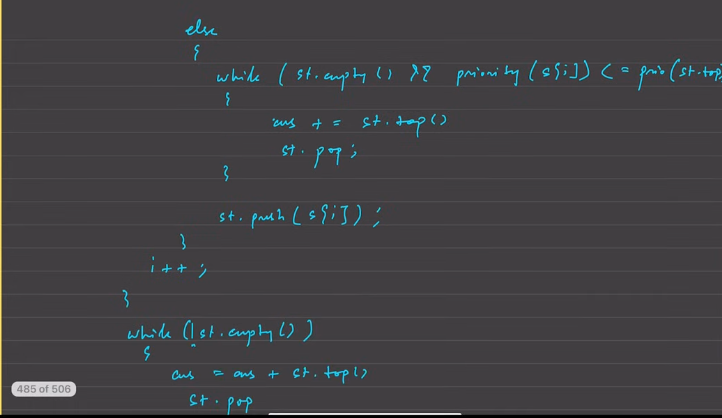
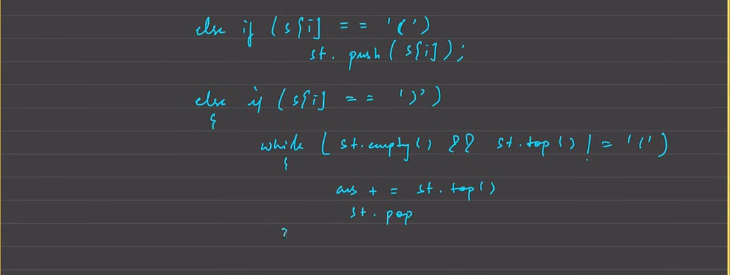
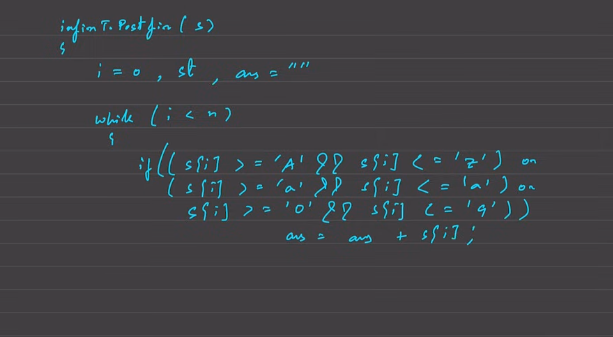
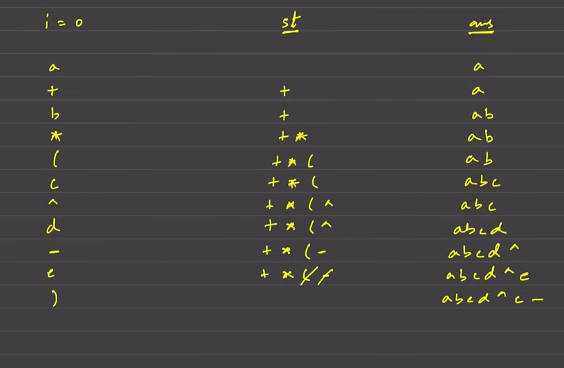
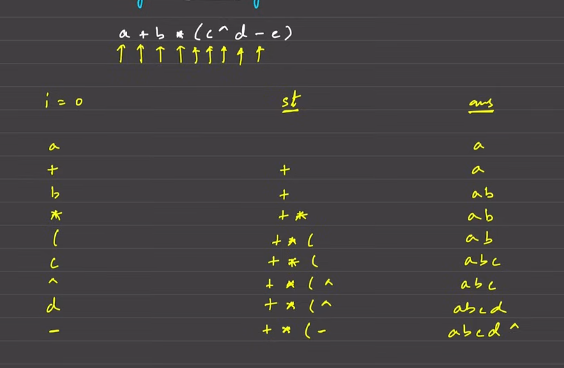
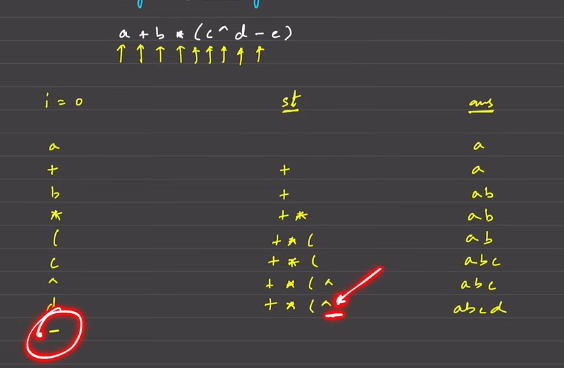
