Q1

1) **Converting A + ((B – C * D) / E) + F – G / H from infix to prefix**

first we reverse the expression H / G – F +) E /) D * C – B ((+ A

then we begin scanning the expression to convert it.

Operands/operators	**Stack (OPERATORS)**	**Expression**
H	EMPTY	H
/	(STACK IS EMPTY -> PUSH OPERATOR) /	H
G	/	H G
-	(/ has higher precedence than - so we pop it then we retest for the next operator in the stack) Empty	H G /
- (retest)	(STACK IS EMPTY -> PUSH OPERATOR) -	H G /
F	-	H G / F
+	(SAME PRECEDENCE -> PUSH) - +	H G / F
)	(PUSH THE CLOSING PRENTHESIS) - +)	H G / F
E	- +)	H G / F E
/	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH) - +) /	H G / F E
)	(PUSH THE CLOSING PRENTHESIS) - +) /)	H G / F E
D	- +) /)	H G / F E D

*****	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH) - +) /) *	H G / F E D
C	- +) /) *	H G / F E D C
-	(* has higher precedence than - so we pop it then we retest for the next operator in the stack) - +) /)	H G / F E D C *
- (retest)	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH) - +) /) -	H G / F E D C *
B	- +) /) -	H G / F E D C * B
(	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) - +) /)	H G / F E D C * B -
((retest)	(REACHED A CLOSING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) - +) /	H G / F E D C * B -
(	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) - +)	H G / F E D C * B - /
((retest)	(REACHED A CLOSING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) - +	H G / F E D C * B - /
+	(SAME PRECEDENCE -> PUSH) - + +	H G / F E D C * B - /
A	- + +	H G / F E D C * B - / A
END OF THE ESPRESSION	(POP OPERATORS UNTIL THE STACK IS EMPTY)	H G / F E D C * B - / A + + -

Lastly, we reverse the expression we obtained for the final result of:

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

Evaluate A = 3, B = 4, C = 2, D = 5, E = 6, F = 8, G = 7, H = 1 :

Expression	**Stack**	**Evaluation**
- + + 3 / - 4 * 2 5 6 8 / 7 1	EMPTY	NONE

- + + 3 / - 4 * 2 5 6 8 / 7	(PUSH 1) 1	NONE
- + + 3 / - 4 * 2 5 6 8 /	(PUSH 7) 1 7	NONE
- + + 3 / - 4 * 2 5 6 8	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 7 AND 1)	7 / 1 = 7
- + + 3 / - 4 * 2 5 6 8	(PUSH EVALUATION RESULT) 7	NONE
- + + 3 / - 4 * 2 5 6	(PUSH 8) 7 8	NONE
- + + 3 / - 4 * 2 5	(PUSH 6) 7 8 6	NONE
- + + 3 / - 4 * 2	(PUSH 5) 7 8 6 5	NONE
- + + 3 / - 4 *	(PUSH 2) 7 8 6 5 2	NONE
- + + 3 / - 4	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 2 AND 5) 7 8 6	2 * 5 = 10
- + + 3 / - 4	(PUSH EVALUATION RESULT) 7 8 6 10	NONE
- + + 3 / -	(PUSH 4) 7 8 6 10 4	NONE
- + + 3 /	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 4 AND 10) 7 8 6	4 – 10 = -6
- + + 3 /	(PUSH EVALUATION RESULT) 7 8 6 -6	NONE
- + + 3	(OPERATOR ENCOUNTERD POP 2 ELEMENTS -6 AND 6) 7 8	-6 / 6 = -1
- + + 3	(PUSH EVALUATION RESULT) 7 8 -1	NONE
- + +	(PUSH 3) 7 8 -1 3	NONE
- +	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 3 AND -1) 7 8	3 + -1 = 2
- +	(PUSH EVALUATION RESULT) 7 8 2	NONE
-	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 2 AND 8) 7	2 + 8 = 10

-	(PUSH EVALUATION RESULT) 7 10	NONE
EMPTY	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 10 AND 7) EMPTY	10 – 7 = 3
EMPTY	(PUSH EVALUATION RESULT) 3	NONE

RESULT IS THE TOP OF THE STACK = 3

2) Converting A + ((B – C * D) / E) + F – G / H from infix to postfix

We begin scanning the expression to convert it.

