① Infix to Postfix :

Infix expression : A*(B*C+D*E)+F:

S.NO	Current TOken	operator stack	Postfix string
1.	A	-	A
2.	*	*	A
3.	(	*(	A
4.	B	*(	AB
5.	*	*(*	AB
6.	C	*(*	AB*C
7.	+	*(*+	ABC*
8.	D	*(+	ABC*D
9.	*	*(+*	ABC*D
10.	E	*(* *	ABC*DE
11.	)	*	ABC*DE*+
12.	+	+	ABC*DE*+*
13.	F	+	ABC*DE*+*F
14.			ABC*DE*+*F+

Postfix Expression is : ABC*DE*+*F+

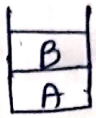
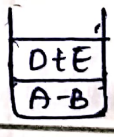
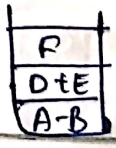
② Infix Expression : A*B^C+D

SNO	current Token	Operator stack	postfix string
1.	A		A
2.	*	*	A
3.	B	*	AB
4.	^	*^	AB
5.	C	*^	ABC
6.	+	+	ABC^*
7.	D	+	ABC^*D
8.		+	ABC^*D+

Postfix Expression : ABC^*D+

③ Postfix to Infix :-

Postfix Expression :- AB-DE+F*/

S.NO	Reading of Postfix	Stack Top	Expression
1.	A	A	
2.	B	B	
3.	-	A-B	
4.	D	D	
5.	E	E	
6.	+	D+E	
7.	F	F	
8.	*	(D+E)*F)	
9.	/	(A-B)/(D+E*F)	

Infix Expression :- (A-B)/(D+E*F)

④ PostFix Conversion : abc*de-/+

S.NO	Symbol	Stack
1.	a	a
2.	b	ab
3.	c	abc
4.	*	a(b*c)
5.	d	a(b*c)d
6.	e	a(b*c)de
7.	-	a(b*c)(d-e)
8.	/	a((b*c)/(d-e))
9.	+	(a+((b*c)/(d-e)))

⑤ Balanced Symbols:

((a+b)*(c-d))

S.NO	Symbol	Stack	Action Taken	Expression so far
1.	(	(	Push '('	(
2.	(	((	Push '('	((
3.	a	((	Append 'a'	((a
4.	+	((	Append '+'	((a+
5.	b	((	Append 'b'	((a+b
6.	)	(	Pop '('	((a+b
7.	*	(*	Push '*'	((a+b)*
8.	(	(*(	Push '('	((a+b)*(
9.	c	(*(	Append 'c'	((a+b)*(c
10.	-	(*(	Append '-'	((a+b)*(c-
11.	d	(*(	Append 'd'	((a+b)*(c-d
12.	)	(*	Pop '('	((a+b)*(c-d)
13.	)		Pop '('	((a+b)*(c-d))

It is valid for "Balauced Symbol"

6. ii) [{(a+b)*(c}-d]

S.NO	Symbol	stack	Action Taken	Expression so far.
1.	(	[(]	Push '('	(
2.	a	[(]	Append 'a'	(a
3.	+	[(,+]	Push '+'	(a+
4.	b	[(,+]	Append 'b'	(a+b
5.	)	[(,+]	Pop '('	(a+b)
6.	*	[(,+,*]	Push '*'	(a+b)*
7.	(	[(,*]	Append '('	(a+b)*(
8.	)	[(]	Pop '('	(a+b)*(
9.	-	[(,-]	Push '-'	(a+b)*(-
10.	d	[(,-]	Append 'd'	(a+b)*(-d
11.	End	ϕ }	Pop remaining operators	(a+b)*(-d)

It is valid for "Balanced symbols"