Class Exercise 4

COSC600 - Advanced Data Structures and Algorithm Analysis

Devere Anthony Weaver

---

1. Evaluate (Compute) the following postfix (reverse Polish) expression using a stack step-by-step).

a) 4 2 + 3 * 4 6 2 / - 1 + 1 * /

Operation	Stack
Place 4 and 2 onto the stack.	2 4

Operation	Stack
$4 + 2 = 6$	6

Operation	Stack
Put 3 onto the stack.	3 6

Operation	Stack
$6 \times 3 = 18$	18

Operation	Stack
Push 4, 6, 2 onto the stack.	2 6 4 18

Operation	Stack
6 / 2 = 3	3 4 18

Operation	Stack
4 - 3 = 1	1 18

Operation	Stack
Place 1 onto the stack.	1 1 18

Operation	Stack
1 + 1 = 2	2 18

Operation	Stack
Put 1 onto the stack.	1 2 18

Operation	Stack
2 * 1 = 2	2 18

Operation	Stack
18 / 2 = 9	9

Thus, the value of the postfix expression is 9.

b) 3 6 7 4 - / + 5 2 1 3 2 - + * - +

Operation	Stack
Place 3, 6, 7, 4 onto the stack	4 7 6 3

Operation	Stack
7 - 4 = 3	3 6 3

Operation	Stack
6 / 3 = 2	2 3

Operation	Stack
3 + 2 = 5	5

Operation	Stack
Put 5, 2, 1, 3, 2 onto the stack.	2 3 1 2 5 5

Operation	Stack
3 - 2 = 1	1 1 2 5 5

Operation	Stack
1 + 1 = 2	2 2 5 5

Operation	Stack
2 * 2 = 4	4 5 9

Operation	Stack
5 - 4 = 1	1 5

Operation	Stack
5 + 1 = 6	6

Therefore, the postfix expression evaluates to 6.

---

2. Convert the following infix expressions to postfix expressions, record the number of push and pop operations, and then evaluate the postfix expressions.

a) 6 − 3 + 2 − 1 + 6 / 3 * 2 − 1 + (3 + 4 / 2 − 1)

Operation	Stack
Push + onto the stack.	-

Display: 63

Operation	Stack
Pop - from the stack. Push + onto stack	+

Display: 63-2

Operation	Stack
Pop + from stack, push - onto stack.	-

Display: 63-2+

Operation	Stack
Push / onto stack.	/
	+

Display: 63-2+1-6

Operation	Stack
Push * onto stack. Pop / from stack.	*
	+

Display: 63-2+1-63/

Operation	Stack
Push * onto stack. Pop / from stack.	*
	+

Display: 63-2+1-63/

Operation	Stack
Push - onto stack. Pop *, + from stack.	-

Display: 63-2+1-63/2*+

Operation	Stack
Push + onto stack. Pop - from stack.	+

Display: 63-2+1-63/2*+1-

Operation	Stack
Push (onto stack.	(+

Display: 63-2+1-63/2*+1-3

Operation	Stack
Push (onto stack.	(+

Display: 63-2+1-63/2*+1-3

Operation	Stack
Push + onto stack.	+ (+

Display: 63-2+1-63/2*+1-3

Operation	Stack
Push / onto stack.	/ + (+

Display: 63-2+1-63/2*+1-34

Operation	Stack
Pop /, + from stack.	(+

Display: 63-2+1-63/2*+1-342/+

Operation	Stack
Pop (from stack.	+

Display: 63-2+1-63/2*+1-342/+1-

Operation	Stack
Pop + from stack.	

Display: 63-2+1-63/2*+1-342/+1-+

Thus the postfix equivalent is: 63-2+1-63/2*+1-342/+1-+.
The total number of push and pop operations is 12.
Using the same stack-based operations as problem 1, the resulting postfix was evaluated to be 11.

(b) (5 - 12 / 3 + 2 * 6 / 3) + (4 * (8 - 5) / 2) + (6 + 10 / (8 – 3 * 2))

Operation	Stack
Push (onto the stack.	(

Display: 5

Operation	Stack
Push - onto the stack.	- (

Display: 5 12

Operation	Stack
Push / onto the stack.	/ - (

Display: 5 12 3

Operation	Stack
Push + onto the stack. Pop / - from stack.	+ (

Display: 5 12 3 / -

Operation	Stack
Push * onto the stack.	* + (

Display: 5 12 3 / - 2 6

Operation	Stack
Push / onto the stack. Pop * from stack.	/ + (

Display: 5 12 3 / - 2 6 * 3

Operation	Stack
Pop everything from stack.	

