

Project 1: Reverse Polish Calculator

Fall 2019

Due: Sunday, November 3, before 11:59 PM

Abstract

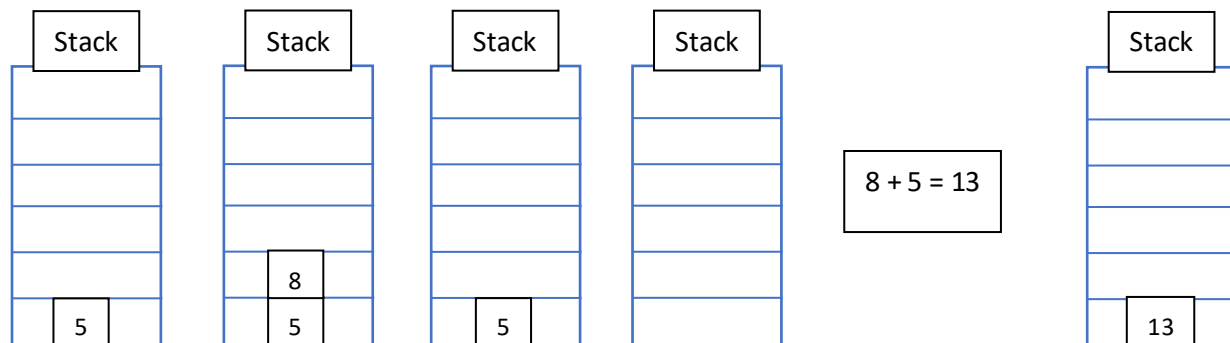
Reverse Polish Notation, also known as Polish postfix notation, is a mathematical notation in which the operands precede the operators. This concept was useful to reduce memory access and utilize a stack to evaluate mathematical expressions. The purpose of this project is to teach students how to implement and use a stack in C, and to show a practical use of it.

Examples

Ex 1- Simple Addition of 5 and 8:

Input → "5 8 +"

Push (5) → Push (8) → Pop () → Pop () → Calculation → Push (result)



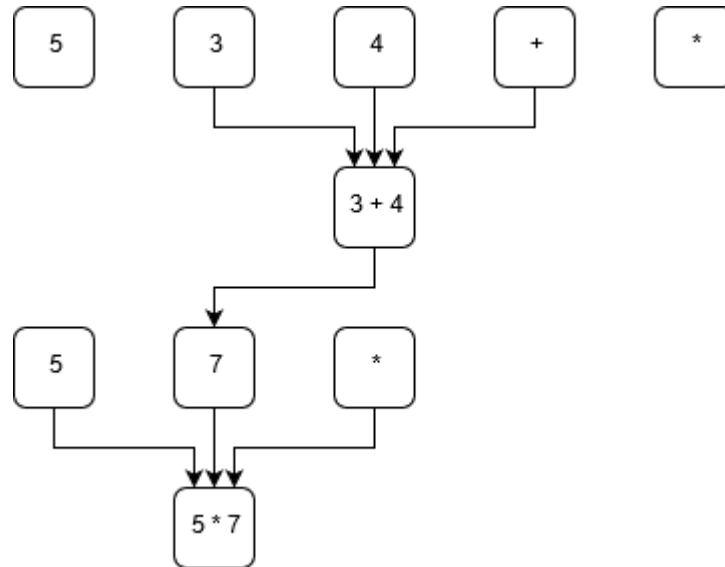
When the program detects an operator (+, -, *, /), it will then pop the top two numbers on the stack and replace it with the result.

Ex 2- Multilayered Operators:

Function: $(3 + 4) * 5 = 35$

Reverse Polish Notation:

$5\ 3\ 4\ +\ * = 35$



Attachment

StackLinkedList.c

Deliverables

Project submissions should include a program file along with screenshots of your output. Your file name should be "Project-1-<LastName>.c"

Project Functionality

One by One Input:

```
int oneByOne ();
```

Description – Function that will accept user input for the stack, computing any operands when provided, until the user indicates they're finished.

Tips – Use a while loop. Consider creating helper functions to increase the modularity and readability of your code. As this function accepts input, it should check when

operators are added to see if the former to entries can be computed. If they can, replace the two entries with a single entry (the result). Moreover, the size of the stack must be at least 2 to calculate the result.

Array input:

```
int arrayInput (char **inputArray);
```

Description – This function will accept an array of strings, push the contents onto a stack, and then calculate the result.

Tips – All arrays used to test your program will be valid, so you only need to worry about checking if the character is an operator, and if it isn't it will be a number. Your function will iterate through the array, pushing each number onto the stack and then performing the Reverse Polish Calculation. Again, consider using helper functions.

Rubric

TOTAL 100 Points

- 50 Points – Code compiles and runs
- 20 Points – Correctly performs operation with single inputs
- 20 Points – Correctly performs operation with array input
- 10 Points – Code is readable and well documented (comments, proper white spacing, good variable/function names)