HARUN ALBAYRAK – 171044014 – Homework 4

CSE 222/505 – Computer Engineering

Q1)

i) $A + ((B - C * D) / E) + F - G / H$

Next token	Stack	Postfix
A		**A**
+	+	**A**
(	+ (	**A**
(	+ ((	**A**
B	+ ((	**AB**
-	+ ((-	**AB**
C	+ ((-	**ABC**
*****	+ ((- *	**ABC**
D	+ ((- *	**ABCD**
)	+ (	**ABCD*-**
/	+ (/	**ABCD*-**
E	+ (/	**ABCD*-E**
)	+	**ABCD*-E/**
+	+	**ABCD*-E/+**
F	+	**ABCD*-E/+F**
-	-	**ABCD*-E/+F+**
G	-	**ABCD*-E/+F+G**
/	- /	**ABCD*-E/+F+G**
H	- /	**ABCD*-E/+F+GH**
End	-	**ABCD*-E/+F+GH/**
End		**ABCD*-E/+F+GH/-**

Next token	Operation	Stack
A	Push	1
B	Push	1,22
C	Push	1,22,3
D	Push	1,22,3,4
*****	Pop(2 el.) & evaluate	1,22,3*4
-	Pop(2 el.) & evaluate	1,22-12
E	Push	1,10,5
/	Pop(2 el.) & evaluate	1,(10/5)
+	Pop(2 el.) & evaluate	(1+2)
F	Push	3,6
+	Pop(2 el.) & evaluate	(3+6)
G	Push	9,16
H	Push	9,16,8
/	Pop(2 el.) & evaluate	9,2
-	Pop(2 el.) & evaluate	7

The postfix = ABCD*-E/+F+GH/-
A=1,B=22,C=3,D=4,E=5,F=6,H=16,H=8

The postfix evaluation = (((A+((B-C*D) / E))+F)-(G/H))
The postfix evaluation = (((1+((22-3*4) / 5))+6)-(16/8)) = 7

Infix) A + ((B – C * D) / E) + F – G / H

Reverse Infix) H / G – F +) E /) D * C – B ((+ A

Next token	Stack	Postfix
H		H
/	/	H
G	/	HG
-	-	HG/
F	-	HG/F
+	- +	HG/F
)	- +)	HG/F
E	- +)	HG/FE
/	- +) /	HG/FE
)	- +) /)	HG/FE
D	- +) /)	HG/FED
*	- +) /) *	HG/FED
C	- +) /) *	HG/FEDC
-	- +) /) -	HG/FEDC*
B	- +) /) -	HG/FEDC*B
(	- +) /	HG/FEDC*B-
(	- +	HG/FEDC*B-/
+	- + +	HG/FEDC*B-/
A	- + +	HG/FEDC*B-/A
End	-	HG/FEDC*B-/A++
End	- + +	HG/FEDC*B-/A++-

Next token	Operation	Stack
H	Push	8
G	Push	8,16
/	Pop(2 el.) & evaluate	(16/8)
F	Push	2,6
E	Push	2,6,5
D	Push	2,6,5,4
C	Push	2,6,5,4,3
*	Pop(2 el.) & evaluate	2,6,5,(3*4)
B	Push	2,6,5,12,22
-	Pop(2 el.) & evaluate	2,6,5,(22-12)
/	Pop(2 el.) & evaluate	2,6,(10/5)
A	Push	2,6,2,1
+	Pop(2 el.) & evaluate	2,6,3
+	Pop(2 el.) & evaluate	2,9
-	Pop(2 el.) & evaluate	7

The prefix(reverse) = HG/FEDC*B-/A++-

The prefix = -++A/-B*CDEF/GH

A=1,B=22,C=3,D=4,E=5,F=6,H=16,H=8

The prefix evaluation = (((A+((B-C*D)/E))+F)-(G-H))
The postfix evaluation = (((1+((22-3*4) / 5))+6)-(16/8)) = 7

