strings

list\_number: is the integer value corresponding to the number of previous string the user wants to recall

user\_input: asks the user if they’d like to end the program

Algorithm:

The user is prompted to enter a string of character in the form “( number +-/\* number )” if the user enters characters that do not coincide with the form they will receive an output stating an invalid input or computation failure. If the user enters a string with the desired form the program will pick out the mathematical argument by finding the index of the symbol. Then the code will determine the value of the characters to the left and right of the argument. Once the integers have be located and defined the program will run the mathematical expression and output the value of the user’s string. In addition the program will append that value to the empty list that exists outside of the while loop. The program will then prompt the user to recall previous expressions. If the user chooses, they can recall any number of previous expressions that will be presented in the form of a list. The program will finally prompt the user to end the program, unless the user chooses to end the program it will continue to prompt the user for more expressions.

Problems:

Only one argument can exist in the user’s string, for example, (5\*2) will yield a result whereas (5\*2+3) will create an error. Other characters in the initial input such as letters, symbols, and/or undefined mathematical symbols (ie. %) will also cause the program to crash. Attempts to pull from the memory list following an input that “does not compute” will crash the program.