Tugas Stack

1. Write an algorithm to convert an infix expression to postfix and infix to prefix

POSTFIX EXPRESSION :- A B * C D + – E +

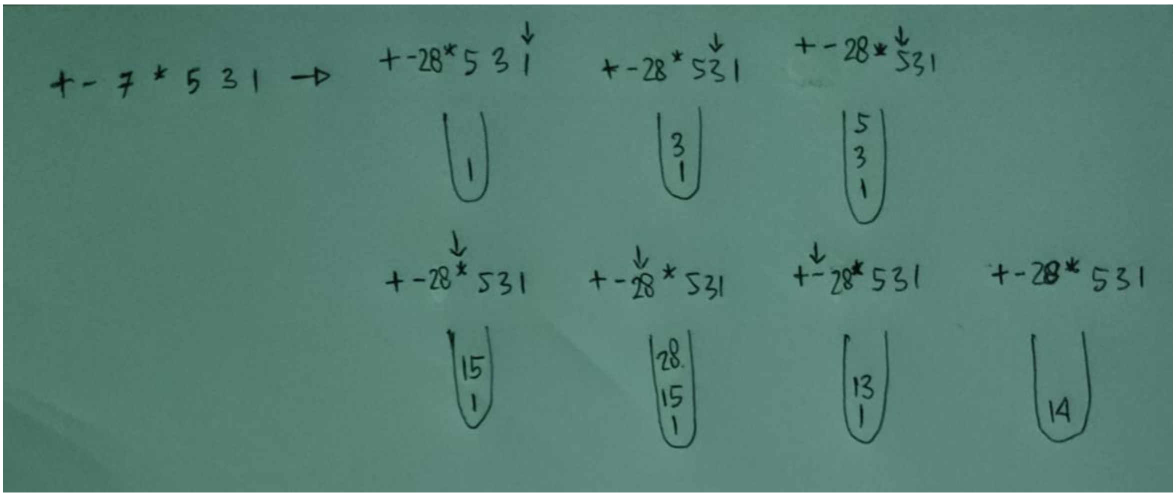
INPUT CHARACTER	**STACK**	**POSTFIX EXPRESSION**
A	Empty	A
*	*	A
B	*	A B
–	–	A B *
(	– (	A B *
C	– (	A B * C
+	– (+	A B * C
D	– (+	A B * C D
)	–	A B * C D +
+	+	A B * C D + –
E	+	A B * C D + – E
End of Input	Empty	A B * C D + – E +

PREFIX EXPRESSION : ++AB-C^-DEF

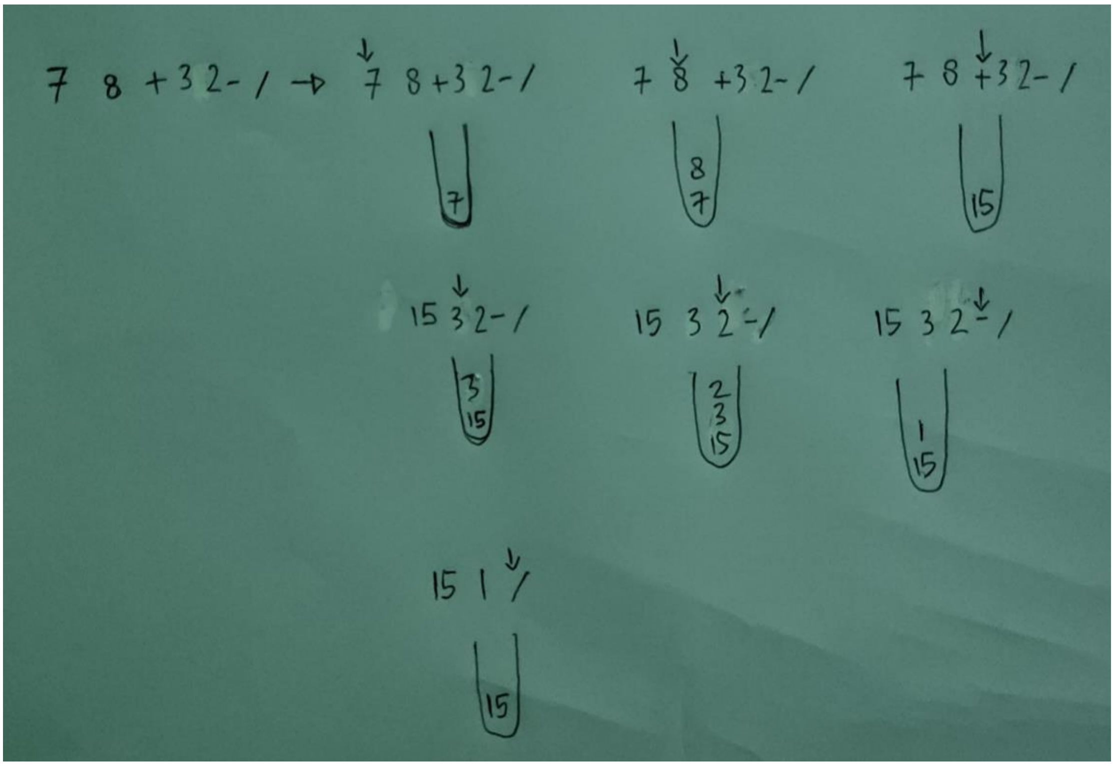
INPUT CHARACTER	**STACK**	**EXPRESSION**
F		F
^	^	F
)	^)	FE
E	^)	FE
–	^)-	FE
D	^)-	FED
(	^)-	FED-
–	–	FED-^
C	–	FED-^C
+	+	FED-^C-
)	+)	FED-^C-
B	+)	FED-^C-B
+	+)+	FED-^C-B
A	+)+	FED-^C-BA
(	+	FED-^C-BA+
		FED-^C-BA++

2. Write an algorithm to evaluate a postfix and a prefix expression

A. Prefix



B. Postfix


7 8 + 3 2 - / →
7 8 + 3 2 - /
7
7 8 + 3 2 - /
8
7
7 8 + 3 2 - /
15
15 3 2 - /
3
15
15 3 2 - /
2
3
15
15 3 2 - /
1
15
15 1 /
15