GTU Department of Computer Engineering

CSE 222/505 - Spring 2020

Homework 4-Q1

Mert Can BEŞİRLİ
1801042663

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

Token	Action	Result	Stack	Notes
A	Add A to the result	A		
+	Push + to stack	A	+	
(	Push (to stack	A	(+	
(	Push (to stack	A	((+	
B	Add B to the stack	AB	((+	
-	Push – to stack	AB	-((+	
C	Add C to the result	ABC	-((+	
*	Push * to stack	ABC	*-((+	* has higher predence than -
D	Add D to the stack	ABCD	*-((+	
)	Pop * from stack and add result Pop – from stack and add result Pop (from stack	ABCD* ABCD*- ABCD*-	-((+ ((+ (+	Do process until (is popped from stack
/	Push / to stack	ABCD*-	/(+	
E	Add E to the result	ABCD*-E	/(+	
)	Pop / from stack and add result Pop (from stack	ABCD*-E/ ABCD*-E/	(+ +	Do process until (is popped from stack
+	Pop + from stack and add result	ABCD*-E/+		+ same predece of +

	Push + to stack	ABCD*-E/+	+	
F	Add F to the result	ABCD*-E/+F	+	
-	Pop + from stack and add result Push – to stack	ABCD*-E/+F+ ABCD*-E/+F+	 -	- lower predence than +
G	Add G to the result	ABCD*-E/ +F+G	-	
/	Push / to stack	ABCD*-E/ +F+G	/-	/ higher predence than -
H	Add H to the result	ABCD*-E/ +F+GH	/-	
	Pop / from stack and add result Pop – from stack and add result	ABCD*-E/ +F+GH/ ABCD*-E/ +F+GH/-	-	Given expression is iterated, do process till stack is not empty, it will give final result
Postfix Expression: ABCD*-E/+F+GH/-				

For prefix, reverse given expression then apply algorithm of infix to postfix expression then reverse the expression

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

Token	Action	Result	Stack	Notes
H	Add H to the result	H		
/	Push / to stack	H	/	
G	Add H to the result	HG	/	
-	Pop / from stack and add result Push – to stack	HG/ HG/	 -	- lower predence than /
F	Add F to the	HG/F	-	

	result			
+	Pop – from stack and add result Push + to stack	HG/F- HG/F-	 +	+ same predence -
(	Push (to stack	HG/F-	(+	
E	Add E to the result	HG/F-E	(+	
/	Push / to stack	HG/F-E	/(+	
(	Push (to stack	HG/F-E	(/(+	
D	Add D to the result	HG/F-ED	(/(+	
*	Push * to stack	HG/F-ED	*(/(+	
C	Add C to the result	HG/F-EDC	*(/(+	
-	Pop * from stack and add result Push – to stack	HG/F-EDC* HG/F-EDC*	(/(+ -(/(+	- lower predence than *
B	Add B to the result	HG/F-EDC*B	-(/(+	
)	Pop – from stack and add result Pop (from stack	HG/F-EDC*B- HG/F-EDC*B-	(/(+ /(+	Do process until (is popped from stack
)	Pop / from stack and add result Pop (from stack	HG/F-EDC*B-/ HG/F-EDC*B-/	(+ +	Do process until (is popped from stack
+	Pop + from stack and add result Push + to stack	HG/F-EDC*B-/ + HG/F-EDC*B-/	 +	+ same predence +

		+		
A	Add A to the result	HG/F-EDC*B-/+A	+	
	Pop + from stack and add result	HG/F-EDC*B-/+A+		Given expression is iterated, do process till stack is not empty, it will give final result
Posfix Expression HG/F-EDC*B-/+A+ then reverse it **Prefix Expression --> +A+/-B*CDE-F/GH**				

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

Token	Action	Result	Stack	Notes
!	Push ! To stack		!	
(	Push (to stack		(!	
A	Add A to the result	A	(!	
&&	Push && to stack	A	&&(!	
!	Push ! To stack	A	!&&(!	! higher predence &&
(	Push (to stack	A	(!&&(!	
(	Push (to stack	A	((!&&(!	
B	Add B to the result	AB	((!&&(!	
<	Push < to stack	AB	<((!&&(!	
C	Add C to the result	ABC	<((!&&(!	
)	Pop < from stack and add result Pop (from stack	ABC< ABC<	((!&&(! (!&&(!	Do process until (is popped from stack
||	Push || to stack	ABC<	||(!&&(!	
(	Push (to stack	ABC<	(||(!&&(!	
C	Add C to the result	ABC<C	(||(!&&(!	
>	Push > to stack	ABC<C	>(||(!&&(!	
D	Add D to the result	ABC<CD	>(||(!&&(!	

