COL 216: Computer Architecture

Assignment 2

Saurabh Verma, 2019CS50129

Preface:

Any mathematical expression can be expressed in different forms such as infix: A+B*C, prefix: +*ABC or postfix: ABC+*. In this assignment we consider the postfix expression and our objective is to evaluate it in O(n) time complexity where n is the number of operands.

Approach:

Here we have taken use of the **stack** functionality of the mips instruction set using $sp address registers. We have statically allotted the memory of 10^5 bytes for the purpose of this assignment.

The basic approach is we scan the input string one character by character and check the ascii values and if the ascii value is for 0-9 integer we push it onto the stack after converting into int from character. If we encounter any operator we pop from stack and perform operation and push on stack again. If invalid character is entered error is flagged. If we get the newline we terminate the function and give out the answer which is required.

User Interface

First the system asks for the postfix expression which has to be evaluated.

-Enter the postfix expression which has to be evaluated:

Now after this the computations start and then if anything unexpected happens errors are reported as mentioned below.

Errors

i) If the no. of operands is provided less than the operator demands then we give out the invalid expression error as "*-Invalid Expression: More operands expected"*

ii) If the no. of operator is provided less than the operands then we give out the invalid expression error as - "*Invalid Expression: More operators expected"*

iii) If an invalid character is input the error is as follows-*Invalid character:<char> found in postfix expression at position <position>"*

Testing Strategy:

We have tested our code on a variety of test cases and corner cases which give us the error as well as the specified output

i) 4=5: Gives error

ii) 35*4-14*+4-6+: 17 which is the required output.
iii) 34*7- : 5 which is the required output.
iv) 23++: gives error.
v) 233+: gives error.
vi) 231*+9-: -4 which is the required output.
vii) 123+*8-: -3 which is the required output.
viii) 562*-7+:0 which is the required output.
ix) 123+*8+: 13 which is the required output.
x) 641*+8-:2 which is the required output.
xi) Test cases from the site checked
https://practice.geeksforgeeks.org/problems/evaluation-of-postfix-expression1735/1
https://www.includehelp.com/icp/postfix-expression-evaluation.aspx

Alternate Approach

In the end of this approach we have in comments provided an alternate approach which uses dynamic memory allocation rather than static.

****