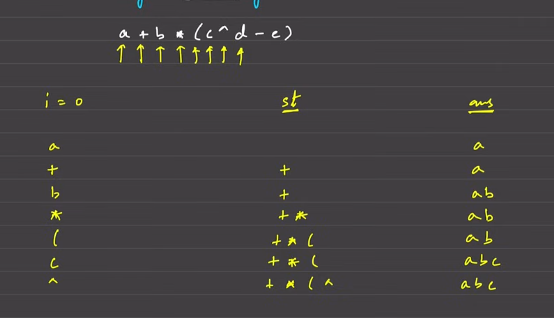
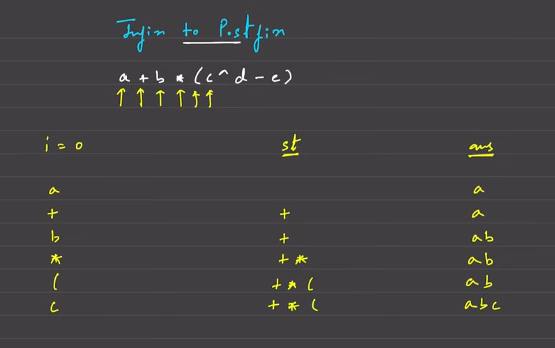
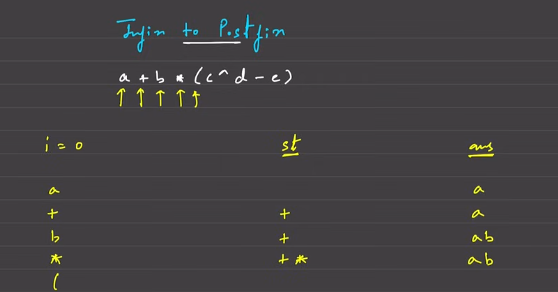
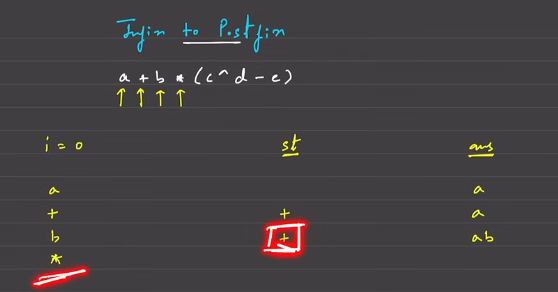
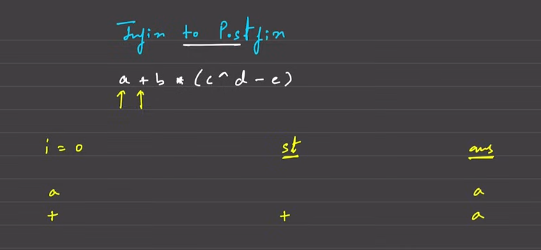
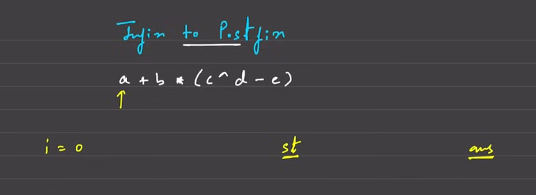