)	Pop > from stack and add result Pop (from stack	ABC<CD> ABC<CD>	(||(!&&(! ||(!&&(!	Do process until (is popped from stack
)	Pop || from stack and add result Pop (from stack	ABC<CD>|| ABC<CD>||	(!&&(! !&&(!	Do process until (is popped from stack
)	Pop ! From stack and add result Pop && from stack and add result Pop (from stack	ABC<CD>||! ABC<CD>||! && ABC<CD>||! &&	&&(! (! !	Do process until (is popped from stack
||	Pop ! From stack and add result Push || from stack	ABC<CD>||! &&! ABC<CD>||! &&!	 ||	|| lower predence than !
(	Push (to stack	ABC<CD>||! &&!	(||	
C	Add C to the result	ABC<CD>||! &&!C	(||	
<	Push < to stack	ABC<CD>||! &&!C	<(||	
E	Add E to the result	ABC<CD>||! &&!CE	<(||	
)	Pop < from stack and add result Pop (from stack	ABC<CD>||! &&!CE< ABC<CD>||! &&!CE<	(|| ||	Do process until (is popped from stack
	Pop || from stack and add result	ABC<CD>||! &&!CE<||		Given expression is iterated, do

				process till stack is not empty, it will give final result
Posfix Expression: ABC<CD>||!&&!CE<||				

For prefix, reverse given expression then apply algorithm of infix to postfix expression then reverse the expression

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

(	Push (to stack		(	
E	Add E to the result	E	(	
>	Push > to stack	E	>(	
C	Add C to stack	EC	>(	
)	Pop > from stack and add result Pop (from stack	EC> EC>	(	Do process until (is popped from stack
||	Push || to stack	EC>	||	
(	Push (to stack	EC>	(||	
(	Push (to stack	EC>	((||	
(	Push (to stack	EC>	(((||	
D	Add D to the result	EC>D	(((||	
<	Push < to stack	EC>D	<(((||	
C	Add C to the result	EC>DC	<(((||	
)	Pop < from stack and add result Pop (from stack	EC>DC< EC>DC<	(((|| ((||	Do process until (is popped from stack
||	Push || to stack	EC>DC<	||((||	
(	Push (to stack	EC>DC<	(||((||	

C	Add C to the result	EC>DC<C	(||((||	
>	Push > to stack	EC>DC<C	>(||((||	
B	Add B to the result	EC>DC<CB	>(||((||	
)	Pop > from stack and add result Pop (from stack	EC>DC<CB> EC>DC<CB>	(||((|| ||((||	Do process until (is popped from stack
)	Pop || from stack and add result Pop (from stack	EC>DC<CB>|| EC>DC<CB>||	((|| (||	Do process until (is popped from stack
!	Push ! To stack	EC>DC<CB>||	!(||	
&&	Pop ! From stack and add result Push && to stack	EC>DC<CB>||! EC>DC<CB>||!	(|| &&(||	&& lower predence than !
A	Add A to the result	EC>DC<CB>||!A	&&(||	
)	Pop && from stack and add result Pop (from stack	EC>DC<CB>||!A&& EC>DC<CB>||!A&&	(|| ||	Do process until (is popped from stack
!	Push ! To stack	EC>DC<CB>||!A&&	!||	! higher predence than ||
	Pop ! From stack and add result Pop || from stack and add result	EC>DC<CB>||!A&&! EC>DC<CB>||!A&&!||	||	Given expression is iterated, do process till stack is not empty, it will give final result

Postfix expression: EC>DC<CB>||!A&&!|| then reverse it
Prefix Expression --> ||!&&A!||<BC>CD<CE