## CSE 222 HW4

Melihcan Çilek 1801042092

13 April 2020

## Question 1:

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

Assume that $A = 1, B = 2, C = 3, D = 4, E = 5, F = 6, G = 7, H = 8$	1 + ((2 - 3 * 4) / 5) + 6 - 7/8	$\begin{array}{c} 1+(\ (2-3\ ^*4)\ /\ 5)+6-7/8\\ 1+(\ (2-12)\ /\ 5)+6-7/8\\ 1+(\ (\textbf{-10})\ /\ 5)+6-7/8\\ 1+(\ \textbf{-2}+6)-7/8\\ 1+4-7/8\\ 1+4-0.875\\ 1+3.125\\ 4.125 \end{array}$
--	---------------------------------	---

			4.125
Expression	Stack	Postfix(String)	Explanation
Α	(	Δ	First character came to
A	(	A	the postfix
			Because of '+' is a symbol
			not a letter, we push it to our
+	(+	A	stack and because of priority of
			'(' less than '+', we push it to our stack
,	, ,		If '(' comes, we push it to stack so we
(	(+(	A	push '(' character stack again
_	, ,,		B is a letter so we append it end of
В	(+((	A B	the postfix
			Character '-'s priority is higher than '('
_	(+((-	A B	character which is peek of the stack
	( ) ((		so we push '-' to the stack
			C is a letter so we append it end of
$\mathbf{C}$	(+((-	ABC	the postfix
			Character '*'s priority is higher than '-'
*	(   (( *	АВС	
	(+((-*	ADC	character which is peek of the stack
			so we push '*' to the stack
D	(+((-*	ABCD	D is a letter so we append it end of
			the postfix
`	, ,		Because ')' character came, we pop all
)	(+(	A B C D *-	symbols to end of the postfix till we see
			'(' character.
			Character '/'s priority is higher than '('
/	(+(/	A B C D *-	character which is peek of the stack
			so we push '/' to the stack
E	(+(/	A B C D *- E	E is a letter so we append it end of
D	(+(/	ABOD - E	the postfix
			Because ')' character came, we pop all
)	(+	A B C D *- E /	symbols to end of the postfix till we see
			'(' character.
			Priority of '+' which is peek of the stack
+	(++	A B C D *- E /	and '+' which is next character's priorities
	`	,	are the same so we can push it to stack.
		1 D C D * D / D	F is a letter so we append it end of
F	(++	A B C D *- E / F	the postfix
			'-'s priority is lower than plus because
	,		while calculating, order of '-' operation is
-	(-	A B C D *- E / F + +	important so we pop both '+' operations
			from stack to end of the postfix
			G is a letter so we append it end of
G	(-	A B C D *- E / F + + G	the postfix
			Character '/'s priority is higher than '-'
/	( /	A B C D *- E / F + + G	character which is peek of the stack
/	(-/	$\begin{bmatrix} \mathbf{A} \mathbf{B} \mathbf{C} \mathbf{D} & \mathbf{E} / \mathbf{F} + \mathbf{G} \end{bmatrix}$	_
			so we push '/' to the stack  H is a letter so we append it and of
H	(-/	A B C D *- E / F + + G H	H is a letter so we append it end of
	` '	,	the postfix
			Because ')' character came, we pop all
`		A D C D * D / D	symbols to end of the postfix till we see
)		A B C D *- E / F + + G H / -	'(' character.Our expression is over and
			our stack is empty so our conversion is
		2	successful.

Question 1 Postfix Expression Evaluation:

Assume that $A = 1, B = 2, C = 3, D = 4, E = 5, F = 6, G = 7, H = 8$	1 2 3 4 * - 5 / 6 + + 7 8 / -	$egin{array}{cccccccccccccccccccccccccccccccccccc$
--	-------------------------------	--

