**DSF ASSIGNMENT 2**

**AIM :**

Construct an expression tree from postfix expression and perform non-recursive Inorder and Preorder traversals

**THEORY:**

**Algorithm :**

**Non-recursive Inorder traversal**

* 1) Create an empty stack S
* 2) Initialize *Current* node as root
* 3) Push the *Current* node to S; Set *Current* = *Current* ->left
* 4) Until *Current* is NULL
* 5) Push Current
* 6) Set *Current* = *Current* ->left
* 7) If current is NULL and stack is not empty then
* a) Pop from the stack S
* b) Print the popped item
* c) Set current = popped\_item->right
* d) Go to step 3
* 8) If current is NULL and stack is empty then return

**Non-recursive Preorder**

* Create an empty stack; make current=root
* Push current node to stack  
  Do following while is not empty

**a)** Pop an item from stack and print it.  
 **b)** Push right child of popped item to stack

**c)** Push left child of popped item to stack

*Note: Right child is pushed before left child to make sure that left subtree is processed first*

OUTPUT:

