

Evaluation of Postfix Expression

Algorithm for evaluation of postfix

Begin

Traverse from left, for each character in postfix expression,

Do if operand is encountered, push it on to stack,

else if operator is encountered, POP 2 elements

a -> top element

b -> next to top element

result = b operator a

push result on to the stack

Return element of top of stack

End

Evaluate the following postfix expression:

$$\begin{aligned} & 5\ 3\ +\ 6\ 2\ /\ * \ 3\ 5\ *\ + \\ &= 5+3\ 6\ 2\ /\ * \ 3\ 5\ *\ + \\ &= 8\ 6\ 2\ /\ * \ 3\ 5\ *\ + \\ &= 8\ 6/2\ * \ 3\ 5\ *\ + \\ &= 8\ 3\ * \ 3\ 5\ *\ + \\ &= 8*3\ 3\ 5\ *\ + \\ &= 24\ 3\ 5\ *\ + \\ &= 24\ 3*5\ + \\ &= 24\ 15\ + \\ &= 24+15 \\ &= 39 \end{aligned}$$

Algorithm for evaluation of prefix

Begin

Traverse from right, for each character in prefix expression,

Do if operand is encountered, push it on to stack,

else if operator is encountered, POP 2 elements

a -> top element

b -> next to top element

result = b operator a

push result on to the stack

Return element of top of stack

End