Operands/operators	Stack (OPERATORS)	Expression
A	EMPTY	A
+	(STACK IS EMPTY -> PUSH OPERATOR) +	A
(	(PUSH THE OPENNING PRENTHESIS) + (	A
(	(PUSH THE OPENNING PRENTHESIS) + ((	A
B	+ ((	A B
-	(OPERATOR AFTER OPENNING PRENTHESIS -> PUSH) + ((-	A B
C	+ ((-	A B C
*****	(* HAS HIGHER PRECEDENCE THAN - -> PUSH) + ((- *	A B C
D	+ ((- *	A B C D

)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) + ((-	A B C D *
) (RETEST)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) + ((	A B C D * -
) (RETEST)	(REACHED AN OPENNING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) + (	A B C D * -
/	(OPERATOR AFTER OPENNING PRENTHESIS -> PUSH) + (/	A B C D * -
E	+ (/	A B C D * - E
)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) + (	A B C D * - E /
) (RETEST)	(REACHED AN OPENNING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) +	A B C D * - E /
+	(SAME PRECEDENCE OPERATORS->POP TOP -> PUSH) +	A B C D * - E / +
F	+	A B C D * - E / + F
-	(SAME PRECEDENCE OPERATORS->POP TOP -> PUSH) -	A B C D * - E / + F +
G	-	A B C D * - E / + F + G
/	(/ HAS HIGHER PRECEDENCE THAN - -> PUSH) - /	A B C D * - E / + F + G
H	- /	A B C D * - E / + F + G H
END OF THE ESPRESSION	(POP OPERATORS UNTIL THE STACK IS EMPTY)	A B C D * - E / + F + G H / -

So, our result of the postfix expression is:
A B C D * - E / + F + G H / -

Evaluate A = 3, B = 4, C = 2, D = 5, E = 6, F = 8, G = 7, H = 1 :

Expression	Stack	Evaluation
3 4 2 5 * - 6 / + 8 + 7 1 / -	EMPTY	NONE
4 2 5 * - 6 / + 8 + 7 1 / -	(PUSH 3) 3	NONE
2 5 * - 6 / + 8 + 7 1 / -	(PUSH 4) 3 4	NONE
5 * - 6 / + 8 + 7 1 / -	(PUSH 2) 3 4 2	NONE
*** - 6 / + 8 + 7 1 / -**	(PUSH 5) 3 4 2 5	NONE
- 6 / + 8 + 7 1 / -	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 5 AND 2) 3 4	5 * 2 = 10
- 6 / + 8 + 7 1 / -	(PUSH EVALUATION RESULT) 3 4 10	NONE
6 / + 8 + 7 1 / -	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 10 AND 4) 3	4 – 10 = -6
6 / + 8 + 7 1 / -	(PUSH EVALUATION RESULT) 3 -6	NONE
/ + 8 + 7 1 / -	(PUSH 6) 3 -6 6	NONE
+ 8 + 7 1 / -	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 6 AND -6) 3	-6 / 6 = -1
+ 8 + 7 1 / -	(PUSH EVALUATION RESULT) 3 -1	NONE
8 + 7 1 / -	(OPERATOR ENCOUNTERD POP 2 ELEMENTS -1 AND 3) EMPTY	3 + -1 = 2
8 + 7 1 / -	(PUSH EVALUATION RESULT) 2	NONE
+ 7 1 / -	(PUSH 8) 2 8	NONE

7 1 / -	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 8 AND 2) EMPTY	2 + 8 = 10
7 1 / -	(PUSH EVALUATION RESULT) 10	NONE
1 / -	(PUSH 7) 10 7	NONE
/ -	(PUSH 1) 10 7 1	NONE
-	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 1 AND 7) 10	7 / 1 = 7
-	(PUSH EVALUATION RESULT) 10 7	NONE
EMPTY	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 7 AND 10) EMPTY	10 – 7 = 3
EMPTY	(PUSH EVALUATION RESULT) 3	NONE

RESULT IS THE TOP OF THE STACK = 3

3) **Converting ! (A && ! ((B < C) | | (C > D))) | | (C < E) from infix to prefix**

first we reverse the expression **) E < C (| |))) D > C (| |) C < B ((! && A (!**

then we begin scanning the expression to convert it.

Operands/operators	Stack (OPERATORS)	Expression
)	(PUSH THE CLOSING PRENTHESIS))	EMPTY

E	)	E
<	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH)) <	E
C	) <	E C
(	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS))	E C <
((retest)	(REACHED A CLOSING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) EMPTY	E C <
||	(STACK IS EMPTY -> PUSH OPERATOR) ||	E C <
)	(PUSH THE CLOSING PRENTHESIS) ||)	E C <
)	(PUSH THE CLOSING PRENTHESIS) ||))	E C <
)	(PUSH THE CLOSING PRENTHESIS) ||)))	E C <
D	||)))	E C < D
>	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH) ||))) >	E C < D
C	||))) >	E C < D C
(	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) ||)))	E C < D C >
((retest)	(REACHED A CLOSING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) ||))	E C < D C >

