Assembly Programming and Computer Organization

University at Albany
Department of Computer Science
ICSI 404 – Spring 2023

Programming Assignment 2

Programming Assignment-2
Assigned: Tuesday, March 28th, 2023.

Due: Tuesday, April 18th through your Blackboard account by 11:59 PM. Submissions with 20% penalty will be accepted by Thursday, April 20th, by 11:59.

Unlimited number of submissions is allowed.

Objective

To acquire expertise in stack manipulation and management, subroutine linkage and return conventions.

Description

You are to write a complete program in MIPS assembly language that evaluates arithmetic expressions. The expressions must be *fully parenthesized* and include the following expressions.

1. + (addition)
2. - (subtraction)

For simplicity all input values for the expressions will be a single base ten digit (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Your program must be composed of four states: *input, convert-to-postfix, evaluate, and output* states. At *input* data must be provided through the keyboard and stored as an array of characters. After one expression is entered your program moves to the *convert-to-postfix* state. At this state your expression must be converted to the postfix notation using a stack-based algorithm. Your program must then move to the evaluate state which evaluates the postfix expression using a stack-based algorithm. At the output state your code must display the complete expression in the postfix notation followed by the = symbol and the expression's numeric result.

Example

Some valid expressions and their corresponding postfix notations are:

a) ((1-3)+5) corresponds to 13-5+ in postfix notation.
b) (1-(3+5)) corresponds to 135+- in postfix notation.

Shown below is the Console display for expression a) above:

Console

```
Expression to be evaluated:
((1 - 3) + 5)
13-5+ = 3
```

Valid Input Expressions

Valid input expressions are completely parenthesized, infix arithmetic expressions consisting of nonnegative integer digits, and the two operations +, -. The following definition gives all such valid expressions:

1. Any nonnegative integer is a valid infix expression.

2. If a and b are valid infix expressions, then (a + b), and (a - b) are valid infix expressions.

3. The only valid infix expressions are those defined by steps 1 and 2.

The character string ((5 - 1) + 3) is an example of a complete parenthesized expression. All valid fully parenthesized infix expressions will have at least one operator.

Documentation

Your program must be developed using SPIM. It should be modularized and well commented. The following is a tentative marking scheme and what is expected to be submitted.

1. External Documentation including as many pages as necessary to fulfill the requirements listed below.
    a. Title page.
    b. [10%] Test documentation.
        i. How you tested your program.
        ii. Testing outputs.
    c. [10%] User documentation.
        i. How to run your program.
        ii. Describe parameter (if any).

2. Source Code.
    a. [75% total] Correctness.
        The following expressions will be used for correctness verification.
        i. [5%] (1+2)
        ii. [5%] (1-(3+5))
        iii. [10%] ((5-1) + 3)
        iv. [10%] (4 - (1 - 2))
        v. [10%] ((6-2) + (2-7))
        vi. [15%] (((2+1) - 5) + (8 - 4))
        vii. [20%] ((8+1) - (((3-1) +2) - 3))
    b. [5%] Programming style
        i. Layering.
        ii. Readability.
        iii. Comments.

What to Submit

The following are to be submitted through Blackboard:

1. Copies of all .asm files you created for this exercise as well as
2. Screenshots of the results produced by your solution.

All above listed information must be placed in a Microsoft compressed (zipped) folder (.zip). Your .zip folder must be named: *404 Programming Assignment 2- Your Name*. Marks will be deducted if you do not follow this requirement.