SHERAZ TARIQ 21i-1188 CS-A

1- $(A*(B/C-D)\hat{}E)+(E/F/(G+H))$

Reserve Polish:

$(A*((/BC)-D)\hat{}E)+(E/F/(+GH))$

$(A*(-/BCD)\hat{}E)+((/EF)/(+GH)$

$(A*(\hat{}-/BCDE))+(//EF)/(+GH)$

$(*A\hat{}-/BCDE)+(//EF+GH)$

$+*A\hat{}-/BCDE//EF+GH.$

Reverse Polish:

$(A*((BC/)-D)\hat{}E)+(E/F/(GH+))$

$(A*(BC/D-)\hat{}E)+((EF/)/(GH+))$

$(A*(BC/D-E\hat{}))+(EF/GH+/)$

$(ABC/D-E\hat{}*)+(EF/GH+/)$

$ABC/D-E\hat{}*EF/GH+/+$

2- $A+B\&(((C+D\%E)*F)/G)$

$A+B\&(((C+(\%DE))*F)/G)$

$A+B\&(((+C\%DE)*F)/G)$

$A+B\&((*+C\%DEF)/G)$

$A+B\&(/*+C\%DEFG)$

$(+AB)\&(/*+C\%DEFG)$

$\&+AB/*+C\%DEFG$

Reverse Polish

$A+B\&(((C+(DE\%))*F)/G)$

$A+B\&(((CDE\%+)*F)/G)$

$A+B\&((CDE\%+F*)/G)$

$A+B\&(CDE\%+F*G/)$

$(AB+)\&(CDE\%+F*G/)$

$AB+CDE\%+F*G\&$

Part B

(i) = ABCD % E + F * G / * +

The infix notation of the above postfix notation is,

= (A + (B * ((((C % D) + E) * F) / G)))

(ii) = + A + * BC / ^ DE * FG

The infix notation of the above prefix notation is,

= (A + ((B * C) + ((D ^ E) / (F * G))))

Part A:

(ii) A + B & (((C + D % E) * F) / G) ——— ①

In order to convert the above expression into reverse polish notation (RPN), we do the following.

= A + B & (((C + DE) %) * F) / G)

= A + B & (((CDE % +) * F) / G)

= A + B & ((CDE % + F *) / G)

= A + B & CDE % + F * G /

= AB & CDE % + F * G / +

Part C

A+B*C^D^E%F*G

Symb	Postfix string	Opstk	statement
A	A		add symb to the postfix string
+	A	+	push (opstk, symb)
B	AB	+	add symb to the postfix string
*	AB	+*	push (opstk, symb)
C	ABC	+*^	add symb to the postfix string
^	ABC	+*^	push (opstk, symb)
D	ABCD	//	add symb to the postfix string
^	ABCD	+*^^	push (opstk, symb)
E	ABCDE	//	add symb to the postfix string
%	ABCDE^	+*^	top symb = pop (opstk)
	ABCDE^	//	add top symb to the postfix string
	ABCDE^	+*	top symb = pop (opstk)
	ABCDE^^	+*	add top symb to the postfix string
	ABCDE^^	+	
	ABCDE^^*	+	add topsymb to the postfix string
F	ABCDE^^*	+%	push (opstk, symb)
	ABCDE^^*F	+%	add symb to postfix string
	ABCDE^^*F	+	top symb = pop (opstk)
	ABCDE^^*F%	+	add top symb to postfix string
	ABCDE^^*F%	+*	push (opstk, symb)
G	ABCDE^^*F%G	+*	add symb to postfix string
	//	+	top symb = pop (opstk)
	ABCDE^^*F%G	+	add topsymb to postfix string
	//		top symb = pop (opstk)
	ABCDE^^*F%G		add top symb to postfix string

Part(d)

A+B*C^(((D^E)%F)*G)

Symb	Postfix string	Opstk	Statement.
A	A		
+	A	+	add Symb to postfix string
B	AB	+	push (opstk, symb).
*	AB	+*	add symb to postfix string.
C	ABC	+*	push (opstk, symb)
^	ABC	+*^	add symb to postfix string.
(	ABC	+*^(	push (opstk, symb)
(	ABC	+*^((	push (opstk, symb)
(	ABC	+*^(((	push (opstk, symb)
D	ABCD	+*^(((	add symb to postfix string.
^	ABCD	+*((^	push (opstk, symb)
E	ABCDE	+*^(((^	add symb to postfix string
)	ABCDE	+*^(((^	top symb = pop (opstk)
	ABCDE^	+*^((	pop (opstk)
%	ABCDE^	+*^((%	~~add~~ sy. push (opstk, symb)
)	ABCDE^	+*^((%	top symb = pop (opstk)
	ABCDE^%	+*^(	pop (opstk)
*	ABCDE^%	+*^(*	top symb = pop (opstk
)	ABCDE^%	+*^(*	top symp.
	ABCDE^%*	+*^	pop (opstk)
	ABCDE^%*	+*	top symb = pop (opstk)
	ABCDE^%**	+	add top symb to postfix string
	ABCDE^%*^*		add top symb to postfix string
	ABCDE^%*^**		top symb = pop (opstk)
	ABCDE^%*^**+		add top symb to postfix string.