The above mentioned is in order of the infix , prefix and postfix

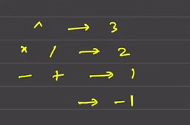
Postfix is used in stac based ocmputers

Prefix- tree ased evaluation

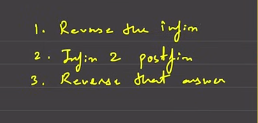
Infix- most used in computer based applications



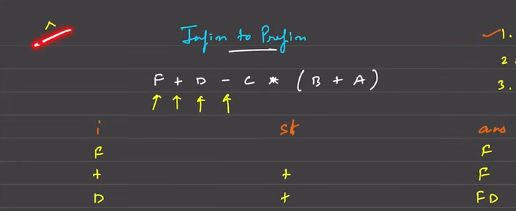
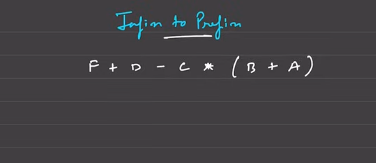
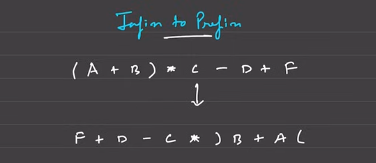
Write priority function



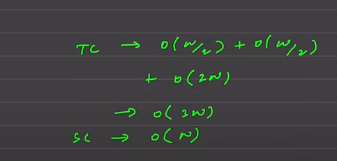
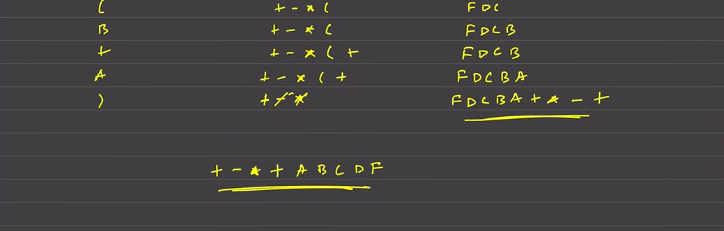
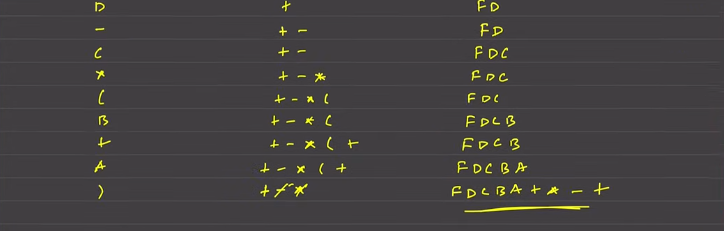
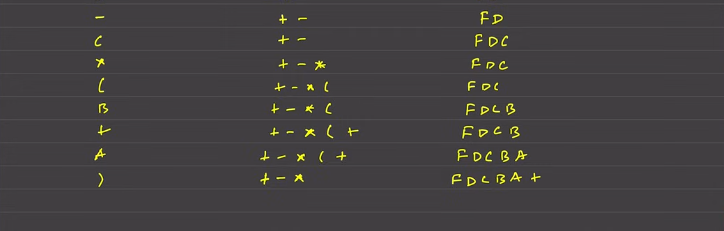
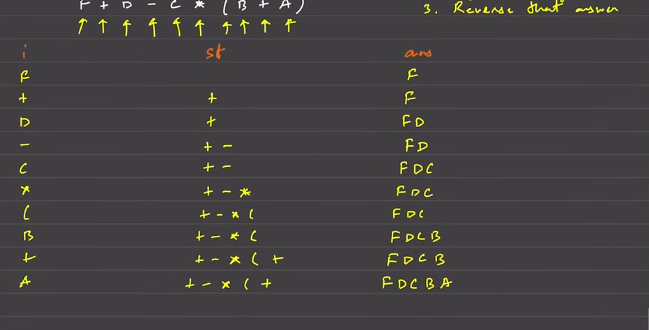
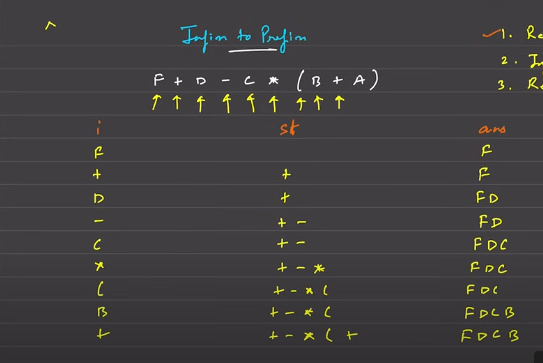
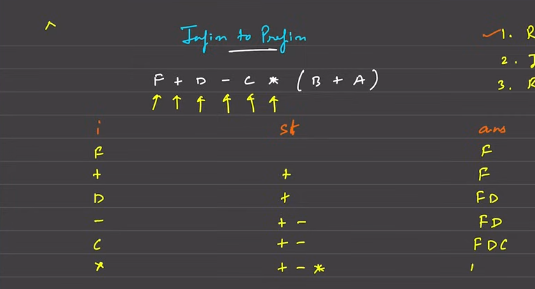
**INFIX to prefix**