Expression	Stack	Prefix(String)	Explanation
			First character came to
тт	,	11	the prefix, '(' and ')' characters
Н	(	H	added beginning and end of the
			string so '(' added to stack
			Because of '/' is a symbol
			not a letter, we push it to our
/	(/	H	stack and because of priority of
			'(' less than '/', we push it to our stack
G	(/	H G	G is a letter so we append it end of
	V/		the prefix
			Because of '-' is a symbol
_	(-	H G /	not a letter and priority of
	(-	n d /	'-' less than '/', '/' will be added to
			prefix and '-' character pushed to stack
Б	,	H.C. / E	F is a letter so we append it end of
$\mathbf{F}$	(-	HG/F	the prefix
			E is a letter so we append it end of
+	(-+	$\mid$ H G $/$ F E	the prefix
			Character '/'s priority is higher than '('
,	( ) ( /	HC/EE	
/	(-+(/	$\mid$ H G $/$ F E	character which is peek of the stack
/		H.G./P.P.	so we push '/' to the stack
(	(-+(/(	HG/FE	We push '(' character into stack anyway
D	(-+(/(	HG/FED	D is a letter so we append it end of
	(-   () (		the prefix
			Character '*'s priority is higher than '('
*	(-+(/(*	HG/FED	character which is peek of the stack
		,	so we push '*' to the stack
-	7 7 7 7 7 T		C is a letter so we append it end of
С	(-+(/(*	$\mid$ H G $/$ F E D C	the prefix
			Because of '-' is a symbol
			not a letter and priority of
_	(-+(/(-	m H~G~/~F~E~D~C~*	'-' less than '*', '*' will be added to
			prefix and '-' character pushed to stack
			_
В	(-+(/(-	HG/FEDC*B	B is a letter so we append it end of
	( . (/ (	,	the prefix
			Because ')' character came, we pop all
)	(-+(/	H G / F E D C * B -	symbols to end of the prefix till we see
			'(' character
			Because ')' character came, we pop all
)	(-+	H G / F E D C * B -/	symbols to end of the prefix till we see
<b>'</b>		, , , , , , , , , , , , , , , , , , , ,	'(' character.
			Priority of '+' which is peek of the stack
+	(-++	H G / F E D C * B -/	and '+' which is next character's priorities
1	(- 1 1.		are the same so we can push it to stack.
			_
A	(-++	H G / F E D C * B -/ A	A is a letter so we append it end of
	,	, ,	the prefix
			Because ')' character came, we pop all
			symbols to end of the prefix till we see
)		H G / F E D C * B - / A + + -	'(' character.Our expression is over and
			our stack is empty so our conversion is
			successful.
			At the end, we will reverse this string
		-++A/-B*CDEF/GH	shown as Prefix column
			bliowii ab i iciix coluiliii

## Question 1 Prefix Expression Evaluation:

Assume that $A = 1, B = 2, C = 3, D = 4, E = 5, F = 6, G = 7, H = 8$	-++1/-2*3456/78	$egin{array}{cccccccccccccccccccccccccccccccccccc$
--	-----------------	--

Question 2: ii) ! ( A  $\,$ ! (( B < C ) || ( C > D ))) || ( C < E )

	• //
Assume that	,

		$ \frac{1}{1}  0$			
Eumaggion	Stack	Dogtfor/Ctmin m	Explanation		
Expression	Stack	Postfix(String)	Explanation First character came to		
!	!				
	!		the stack because '!' character is a symbol		
			'(' always added to stack until		
(	!(				
			')' symbol A is a letter so we append it end of		
A	!(	A	the postfix. 'A' is added as first		
А	:(	A	character		
			'&&' symbol's priority is more than		
&&	!(&&	A	'(' symbol so it will be added to stack		
	_		'!' symbol's priority is more than		
!	!(&&!	A	'&&' symbol so it will be added to stack		
			'(' always added to stack until		
(	!(&&!(	A	')' symbol		
,			'(' always added to stack until		
(	!(&&!((	A	')' symbol		
D	1/0 0 1//	A.D.	B is a letter so we append it end of		
В	!(&&!((	AB	the postfix.		
	1/0 0 1// -	A.D.	'<' symbol's priority is more than		
<	!(&&!((<	AB	'(' symbol so it will be added to stack		
C	1(0-0-1(( <	ABC	C is a letter so we append it end of		
С	!(&&!((<	ABC	the postfix.		
			Because ')' character come, we pop all the		
)	!(&&!	ABC<	characters until we peek '(' character at the		
			top. After than we pop '(' character at the top		
			'  ' symbol's priority is more than		
	!(&&!	ABC<	'!' which is top of the stack		
			symbol so it will be added to stack		
(	!(&&!  (	ABC<	'(' always added to stack until ')' symbol		
$\mathbf{C}$	!(&&!  (	ABC <c< td=""><td colspan="2">C is a letter so we append it end of</td></c<>	C is a letter so we append it end of		
	111		the postfix.  '>' symbol's priority is more than		
>	!(&&!  (>	ABC <c< td=""><td colspan="2">'(' symbol's priority is more than '(' symbol so it will be added to stack</td></c<>	'(' symbol's priority is more than '(' symbol so it will be added to stack		
D	!(&&!  (>	ABC <cd< td=""><td colspan="2">D is a letter so we append it end of the postfix.</td></cd<>	D is a letter so we append it end of the postfix.		
			Because ')' character come, we pop all the		
\	!(&&!	ABC <cd></cd>	·		
)	:(&&:	ABC <cd></cd>	characters until we peek '(' character at the top. After than we pop '(' character at the top		
			Because ')' character come, we pop all the		
)	!	ABC <cd>  !&amp;&amp;</cd>	characters until we peek '(' character at the		
,	•	1150 (05)	top. After than we pop '(' character at the top		
			Because ')' character come, we pop all the		
)		ABC <cd>  !&amp;&amp;!</cd>	characters until we peek '(' character at the		
,			top. After than we pop '(' character at the top		
11		There is no element at the stack so '  ' char-			
		ABC <cd>  !&amp;&amp;!</cd>	will be pushed to stack		
(		ABC <cd>  !&amp;&amp;!</cd>	'(' always added to stack until ')' symbol		
C			C is a letter so we append it end of		
С		ABC <cd>  !&amp;&amp;!C</cd>	the postfix.		
			'<' symbol's priority is more than		
<	(<	ABC <cd>  !&amp;&amp;!C</cd>	'(' which is top of the stack		
			symbol so it will be added to stack		
E	11/2	ABC <cd>  !&amp;&amp;!CE</cd>	E is a letter so we append it end of		
12	(<	ADO COD >    :&&:OE	the postfix.		

