Rakshit Raj (Reg. No. 2017103181) 6th Sep 2020

Department of Electrical and Electronics Engineering

National Institute of Technology Nagaland, IN

Dr. Thithung

Data Structures and Algorithms in C – CS208

Assignment 8

(corresponds to Lecture 9)

Convert expressions from *infix* to *post-fix* using stack.

1. A*(B+C)*D

Adding parenthesis to our expression: (A*(B+C)*D)

	Element	Stack	Post-fix	Description
1.	(	(		START
2.	A	(	A	Add element to post-fix
3.	*	(*	A	PUSH symbol to stack
4.	(	(*(	A	
5.	B	(*(	AB	
6.	+	(*(+	AB	
7.	C		ABC	
8.	)	(*	ABC+	) encountered Pop from stack
9.	*	(**	ABC+	
10.	D	(**	ABC+D	
11.	)		ABC+D**	POP elements from stack upto (
			ABC+D*	

Post-fix expression: ***ABC+D****

2. 5 * (6 + 2) - (12 / 4)

Adding parenthesis to our expression: (5 * (6 + 2) - (12 / 4))

	Elements	Stack	Post-fix	Comment
1.	(	(		START
2.	5	(	5	Add operand to expression
3.	*	(*	5	PUSH operator to stack
4.	(	(*(		
5.	6	(*(	5,6	
6.	+	(*(+	5,6	PUSH + to stack
7.	2	(*(+	5,6,2	
8.	)	(*	5,6,2+	) scanned POP upto (
9.	-	(-	5,6,2+*	- scanned POP all operators with higher precedence
10.	(	(-(	5,6,2+*	
11.	12	(-(	5,6,2+*12	
12.	/	(-(/	5,6,2+*12	PUSH / to stack
13.	4	(-(/	5,6,2+*12,4	
14.	)	(-	5,6,2+*12,4/	POP until (
15.	)		5,6,2+*12,4,/,-	Separated by , for clarity
			5,6,2+*12,4/-	

Post-fix expression: ***5,6,2+*12,4/-***

****