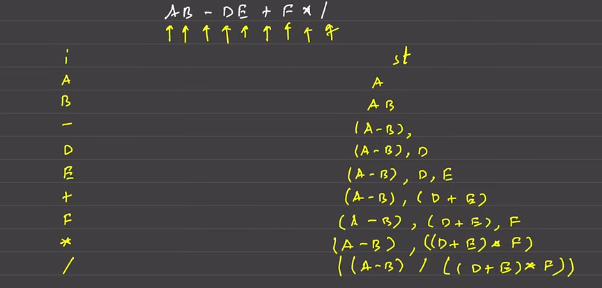
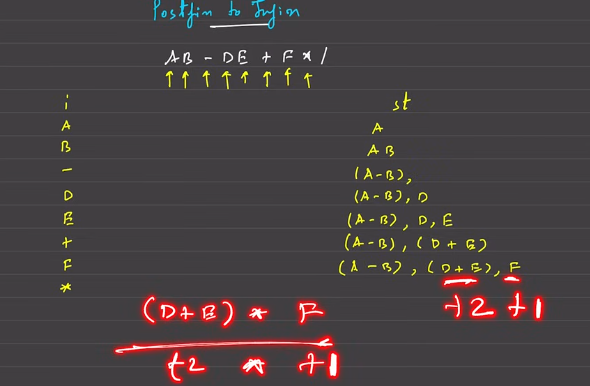
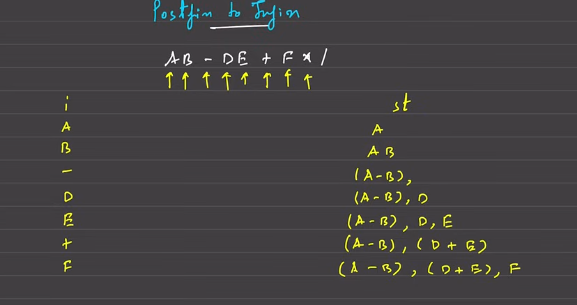
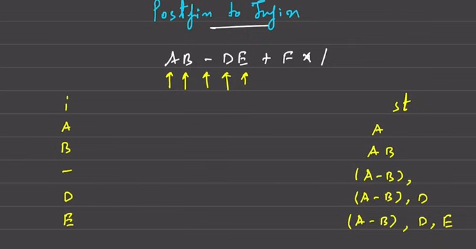
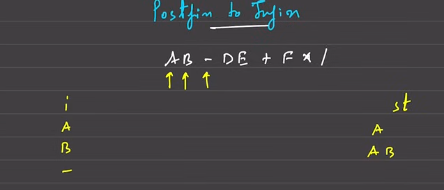
After reversing, we need to change the closing bracket to opening bracket, ad then apply for the infix to prefix again



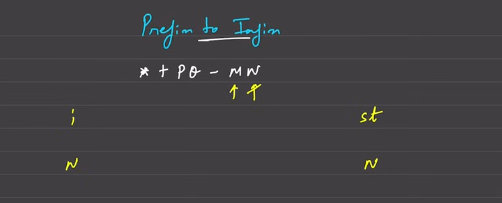
Anything which is not ^ should be puhed inside the stack



**Postfix to infix**



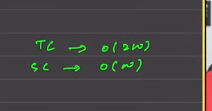
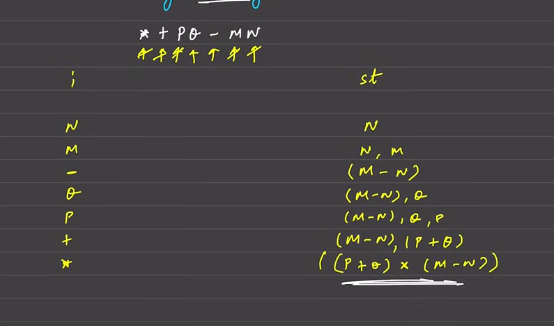
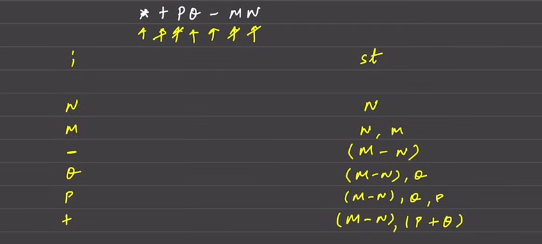
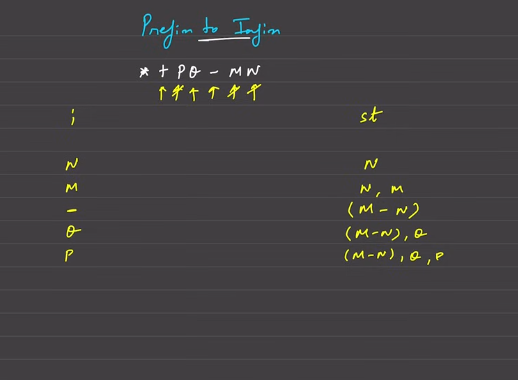
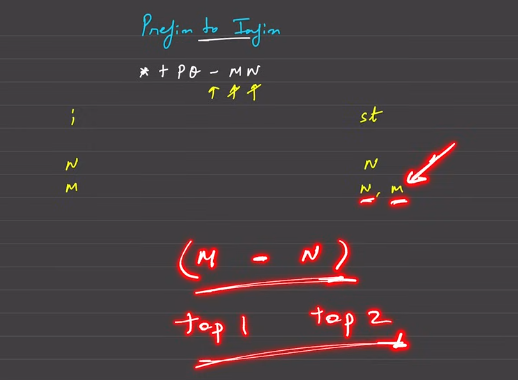
**PREFIX TO INFIX**



