CSE-213
(Data Structure)

Lecture on

Arithmetic expression (Infix, prefix postfix, and conversion) using stack and queue

Md. Jalal Uddin

Lecturer

Department of CSE

City University

Email: jalalruice@gmail.com

No: 01717011128 (Emergency Call)

Department of Computer Science & Engineering (CSE)
City University, Khagan, Birulia, Savar, Dhaka-1216, Bangladesh

STACKS: Arithmetic Expressions;

- ✔ Let Q be an arithmetic expression involving constants and operation.

Q: 2 ↑3+5*2 ↑2-12/6, we have to find the value of Q

- ✔ Q may have different levels of precedence in its binary operations.
- ✔ We assume the following three levels of precedence for the usual five binary operations.
  - Highest: Exponential (↑)
  - Next Highest: Multiplication (*) and division(/)
  - Lowest: Addition (+) and Subtraction(-)
- ✔ Thus, we obtain after exponentiations (8+5*4-12/6)
- ✔ After, multiplication and division (8+20-2)
- ✔ After. Addition and subtraction , 26

STACKS: Polish Notation

For most common arithmetic operations:

A+B, C-D, E*F, G/H (*infix notation*)

Polish notation refers to the notation in which the operator symbol is placed before its two operands.

For example: +AB, -CD, *EF, /GH

Instant Exercise: (***Infix expression*** to ***polish notation***)

(A+B)*C = ?
= [+AB]*C=?
= *+ABC

A+(B*C) = ?
= A+[*BC]=?
= +A*BC

STACKS: Polish Notation

(A+B)/(C-D) = ?

= +AB/[-CD] = ?

= /+AB-CD

- ✔ The fundamental property of polish notation is that the order in which the operations are to be performed is completely determined by the positions of the operators and operands in the expression.
- ✔ One never needs parentheses when writing expressions in polish notation.

STACKS: Reverse Polish Notation

Reverse Polish Notation refers to the analogous notation in which the operator symbol is placed after its two operands:

AB+, CD-, EF*, GH/

- ✔ This notation is frequently called ***postfix notation***
- ✔ One never needs parentheses to determine the order of the operations in any arithmetic expression in ***postfix notation***

STACKS: Evaluation of Postfix Expression

P: 5, 6, 2, +, *, 12, 4, /, -

Symbol Scanned	STACK
(1) 5	5
(2) 6	5, 6
(3) 2	5, 6, 2
(4) +	5, 8
(5) *	40
(6) 12	40, 12
(7) 4	40, 12, 4
(8) /	40, 3
(9) -	37
(10))	

STACKS: Transforming Q into P expression

Q: A + (B * C- (D / E ↑ F) * G) * H = (?) P expression

Symbol Scanned	Stack	Expression P
(1) A	(	A
(2) +	(+	A
(3) (	(+ (	A
(4) B	(+ (	A B
(5) *	(+ (*	A B
(6) C	(+ (*	A B C
(7) -	(+ (-	A B C *
(8) (	(+ (- (	A B C *
(9) D	(+ (- (	A B C * D
(10) /	(+ (- (/	A B C * D
(11) E	(+ (- (/	A B C * D E
(12) ↑	(+ (- (/ ↑	A B C * D E
(13) F	(+ (- (/ ↑	A B C * D E F
(14))	(+ (-	A B C * D E F ↑ /
(15) *	(+ (- *	A B C * D E F ↑ /

STACKS: Transforming Q into P expression

Q: A + (B * C- (D / E ↑ F) * G) * H = (?) P expression

Symbol Scanned	Stack	Expression P
(7) -	(+ (-	A B C *
(8) (	(+ (- (	A B C *
(9) D	(+ (- (	A B C * D
(10) /	(+ (- (/	A B C * D
(11) E	(+ (- (/	A B C * D E
(12) ↑	(+ (- (/ ↑	A B C * D E
(13) F	(+ (- (/ ↑	A B C * D E F
(14))	(+ (-	A B C * D E F ↑ /
(15) *	(+ (- *	A B C * D E F ↑ /
(16) G	(+ (- *	A B C * D E F ↑ / G
(17))	(+	A B C * D E F ↑ / G * -
(18) *	(+ *	A B C * D E F ↑ / G *
(19) H	(+ *	A B C * D E F ↑ / G * - H
(20))		A B C * D E F ↑ / G * - H * +

Instant TEST

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

Which Expression?

Postfix or Prefix

✔ **Postfix**

Equivalent infix expression ?

P = 12, [7-3], / , 2 ,1, 5, +, *, +

= [12/(7-3)], 2, [1+5], *, +

= [12/(7-3)], [2* (1+5)], +

= 12/(7-3) + 2* (1+5)

Result: 15