GIT Department of Computer Engineering

CSE 222 - Spring 2020

Homework 4-Q1

Ali BAHAR

171044066

Note:Right hand side has been considered as top of the stack.

A + ((B – C * D) / E) + F – G / H

Infix to postfix.

Postfix	Stack	Token	Operations
A		A	Append(‘A’)
A	+	+	Push(‘+’)
A	+ (	(	Push(‘(‘)
A	+ ((	(	Push(‘(‘)
AB	+ ((	B	Append(‘B’)
AB	+ ((-	-	Push(‘-‘)
ABC	+ ((-	C	Append(‘C’)
ABC	+ ((- *	*	Push(‘*’)
ABCD	+ ((- *	D	Append(‘D’)
ABCD*-	+ (	)	Pop(‘*’) Append(‘*’) Pop(‘-’) Append(‘-’) Pop(‘(’)
ABCD*-	+ (/	/	Push(‘/’)
ABCD*-E	+ (/	E	Append(‘E’)
ABCD*-E/	+	)	Pop(‘/’) Append(‘/’) Pop(‘(‘)
ABCD*-E/+	+	+	Pop(‘+’) Append(‘+’) Push(‘+’)
ABCD*-E/+F	+	F	Append(‘F’)
ABCD*-E/+F+	-	-	Pop(‘+’) Append(‘+’) Push(‘-’)
ABCD*-E/+F+G	-	G	Append(‘G’)
ABCD*-E/+F+G	-/	/	Push(‘/’)
ABCD*-E/+F+GH	-/	H	Append(‘H’)
ABCD*-E/+F+GH/-		End Of The Tokens	Pop(‘/’) Append(‘/’) Pop(‘-’) Append(‘-’)

Postfix:

ABCD*-E/+F+GH/-

Evaluation of postfix expression.

ABCD*-E/+F+GH/-

Let A = 10 , B = 20 , C = 4 , D = 2 ,E = 6 , F = 11 , G = 14 , H = 7

Expression	Action	Stack
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Push 10	10
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Push 20	10 20
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Push 4	10 20 4
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Push 2	10 20 4 2
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Pop 2 and 4 Evaluate 4*2 Push 8	10 20 8
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Pop 8 and 20 Evaluate 20-8 Push 12	10 12
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Push 6	10 12 6
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Pop 6 and 12 Evaluate 12/6 Push 2	10 2
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Pop 2 and 10 Evaluate 10+2 Push 12	12
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Push 11	12 11
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Pop 11 and 12 Evaluate 12 + 11 Push 23	23
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Push 14	23 14
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Push 7	23 14 7
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Pop 7 and 14 Evaluate 14/7 Push 2	23 2
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Pop 2 and 23 Evaluate 23-2 Push 21	21
10 20 4 2 * - 6 / + 11 + 14 7 / - ↑	Pop 21 Stack is empty Result is 21	

A + ((B – C * D) / E) + F – G / H

Infix to prefix.

Prefix	Stack	Token	Operations
H		H	Insert(0 , 'H')
H	/	/	Push('/')
GH	/	G	Insert(0 , 'G')
/GH	-	-	Pop('/') Insert(0 , '/') Push('-')
F/GH	-	F	Insert(0 , 'F')
-F/GH	+	+	Pop('-') Insert(0 , '-') Psuh('+')
-F/GH	+)	)	Push(')')
E-F/GH	+)	E	Insert(0 , 'E')
E-F/GH	+) /	/	Push('/')
E-F/GH	+) /)	)	Push(')')
DE-F/GH	+) /)	D	Insert(0 , 'D')
DE-F/GH	+) /) *	*	Push('*')
CDE-F/GH	+) /) *	C	Insert(0 , 'C')
*CDE-F/GH	+) /) -	-	Push('-')
B*CDE-F/GH	+) /) -	B	Insert(0 , 'B')
-B*CDE-F/GH	+) /	(	Pop('-') Insert(0 , '-') Pop(')')
/-B*CDE-F/GH	+	(	Pop('/') Insert(0 , '/') Pop(')')
+/-B*CDE-F/GH	+	+	Pop('+') Insert(0 , '+') Push('+')
A+/-B*CDE-F/GH	+	A	Insert(0 , 'A')
+A+/-B*CDE-F/GH		End Of The Tokens	Pop('+') Insert(0 , '+')

Prefix :

+A+/-B*CDE-F/GH

Evaluation of prefix expression.

+A+/-B*CDE-F/GH

Let A = 10 , B = 20 , C = 4 , D = 2 ,E = 6 , F = 11 , G = 14 , H = 7

Expression	Action	Stack
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Push 7	7
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Push 14	7 14
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Pop 14 and 7 Evaluate 14/7 Push 2	2
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Push 11	2 11
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Pop 11 and 2 Evaluate 11-2 Push 9	9
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Push 6	9 6
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Push 2	9 6 2
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Push 4	9 6 2 4
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Pop 4 and 2 Evaluate 4*2 Push 8	9 6 8
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Push 20	9 6 8 20
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Pop 20 and 8 Evaluate 20 – 8 Push 12	9 6 12
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Pop 12 and 6 Evaluate 12/6 Push 2	9 2
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Pop 2 and 9 Evaluate 9+2 Push 11	11
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Push 10	11 10
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Pop 10 and 11 Evaluate 10 + 11 Push 21	21
+ 10 + / - 20 * 4 2 6 – 11 / 14 7 ↑	Pop 21 Stack is empty Result is 21	

! (A && ! ((B < C) || (C > D))) || (C < E)

Infix to postfix.