Display: 5 12 3 / - 2 6 * 3 / +

Operation	Stack
Push + onto stack.	+

Display: 5 12 3 / - 2 6 * 3 / +

Operation	Stack
Push (onto stack.	(+

Display: 5 12 3 / - 2 6 * 3 / +

Operation	Stack
Push (onto stack.	(+

Display: 5 12 3 / - 2 6 * 3 / + 4

Operation	Stack
Push *, (, - onto stack.	- (* (+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5

Operation	Stack
Pop -, (from stack.	- (* (+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 -

Operation	Stack
Push to stack.	/ (+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2

Operation	Stack
Pop /, (from stack.	+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 /

Operation	Stack
Pop + from stack. Push + to stack	+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / +

Operation	Stack
Push (to stack	
	(
	+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / + 6

Operation	Stack
Push +, /, (to stack	(
	/
	+
	(
	+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / + 6 10 8

Operation	Stack
	-
Push - to stack	(
	/
	+
	(
	+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / + 6 10 8 3

Operation	Stack
	*
Push * to stack	-
	(
	/
	+
	(
	+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / + 6 10 8 3 2

Operation	Stack
Pop *, -, (from stack	/
	+
	(
	+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / + 6 10 8 3 2 * -

Operation	Stack
Pop /, +, (from stack	+

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / + 6 10 8 3 2 * - / +

Operation	Stack
Pop + from stack.	

Display: 5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / + 6 10 8 3 2 * - / + +

Thus the postfix is given by:
5 12 3 / - 2 6 * 3 / + 4 8 5 - * 2 / + 6 10 8 3 2 * - / + +

The total number push and pop opeartions is 19. Using the stack-based approach to evaluate the postfix expression, the result is 22.

c) (8 + 6 / (3 − 1) * 2 − 9 / (4 − 1)) − (4 − (8 − 2) / 3)

Operation	Stack
Push (onto stack.	(

Display: 8

Operation	Stack
Push + onto stack.	+ (

Display: 8 6

Operation	Stack
Push / onto stack.	/ + (

Display: 8 6

Operation	Stack
	(
Push (onto stack.	/
	+
	(

Display: 8 6 3

Operation	Stack
	-
Push (onto stack.	(
	/
	+
	(

Display: 8 6 3 1

Operation	Stack
Pop -, (from stack.	/
	+
	(

Display: 8 6 3 1 -

Operation	Stack
Pop / from stack. Push * onto stack.	*
	+
	(

Display: 8 6 3 1 - / 2

Operation	Stack
Pop *, + from stack. Push - onto stack.	-
	(

Display: 8 6 3 1 - / 2 * + 9

Operation	Stack
Push / onto stack.	/
	-
	(

Display: 8 6 3 1 - / 2 * + 9

Operation	Stack
Push (onto stack.	(/ - (

Display: 8 6 3 1 - / 2 * + 9 4

Operation	Stack
Push (onto stack.	- (/ - (

Display: 8 6 3 1 - / 2 * + 9 4 1

Operation	Stack
Pop -, (from stack.	/ - (

Display: 8 6 3 1 - / 2 * + 9 4 1 -

Operation	Stack
Pop /, -, (from stack.	

Display: 8 6 3 1 - / 2 * + 9 4 1 - / -

Operation	Stack
Pop /, -, (from stack.	

Display: 8 6 3 1 - / 2 * + 9 4 1 - / -

Operation	Stack
Push - onto stack.	-

Display: 8 6 3 1 - / 2 * + 9 4 1 - / -

Operation	Stack
Push (, -, (onto stack.	(- (-

Display: 8 6 3 1 - / 2 * + 9 4 1 - / - 4 8

Operation	Stack
Push - onto stack.	- (- (-

Display: 8 6 3 1 - / 2 * + 9 4 1 - / - 4 8 2

Operation	Stack
Pop -, (from stack.	- (-

Display: 8 6 3 1 - / 2 * + 9 4 1 - / - 4 8 2 -

Operation	Stack
Pop /, -, (from stack.	-

Display: 8 6 3 1 - / 2 * + 9 4 1 - / - 4 8 2 - 3 / -

Operation	Stack
Pop - from stack.	

Display: 8 6 3 1 - / 2 * + 9 4 1 - / - 4 8 2 - 3 / - -

Thus, the postfix expression is given by: 8 6 3 1 - / 2 * + 9 4 1 - / - 4 8 2 - 3 / - -
The number of push and pop operations is 16.
Evaluating the postfix results in 9.

3. Write an algorithm in pseudo-code to convert a given positive integer decimal number, n, to a binary number using a stack, not using a recursive function.

Convert positive integer to binary representation:

Input: An arbitrary number n such that $n \in \mathbb{N}$.

Output: The binary representation of n.

Algorithm:

1. Compute $n \, mod \, 2$:
    - if $n \, mod \, 2 = 0$ push 0 onto the stack
    - if $n \, mod \, 2 = 1$ push 1 onto the stack
2. Assign n to the result $\lfloor n/2 \rfloor$ (integer division):
    - if $n = 0$, pop all elements in the stack until empty. The order in which they are popped is the binary representation of n and the algorithm terminates.
    - if $n \neq 0$, repeat 1.