||	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH) | |)) | |	E C < D C >
)	(PUSH THE CLOSING PRENTHESIS) | |)) | |)	E C < D C >
C	| |)) | |)	E C < D C > C
<	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH) | |)) | |) <	E C < D C > C
B	| |)) | |) <	E C < D C > C B
(	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) | |)) | |)	E C < D C > C B <
((retest)	(REACHED A CLOSING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) | |)) | |	E C < D C > C B <
(	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) | |))	E C < D C > C B < | |
((retest)	(REACHED A CLOSING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) | |)	E C < D C > C B < | |
!	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH) | |) !	E C < D C > C B < | |
&&	(! has higher precedence than && so we pop it then we retest for the next operator in the stack) | |)	E C < D C > C B < | | !
&& (RETEST)	(OPERATOR AFTER CLOSING PRENTHESIS -> PUSH) | |) &&	E C < D C > C B < | | !
A	| |) &&	E C < D C > C B < | | ! A

(	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) ||)	E C < D C > C B < || ! A &&
((retest)	(REACHED A CLOSING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) ||	E C < D C > C B < || ! A &&
!	(! has higher precedence than || -> PUSH) || !	E C < D C > C B < || ! A &&
END OF THE ESPRESSION	(POP OPERATORS UNTIL THE STACK IS EMPTY)	E C < D C > C B < || ! A && ! ||

Lastly, we reverse the expression we obtained for the final result of:

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

Evaluate A = 7, B = 5, C = 9, D = 4, E = 3 :

T = TRUE, F = FALSE

Expression	Stack	Evaluation
|| ! && 7 ! || < 5 9 > 9 4 < 3 9	EMPTY	NONE
|| ! && 7 ! || < 5 9 > 9 4 < 3	(PUSH 9) 9	NONE
|| ! && 7 ! || < 5 9 > 9 4 <	(PUSH 3) 9 3	NONE
|| ! && 7 ! || < 5 9 > 9 4	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 3 AND 9) EMPTY	3 < 9 = T
|| ! && 7 ! || < 5 9 > 9 4	(PUSH EVALUATION RESULT) T	NONE
|| ! && 7 ! || < 5 9 > 9	(PUSH 4) T 4	NONE
|| ! && 7 ! || < 5 9 >	(PUSH 9) T 4 9	NONE
|| ! && 7 ! || < 5 9	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 9 AND 4) T	9 > 4 = T

| | ! && 7 ! | | < 5 9	(PUSH EVALUATION RESULT) T T	NONE
| | ! && 7 ! | | < 5	(PUSH 9) T T 9	NONE
| | ! && 7 ! | | <	(PUSH 5) T T 9 5	NONE
| | ! && 7 ! | |	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 5 AND 9) T T	5 < 9 = T
| | ! && 7 ! | |	(PUSH EVALUATION RESULT) T T T	NONE
| | ! && 7 !	(OPERATOR ENCOUNTERD POP 2 ELEMENTS T AND T) T	T | | T = T
| | ! && 7 !	(PUSH EVALUATION RESULT) T T	NONE
| | ! && 7	(UNARY OPERATOR ENCOUNTERD POP 1 ELEMENT1 T) T	!T = F
| | ! && 7	(PUSH EVALUATION RESULT) T F	NONE
| | ! &&	(PUSH 7) T F 7	NONE
| | !	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 7 AND F) T	7 && F = F
| | !	(PUSH EVALUATION RESULT) T F	NONE
| |	(UNARY OPERATOR ENCOUNTERD POP 1 ELEMENT1 F) T	!F = T
| |	(PUSH EVALUATION RESULT) T T	NONE
EMPTY	(OPERATOR ENCOUNTERD POP 2 ELEMENTS T AND T) EMPTY	T | | T = T
EMPTY	(PUSH EVALUATION RESULT) T	NONE

THE RESULT IS THE TOP OF THE STACK = T

4) Converting ! (A && ! ((B < C) | | (C > D))) | | (C < E) from infix to postfix

We begin scanning the expression to convert it.

Operands/operators	Stack (OPERATORS)	Expression
!	(STACK IS EMPTY -> PUSH OPERATOR) !	EMPTY
(	(PUSH THE OPENNING PRENTHESIS) ! (	EMPTY
A	! (	A
&&	(OPERATOR AFTER OPENNING PRENTHESIS -> PUSH) ! (&&	A
!	(! HAS HIGHER PRECEDENCE THAN && -> PUSH) ! (&& !	A
(	(PUSH THE OPENNING PRENTHESIS) ! (&& ! (	A
(	(PUSH THE OPENNING PRENTHESIS) ! (&& ! ((	A
B	! (&& ! ((	A B
<	(OPERATOR AFTER OPENNING PRENTHESIS -> PUSH) ! (&& ! ((<	A B
C	! (&& ! ((<	A B C