Question 2 Postfix Expression Evaluation:

Assume that $A = 1$ , $B = 0$ , $C = 1$ , $D = 0$ , $E = 1$	101<10>  !&&!11<	101<10>  !&&!11<     1110>  !&&!11<     111  !&&!11<     11!&&!11<     10&&!11<     0!11<     111<     10     1
---	------------------	---

Expression	Stack	Prefix(String)	Explanation	
)	)		')' always added to stack until '(' symbol	
E	)	E	E is a letter so we append it end of the prefix.	
	)<		'<' symbol's priority is more than ')' symbol	
<	) <		so it will be added to stack	
С	)<	EC	C is a letter so we append it end of the prefix	
			Because '(' character come, we pop all the	
(		EC<	characters until we peek ')' character at the	
			top. After than we pop ')' character at the top	
		$\mid$ EC $<$	First character came to the stack because '  '	
11			character is a symbol	
)		EC<	')' always added to stack until '(' symbol	
)	))	EC<	')' always added to stack until '(' symbol	
)	)))	EC<	')' always added to stack until '(' symbol	
D	)))	EC <d< td=""><td>D is a letter so we append it end of</td></d<>	D is a letter so we append it end of	
	11777	20 (2	the prefix.	
>	)))>	EC <d< td=""><td>'&gt;' symbol's priority is more than ')'</td></d<>	'>' symbol's priority is more than ')'	
			symbol so it will be added to stack	
С	)))>	EC <dc< td=""><td>C is a letter so we append it end of the prefix.</td></dc<>	C is a letter so we append it end of the prefix.	
			Because ')' character came, we pop all	
(	))	EC <dc></dc>	symbols to end of the prefix till we see	
	1177		'(' character then append them end of the	
			prefix string.	
	))	EC <dc></dc>	'  ' symbol's priority is more than	
11			')' symbol so it will be added to stack	
)	))  )	EC <dc></dc>	')' always added to stack until '(' symbol	
С	))  )	EC <dc>C</dc>	C is a letter so we append it end of the prefix.	
<	))  )<	EC <dc>C</dc>	'<' symbol's priority is more than ')' symbol so it will be added to stack	
В		EC <dc>CB</dc>	, ,	
D	))  )<	EC <dc>CB</dc>	B is a letter so we append it end of the prefix.  Because ')' character came, we pop all	
			symbols to end of the prefix till we see	
(	))	EC <dc>CB&lt;</dc>	'(' character then append them end of the	
			prefix string.	
			Because ')' character came, we pop all	
	115		symbols to end of the prefix till we see	
(	)	EC <dc>CB&lt;  </dc>	'(' character then append them end of the	
			prefix string.	
,	1171	EC (DC) CD (III	'!' symbol's priority is more than	
!	)!	EC < DC > CB <   !	')' symbol so it will be added to stack	
			'&&' symbol's priority is less than	
&&	)!&&	EC <dc>CB&lt;  !</dc>	'!' symbol so '!' will be appended to	
			prefix string, '&&' will be pushed to stack	
A	)!&&	EC <dc>CB&lt;  !A</dc>	A is a letter so we append it end of the prefix.	
			Because ')' character came, we pop all	
		EC <dc>CB&lt;  !A&amp;&amp;!</dc>	symbols to end of the prefix till we see	
	''	20 (20) (1) (1) (1000)	'(' character then append them end of the	
			prefix string.	
!	!	EC <dc>CB&lt;  !A&amp;&amp;!</dc>	'!' symbol's priority is more than	
	11		'  ' symbol so it will be added to stack	
		EC (DC) CD (HARRING	We pop all symbols at the stack and append	
		EC <dc>CB&lt;  !A&amp;&amp;!  </dc>	all the symbols that are poped from stack we	
			append to postfix	
		!&&A!   <bc>DC<ce< td=""><td>At the end, we will reverse this string shown as Prefix column</td></ce<></bc>	At the end, we will reverse this string shown as Prefix column	
			as i ienx commin	

Question 2 Prefix Expression Evaluation:

Assume that $A=1,B=0,C=1,D=0,E=1$	!&&1!  <01>01<11	!&&1!  <01>01<11   !&&1!  <01>010   !&&1!  <0100   !&&1!  100   !&&1!0   !&&100   !00   10
-----------------------------------	------------------	---