Infix To Postfix Conversion

Contents:

- What is Postfix expression?
- Different methods to convert infix expression to the postfix expression
- Algorithm to convert infix to postfix expression using stack
- How to convert infix to postfix using stack

What is a Postfix expression?

- An expression is a statement which contains operands and different types of operator.
- For example, **c=a+b** is an expression in which a,b,c are operands and +,= is an operator
- The above expression is nothing but an infix expression b'cause an operator + comes in between operands a and b. Also, = comes in between c and a
- Now, a postfix expression is that in which operators comes after operands
- For example, infix expression **a+b** can be written as a postfix expression **ab+**

Different methods to convert infix expression to postfix expression:

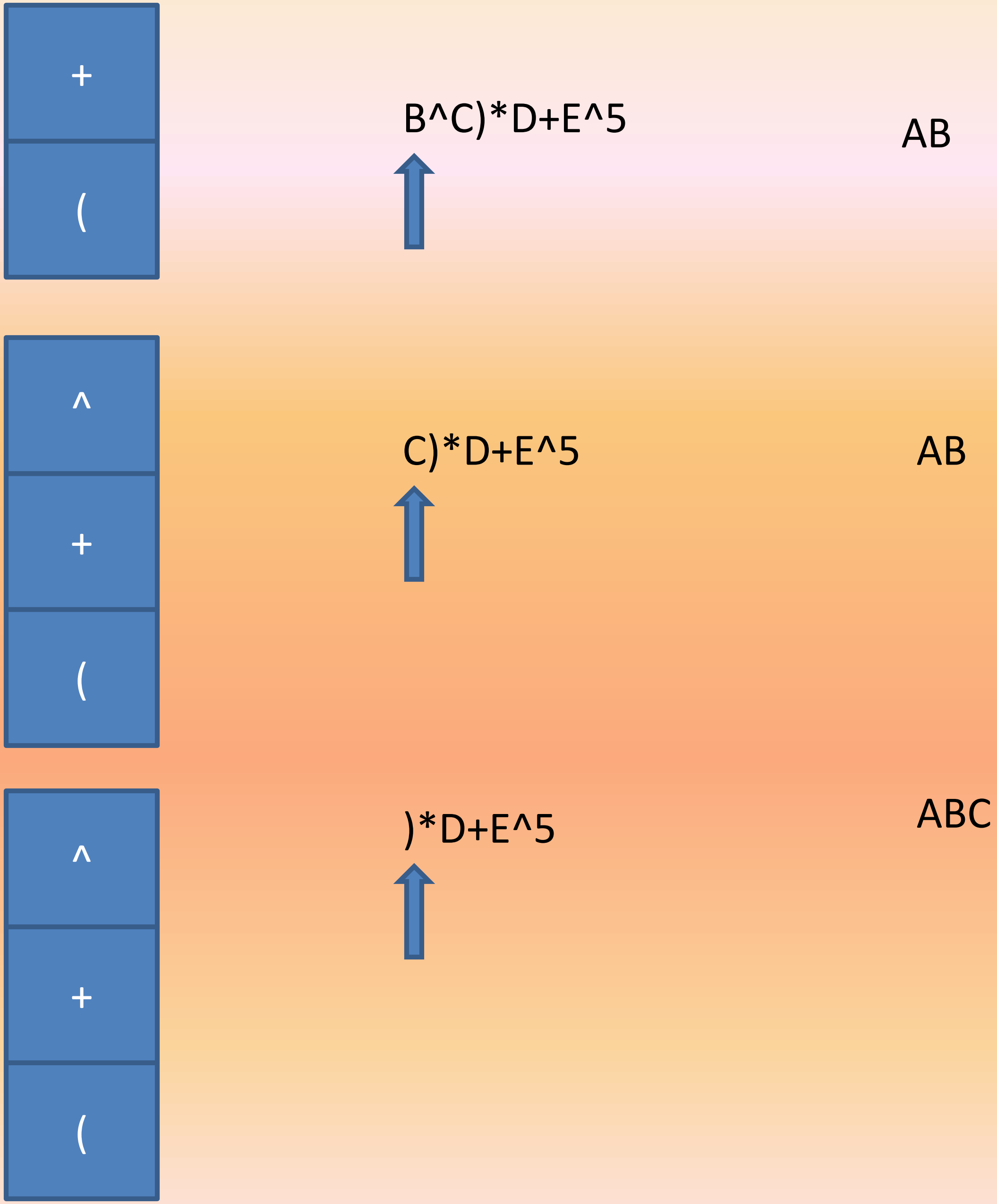
- **Manual** method
- **Fast** method
- Conversion using **Stack**

