

COL216

ASSIGNMENT – 2

ABHINAV SINGHAL (2019CS50768)

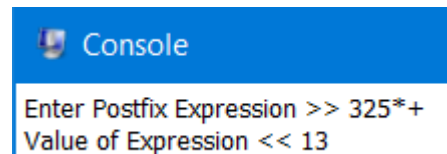
ABHISHEK KUMAR (2019CS10458)

Problem Statement: Write a MIPS Assembly Program for evaluating an expression in postfix format.

- **Major Design Considerations:**

A) Input/Output: The user interface is kept as simplistic as possible. When we execute the code, it asks for the Postfix expression. Once the full Postfix expression has been typed by the user, the value of the expression is given as the output, only if the expression is valid. Otherwise, there will be an error message in the output.

Please note that the entire Postfix expression must be typed in one go without pressing Enter. Once Enter (the newline character) is pressed, the string up to that point is taken as the Postfix expression and is evaluated accordingly.



I/O Interface for sample test case

B) Implementation: After taking the entire Postfix expression as the input, it is stored in memory. Memory allocated for the input string has been fixed at 1000 Bytes, i.e., 1000 Characters. We maintain the Stack data structure for evaluating the expression. Since we are dealing with integers in the range 0-9 and operators +, - and * only, the stack data structure stores 32-bit Integers. Stack size has also been limited to store up to 1000 Integers.

- Once we have the Postfix expression, we begin traversing the expression character by character. If we see an integer character (0-9), we push onto the stack, else if we see a valid operator, we pop the top two elements of the stack and perform the operation and push back the result onto the stack.
- Once the entire Postfix expression has been traversed and if no exceptions are raised then the value of the expression is given as output on the console.

C) Raising Exceptions: We raise exceptions at appropriate locations:

- UndefinedCharError: This exception is raised when a character other than integers in the range 0-9 or operators +, - and * is seen in the Postfix expression. When the exception is raised, the undefined character is also shown on the console. **We do not raise this error upon seeing a whitespace in the expression as whitespace is treated as neutral character which has especially been ignored.**
- LessThanTwoOperandsInStackError: As the name suggests, this exception is raised when there are less than two operands in the stack and it is asked to perform a binary operation such as +, - or *.

- StackSizeMoreThanOneInTheEndError: This expression is raised when there are more than one operand present in the stack after the entire postfix expression has been traversed once. There should be only one element present finally in the stack which would be the answer.
- StackOverflowError: This exception is raised if maximum space allocated to the integer stack is exceeded, i.e., we are trying to push a greater number of operands than allowed into the stack.
- **Test Cases:** We have rigorously tested our implementation using multiple test cases, a variety of them shown below. The corresponding Postfix expressions have been obtained using the C++ Code given to us in the assignment description.
 - Testcase – 0: (Empty String)
 - ⇒ Infix Expression >>
 - ⇒ Postfix Expression >>
 - ⇒ Value of Expression << 0
 - Testcase – 1: (Example given in assignment description)
 - ⇒ Infix Expression >> $3 + 2 * 5$
 - ⇒ Postfix Expression >> $325*+$
 - ⇒ Value of Expression << 13
 - Testcase – 2:
 - ⇒ Infix Expression >> $3 + 4 * (8 - 3) - 2 * 5 + 6$
 - ⇒ Postfix Expression >> $3483-*+25*-6+$
 - ⇒ Value of Expression << 19
 - Testcase – 3:
 - ⇒ Infix Expression >> $2 + 3 / 4 - 1$
 - ⇒ Postfix Expression >> $234/+1-$
 - ⇒ UndefinedCharError: Unknown Character < "/" > detected in Postfix String -> Program Terminated.
 - Testcase – 4:
 - ⇒ Infix Expression >> $4 + 5 * 3 + -6$
 - ⇒ Postfix Expression >> $453*++6-$
 - ⇒ LessThanTwoOperandsInStackError: Cannot apply binary operator as less than 2 operands present -> Program Terminated.
 - Testcase – 5:
 - ⇒ Infix Expression >> $5 + 2 \ 3$
 - ⇒ Postfix Expression >> $52+3$
 - ⇒ StackSizeMoreThanOneInTheEndError: More than one element remaining in stack at the end -> Program Terminated.

Apart from these many more have test cases have been tried and manual calculations have also been done to tally the answers.

-THANK YOU-