

Uttara University



Assignment

On

“Assignment-02”

Course Title: Data Structures Sessional

Course Code: CSEC 216

Submitted To

Md. Torikur Rahman

Assistant Professor

Department Of Computer
Science & Engineering

Uttara University

Submitted By

Md. Mohammod Islam

ID: 2212081007

Batch: 54A (EVE)

Department Of Computer
Science & Engineering

Uttara University

Submission Date: 15 April 2022
Submission Date: 29 March 2022

Assignment-2:

Introduction:

Both pre- and postfix have basically the same advantages over infix notation. Entirely unambiguous. Infix notation requires precedence and associativity rules to disambiguate it, or addition of extra parentheses that are not usually considered part of the notation. As long as the number of arguments to each operator are known in advance, both prefix and postfix notation are entirely unambiguous.

Consider the following arithmetic expression P, written in postfix notation:

P: 12, 7, 3, -, /, 2, 1, 5, +, *, +

a) Translate P into its equivalent Infix and Prefix expression.

b) Evaluate "P" Using Stack.

❖ **Translate P into its equivalent Postfix to Infix expression & evaluate "P" using Stack.:**

Serial	Scanned Variables	Stack Top	Expressions
1.	12	12	12
2.	7	7	12, 7
3.	3	3	12, 7, 3
4.	-	(7-3)	12, (7-3)
5.	/	(12 / (7-3))	(12 / (7-3))
6.	2	2	(12 / (7-3)), 2
7.	1	1	(12 / (7-3)), 2, 1
8.	5	5	(12 / (7-3)), 2, 1, 5
9.	+	(1 + 5)	(12 / (7-3)), 2, (1 + 5)

10.	*	$2 * (1 + 5)$	$(12 / (7-3)) , 2 * (1 + 5)$
11.	+	$(12 / (7-3)) + (2 * (1 + 5))$	$(12 / (7-3)) + (2 * (1 + 5))$

Postfix to Infix expression is: $(12 / (7-3)) + (2 * (1 + 5))$.

❖ **Translate P into its equivalent Infix to Prefix expression & evaluate “P” using Stack.:**

P: $((12 / (7-3)) + (2 * (1 + 5)))$.

Serial	Scanned Variables	Stack	Expressions
1.	((
2.	5	((5
3.	+	(((+	5
4.	1	(((+	1, 5
5.)	((+, 1, 5
6.	*	((*	+, 1, 5
7.	2	((*	2, +, 1, 5
8.)	(*, 2, +, 1, 5
9.	+	(+	*, 2, +, 1, 5
10.	((+ (*, 2, +, 1, 5
11.	((+ ((*, 2, +, 1, 5
12.	3	(+ ((3, *, 2, +, 1, 5
13.	-	(+ ((-	3, *, 2, +, 1, 5
14.	7	(+ ((-	7, 3, *, 2, +, 1, 5
15.)	(+ (-, 7, 3, *, 2, +, 1, 5
16.	/	(+ (/	-, 7, 3, *, 2, +, 1, 5
17.	12	(+ (/	12, -, 7, 3, *, 2, +, 1, 5
18.)	(+	/, 12, -, 7, 3, *, 2, +, 1, 5
19.)		+, /, 12, -, 7, 3, *, 2, +, 1, 5

Infix to Prefix expression is: $+, /, 12, -, 7, 3, *, 2, +, 1, 5$.

Conclusion:

Although Postfix and Prefix notations have similar complexity, Postfix is slightly easier to evaluate in simple circumstances. We understand Infix to Prefix expression and Postfix to Infix expression.