Algorithm/Psuedocode:

```
S:stack
While(more token)
{
X <-  next_token;
If(x is an operand)
    print x
Else
{
While(precedence(x)<=precedence(top(s))
    print(pop(s))
Push(s,x)
}
}
While(!empty(s))
    print(pop(s));
```

HOW TO CONVERT INFIX EXPRESSION TO THE POSTFIX EXPRESSION USING STACK ?

STACK	INPUT EXPRESSION	OUTPUT
EMPTY	(A+B^C)*D+E^5	--------
(	A+B^C)*D+E^5	---------
(	+B^C)*D+E^5	A



+
(

B^C)*D+E^5

AB

^
+
(

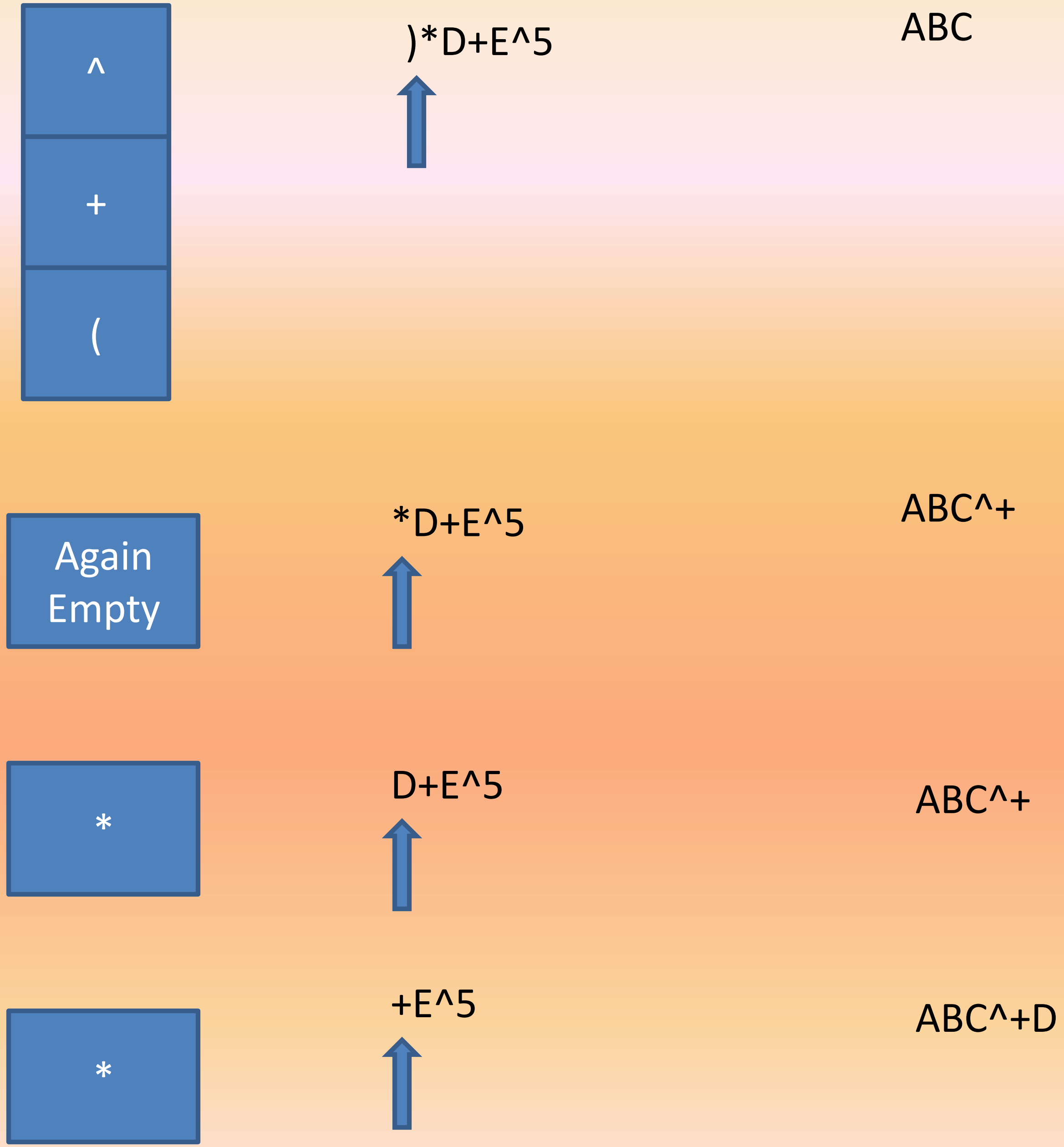
C)*D+E^5

AB

^
+
(

)*D+E^5

ABC



^
+
(

)*D+E^5

ABC

Again Empty

