Washington State University Vancouver

CS 360 Precheck

Reverse Polish Notation Calculator

Professor:
Ben McCamish

Overall Assignment

Write a program (in C) targeted at the Linux platform which implements a stack that will be used to evaluate expressions within files written in Reverse Polish Notation. Reverse Polish Notation is a mathematical notation where operators are placed after their operands instead of between them. This means there is no need for parentheses and allows computers to easily solve expressions using a stack.

Example

An expression written in reverse polish notation can look like this:
3 4 5 + /
Which is equivalent to
$3/(4+5)$
Some more examples:
1 2 / 5 − $\implies$ $(1/2)-5$
1 2 − $\implies$ $(1-2)$
2 2 ∗ 5 + 3 − $\implies$ $((2*2)+5)-3$

Your program should print the calculated value of the expression to stdout, for example for the last expression above your program should print 6

Program Interface

Your program will be written in entirely in a single file and must accept a filename as a commandline argument as follows

```
./revpolnot <filename>
```

After which, your program will sift through the file, updating the stack as necessary and performing the required operations. Once this is complete, if the stack has more than one node in it then the expression in the file was invalid and thus you must print "INVALID" to stdout and exit, otherwise print to stdout the last remaining number in the stack.

File Format

The files will contain expressions in reverse polish notation with each character separated by a single space. The only characters present will be numbers and all numbers will be single digit integers from 1-9 and there will only be 4 operators, "+", "*", "-", and "/".

The Structure

This program is to be implemented using a stack. One way to do this is to implement a linked list style node structure with push and pop functions. An example of a structure would be

```
struct node {
    int num;
    struct node* next;
};
```

when encountering a number, you should push to the stack, and when encountering an operator, you must pop the top 2 numbers in the stack, perform the operation, and push the new value back. If you encounter an operator and there are only one or no numbers in the stack then you must print "INVALID" to stdout, free all memory, and exit.

What to turn in (on Autolab):

- A single C file called "revpol.c"