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

Next token	Stack	Postfix
!	!	
(	! (	
A	! (	**A**
&&	! (&&	**A**
!	! (&& !	**A**
(	! (&& ! (	**A**
(	! (&& ! ((	**A**
B	! (&& ! ((	**AB**
<	! (&& ! ((<	**AB**
C	! (&& ! ((<	**ABC**
)	! (&& ! (	**ABC<**
||	! (&& ! (||	**ABC<**
(	! (&& ! (|| (	**ABC<**
C	! (&& ! (|| (	**ABC<C**
>	! (&& ! (|| (>	**ABC<C**
D	! (&& ! (|| (>	**ABC<CD**
)	! (&& ! (||	**ABC<CD>**
)	! (&& !	**ABC<CD>||**
)	!	**ABC<CD>||!&&**
||	||	**ABC<CD>||!&&!**
(	|| (	**ABC<CD>||!&&!**
C	|| (	**ABC<CD>||!&&!C**
<	|| (<	**ABC<CD>||!&&!C**
E	|| (<	**ABC<CD>||!&&!CE**
)	||	**ABC<CD>||!&&!CE<**
End		**ABC<CD>||!&&!CE<||**

Next token	Operation	Stack
A	Push	1
B	Push	1,2
C	Push	1,2,3
<	Pop(2 el.) & evaluate	1,(2<3)
C	Push	1,0,3
D	Push	1,0,3,4
>	Pop(2 el.) & evaluate	1,0,(3>4)
||	Pop(2 el.) & evaluate	1, (0||0)
!	Pop & evaluate	1, !(0)
&&	Pop(2 el.) & evaluate	(1&&1)
!	Pop & evaluate	!(1)
C	Push	0,3
E	Push	0,3,5
<	Pop(2 el.) & evaluate	0,(3<5)
||	Pop(2 el.) & evaluate	1

The postfix = ABC<CD>||!&&!CE<||
A=1,B=2,C=3,D=4,E=5

The postfix evaluation = ((!(A&&!((B<C)||(C>D))))||(C<E))
The postfix evaluation = ((!(1&&!((2<3)||(3>4))))||(3<5))=1

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

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

Next token	Stack	Postfix
)	)	
E	)	E
<	) <	E
C	) <	EC
(		EC<
||	||	EC<
)	||)	EC<
)	||))	EC<
)	||)))	EC<
D	||)))	EC<D
>	||))) >	EC<D
C	||))) >	EC<DC
(	||))	EC<DC>
||	||)) ||	EC<DC>
)	||)) ||)	EC<DC>
C	||)) ||)	EC<DC>C
<	||)) ||) <	EC<DC>C
B	||)) ||) <	EC<DC>CB
(	||)) ||	EC<DC>CB<
(	||)	EC<DC>CB<||
!	||) !	EC<DC>CB<||
&&	||) &&	EC<DC>CB<||!
A	||) &&	EC<DC>CB<||!A
(	||	EC<DC>CB<||!A&&
!	|| !	EC<DC>CB<||!A&&
End	||	EC<DC>CB<||!A&&!
End		EC<DC>CB<||!A&&!||

Next token	Operation	Stack
E	Push	5
C	Push	5,3
<	Pop(2 el.) & evaluate	(3<5)
D	Push	1,4
C	Push	1,4,3
>	Pop(2 el.) & evaluate	1,(3>4)
C	Push	1,0,3
B	Push	1,0,3,2
<	Pop(2 el.) & evaluate	1,0,(2<3)
||	Pop(2 el.) & evaluate	1,(0||1)
!	Pop(2 el.) & evaluate	1,1
A	Push	1,1,1
&&	Push	1,1
!	Pop(2 el.) & evaluate	1,0
||	Pop(2 el.) & evaluate	1

The prefix evaluation

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

The prefix evaluation

=((!(1&&!((2<3)||(3>4))))||(3<5))

= 1

The prefix(reverse) = EC<DC>CB<||!A&&!||

A=1,B=2,C=3,D=4,E=5
The prefix = ||!&&A!||<BC>CD<CE