Infix, Prefix, and Postfix Notation

1. Evaluate the following **prefix** expressions. Work from right to left.
2. + + - 6 2 \* 5 3 / 8 4 b. \* + 8 / -12 \* 2 3 6
3. Evaluate the following **postfix** expressions. Work from left to right.   
     
   a. 6 3 / 9 7 4 3 - + + \* b. 8 5 + 6 2 / + 9 -
4. Use the following **infix** expressions to create an **expression tree**.  
     
   a. (2 \* 5) + (3 \* 7) – (10 / 2)
5. 10 + 2 \* 5 + 7

1. Given the following expression tree.

\*

/

-

+

+

A

B

C

D

E

F

1. Preform a **preorder** traversal to create a **prefix** expression.

1. Preform a **postorder** traversal to create a **postfix** expression.
2. Use the following prefix expression to create an expression tree, then use a **postorder** traversal to convert the **prefix** expression into a postfix expression.  
     
   prefix: - / A B + + C D E postfix: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Use the following prefix expression to create an expression tree, then use a **postorder** traversal to convert the **prefix** expression into a **postfix** expression.  
     
   prefix: \* + + A B C - D E postfix: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Use the following postfix expression to create an expression tree, then use a **preorder** traversal to convert the **postfix** expression into a **prefix** expression.

postfix: 7 5 – 4 2 + \* prefix: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Use the following postfix expression to create an expression tree, then use a **preorder** traversal to convert the **postfix** expression into a **prefix** expression.

postfix: 2 9 3 - + 5 7 4 + \* / prefix: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_