*D+E^5

ABC^+

*

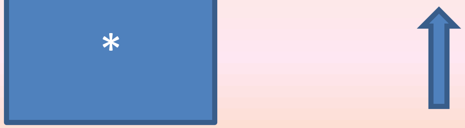
D+E^5

ABC^+

*

+E^5

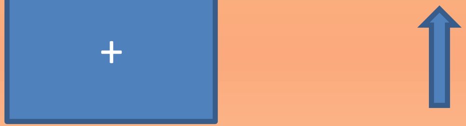
ABC^+D



+E^5

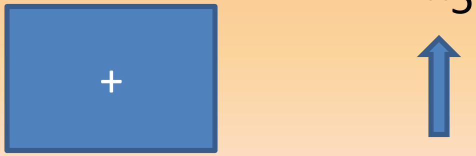
ABC^+D

As priority(*)> priority(+),then * will be popped from the stack and printed as output



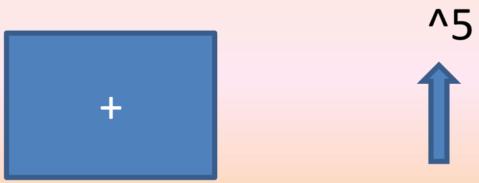
E^5

ABC^+D*



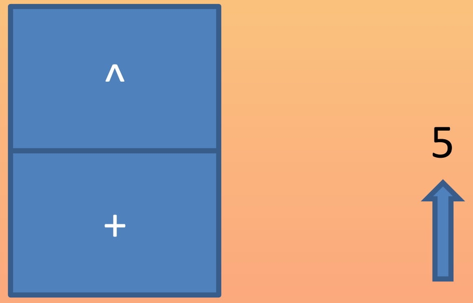
^5

ABC^+D*E



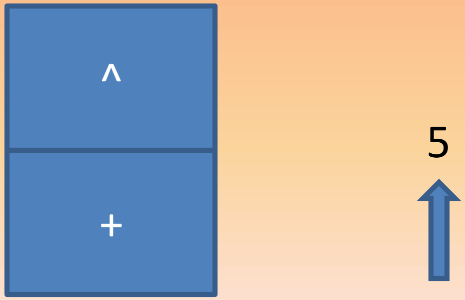
^5

ABC^+D*E



5

ABC^+D*E



5

ABC^+D*E5

Empty stack

\o expression ends here

ABC^+D*E5^+

Thank you!