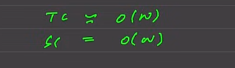
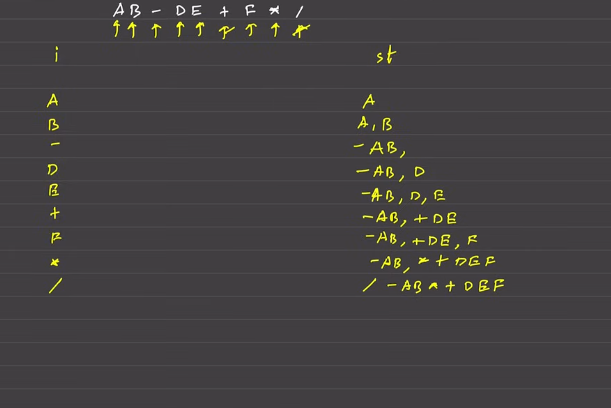
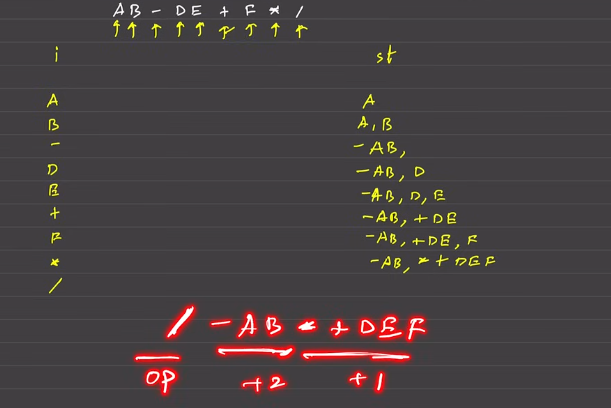
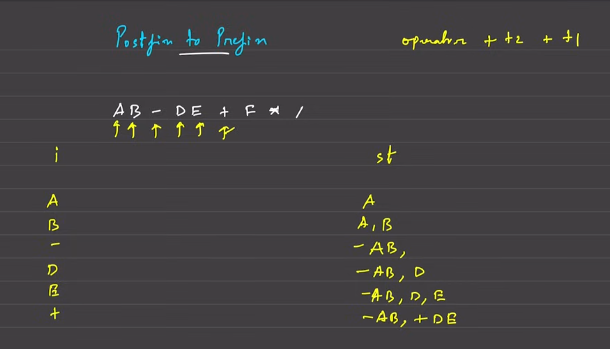
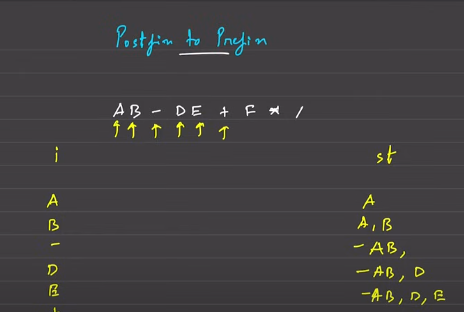
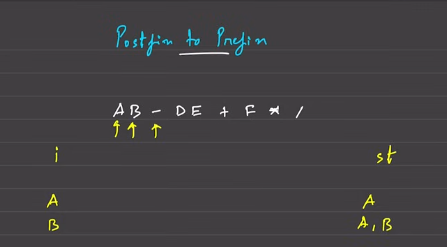
This time we are travelling from behind. Ans when we see the opearnd we push that into the stack and then we mve forward

When we see the operator , the one which in popped as first is considered as the first operand and the one which is popped as second is considered as the second operand.

Then we do the screeing and then poush it agaian into the stack, as the previous method



**Postfix to prefix**



**Prefix to Postfix**