Postfix	Stack	Token	Operations
	!	!	Push('!')
	! (	(	Push('(')
A	! (	A	Append('A')
A	! (&&	&&	Push('&&')
A	! (&& !	!	Push('!')
A	! (&& ! (	(	Push('(')
A	! (&& ! ((	(	Push('(')
A B	! (&& ! ((	B	Append('B')
A B	! (&& ! ((<	<	Push('<')
A B C	! (&& ! ((<	C	Append('C')
A B C <	! (&& ! (	)	Pop('<') Append('<') Pop('(')
A B C <	! (&& ! (||	||	Push('||')
A B C <	! (&& ! (|| (	(	Push('(')
A B C < C	! (&& ! (|| (	C	Append('C')
A B C < C	! (&& ! (|| (>	>	Push('>')
A B C < C D	! (&& ! (|| (>	D	Append('D')
A B C < C D >	! (&& ! (||	)	Pop('>') Append('>') Pop('(')
A B C < C D > ||	! (&& !	)	Pop('||') Append('||') Pop('(')
A B C < C D > || ! &&	!	)	Pop('!') Append('!') Pop('&&') Append('&&') Pop('(')
A B C < C D > || ! && !	||	||	Pop('!') Append('!') Push('||')
A B C < C D > || ! && !	||(	(	Push('(')
A B C < C D > || ! && ! C	||(	C	Append('C')
A B C < C D > || ! && ! C	||(<	<	Push('<')
A B C < C D > || ! && ! C E	||(<	E	Append('E')
A B C < C D > || ! && ! C E <	||	)	Pop('<') Append('<') Pop('(') Push('||')
A B C < C D > || ! && ! C E < ||		End Of The Tokens	Pop('||') Append('||')

Postfix :

A B C < C D > || ! && ! C E < ||

Evaluation of postfix expression.

A B C < C D > || ! && ! C E < ||

Let A = 1 , B = 20 , C = 4 , D = 2 ,E = 6

Expression	Action	Stack
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Push 1	1
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Push 20	1 20
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Push 4	1 20 4
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 4 20 Evaluate 20<4 Push 0	1 20 0
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Push 4	1 0 4
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Push 2	1 0 4 2
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 2 and 4 Evaluate 4>2 Pop 1	1 0 1
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 1 and 0 Evaluate 0 || 1 Push 1	1 1
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 1 Evaluate !1 Push 0	1 0
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 0 and 1 Evaluate 1 &&0 Push 0	0
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 0 Evaluate !0 Push 1	1
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Push 4	1 4
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Push 6	1 4 6
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 6 and 4 Evaluate 4<6 Push 1	1 1
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 1 and 1 Evaluate 1||1 Push 1	1
1 20 4 < 4 2 > || ! && ! 4 6 < || ↑	Pop 1 Stack is empty Result is 1	

!(A && ! ((B < C) || (C > D))) || (C < E)

Infix to prefix.

Prefix	Stack	Token	Operations
	)	)	Push(')')
E	)	E	Insert(0 , 'E')
E	) <	<	Push('<')
C E	) <	C	Insert(0 , 'C')
< C E		(	Pop('<') Insert(0 , '<') Pop(')')
< C E	||	||	Push('||')
< C E	||)	)	Push(')')
< C E	||))	)	Push(')')
< C E	||)))	)	Push(')')
D < C E	||)))	D	Insert(0 , 'D')
D < C E	||)))>	>	Push('>')
C D < C E	||)))>	C	Insert(0 , 'C')
> C D < C E	||))	(	Pop('>') Insert(0 , '>') Pop(')')
> C D < C E	||)) ||	||	Push('||')
> C D < C E	||)) ||)	)	Push (')')
C > C D < C E	||)) ||)	C	Insert(0 , 'C')
C > C D < C E	||)) ||) <	<	Push('<')
B C > C D < C E	||)) ||) <	B	Insert(0 , 'B')
< B C > C D < C E	||)) ||	(	Pop('<') Insert(0 , '<') Pop(')')
|| < B C > C D < C E	||)	(	Pop('||') Insert(0 , '||') Pop(')')
|| < B C > C D < C E	||) !	!	Push('!')
! || < B C > C D < C E	||) &&	&&	Pop('!') Insert(0 , '!') Push('&&')
A ! || < B C > C D < C E	||) &&	A	Insert(0 , 'A')
&& A ! || < B C > C D < C E	||	(	Pop('&&') Insert(0 , '&&') Pop(')')
&& A ! || < B C > C D < C E	|| !	!	Push('!')
|| ! && A ! || < B C > C D < C E		End Of The Tokens	Pop('!') Insert(0 , '!') Pop('||') Insert(0 , '||')

Prefix :

|| ! && A ! || < B C > C D < C E

Evaluation of prefix expression.

|| ! && A ! || < B C > C D < C E

Let A = 1 , B = 20 , C = 4 , D = 2 ,E = 6

Expression	Action	Stack
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Push 6	6
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Push 4	6 4
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 4 and 6 Evaluate 4<6 Push 0	0
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Push 2	0 2
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Push 4	0 2 4
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 4 and 2 Evaluate 4>2 Push 1	0 1
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Push 4	0 1 4
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Push 20	0 1 4 20
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 20 and 4 Evaluate 20<4 Push 0	0 1 0
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 0 and 1 Evaluate 0||1 Push 1	0 1
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 1 Evaluate !1 Push 0	0 0
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Push 1	0 0 1
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 1 and 0 Evaluate 1 && 0 Push 0	0 0
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 0 Evaluate !0 Push 1	0 1
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 0 and 1 Evaluate 0||1 Push 1	1
|| ! && 1 ! || < 20 4 > 4 2 < 4 6 ↑	Pop 1 Stack is empty Result is 1	