)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) ! (&& ! ((	A B C <
) (RETEST)	(REACHED AN OPENNING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) ! (&& ! (	A B C <
||	(OPERATOR AFTER OPENNING PRENTHESIS -> PUSH) ! (&& ! (||	A B C <
(	(PUSH THE OPENNING PRENTHESIS) ! (&& ! (|| (	A B C <
C	! (&& ! (|| (	A B C < C
>	(OPERATOR AFTER OPENNING PRENTHESIS -> PUSH) ! (&& ! (|| (>	A B C < C
D	! (&& ! (|| (>	A B C < C D
)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) ! (&& ! (|| (	A B C < C D >
) (RETEST)	(REACHED AN OPENNING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) ! (&& ! (||	A B C < C D >
)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) ! (&& ! (	A B C < C D > ||
) (RETEST)	(REACHED AN OPENNING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) ! (&& !	A B C < C D > ||
)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) ! (&&	A B C < C D > || !

) (RETEST)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) ! (	A B C < C D > | | ! &&
) (RETEST)	(REACHED AN OPENNING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) !	A B C < C D > | | ! &&
| |	(! HAS HIGHER PRECEDENCE THAN | | -> POP ! AND PUSH | |) | |	A B C < C D > | | ! && !
(	(PUSH THE OPENNING PRENTHESIS) | | (	A B C < C D > | | ! && !
C	| | (	A B C < C D > | | ! && ! C
<	(OPERATOR AFTER OPENNING PRENTHESIS -> PUSH) | | (<	A B C < C D > | | ! && ! C
E	| | (<	A B C < C D > | | ! && ! C E
)	(POP OPERATORS TILL WE REACH A CLOSING PRENTHESIS) | | (	A B C < C D > | | ! && ! C E <
) (RETEST)	(REACHED AN OPENNING PRENTHESIS -> ELEMINATE BOTH PRENTHESIS) | |	A B C < C D > | | ! && ! C E <
END OF THE ESPRESSION	(POP OPERATORS UNTIL THE STACK IS EMPTY)	A B C < C D > | | ! && ! C E < | |

So, our result of the postfix expression is:

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

Evaluate A = 7, B = 5, C = 9, D = 4, E = 3 :

T = TRUE, F = FALSE

Expression	Stack	Evaluation
7 5 9 < 9 4 > || ! && ! 9 3 < ||	EMPTY	NONE
5 9 < 9 4 > || ! && ! 9 3 < ||	(PUSH 7) 7	NONE
9 < 9 4 > || ! && ! 9 3 < ||	(PUSH 5) 7 5	NONE
< 9 4 > || ! && ! 9 3 < ||	(PUSH 9) 7 5 9	NONE
9 4 > || ! && ! 9 3 < ||	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 9 AND 5) 7	5 < 9 = T
9 4 > || ! && ! 9 3 < ||	(PUSH EVALUATION RESULT) 7 T	NONE
4 > || ! && ! 9 3 < ||	(PUSH 9) 7 T 9	NONE
> || ! && ! 9 3 < ||	(PUSH 4) 7 T 9 4	NONE
|| ! && ! 9 3 < ||	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 4 AND 9) 7 T	9 > 4 = T
|| ! && ! 9 3 < ||	(PUSH EVALUATION RESULT) 7 T T	NONE
! && ! 9 3 < ||	(OPERATOR ENCOUNTERD POP 2 ELEMENTS T AND T) 7	T || T = T
! && ! 9 3 < ||	(PUSH EVALUATION RESULT) 7 T	NONE
&& ! 9 3 < ||	(UNARY OPERATOR ENCOUNTERD POP 1 ELEMENT1 T) 7	!T = F
&& ! 9 3 < ||	(PUSH EVALUATION RESULT) 7 F	NONE
! 9 3 < ||	(OPERATOR ENCOUNTERD POP 2 ELEMENTS F AND 7) T	7 && F = F
! 9 3 < ||	(PUSH EVALUATION RESULT) F	NONE
9 3 < ||	(UNARY OPERATOR ENCOUNTERD POP 1 ELEMENT1 F) EMPTY	!F = T

9 3 < ||	(PUSH EVALUATION RESULT) T	NONE
3 < ||	(PUSH 9) T 9	NONE
< ||	(PUSH 3) T 9 3	NONE
||	(OPERATOR ENCOUNTERD POP 2 ELEMENTS 3 AND 9) T	9 < 3 = F
||	(PUSH EVALUATION RESULT) T F	NONE
EMPTY	(OPERATOR ENCOUNTERD POP 2 ELEMENTS F AND T) EMPTY	T || F = T
EMPTY	(PUSH EVALUATION RESULT) T	NONE

THE RESULT IS THE TOP OF THE STACK = T