* Assignment No-: 3*

* Title-) program for constructing a postorder Traversal.

* Objective:

To study and implement the postorder Traversal

* problem statements:-

for a given Exprences the postorder Traversal preorder and inforder it using a-b*c-d/e+f.

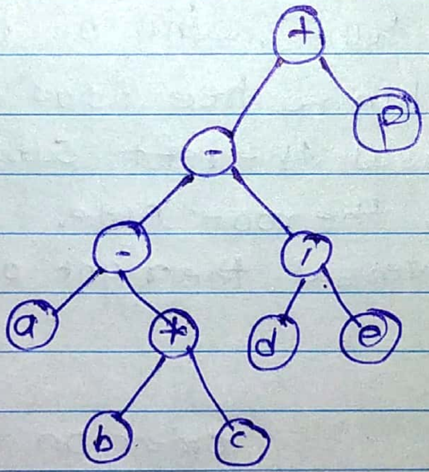
* Outcome:

•input :

Enter the valid expression

a-b*c-d/e+f

output



* Theory-:

Binary tree traversal (DFS) -: Most of the operation requires traversing a tree in a particular order. Traversing a tree and Exactly once.

Since, a binary tree is defined in a recursive manner. tree traversal could be defined recursivly.

for Example :- to traverse a tree ,once may visit the root first ,then the left subtree and finally traverse the right subtree. If we impose the restriction that left. subtree and finally traverse the right subtree .

1) Visit the root , traverse , left subtree , traverse right subtree.

2) Traverse left subtree, visit the root , traverse subtree.

3) Traverse left subtree , traverse right subtree ,visit the root.

These three technique of traversal are known as postorder , inorder. and preorder traversal of a binary tree.

1) Inordor traversal-:

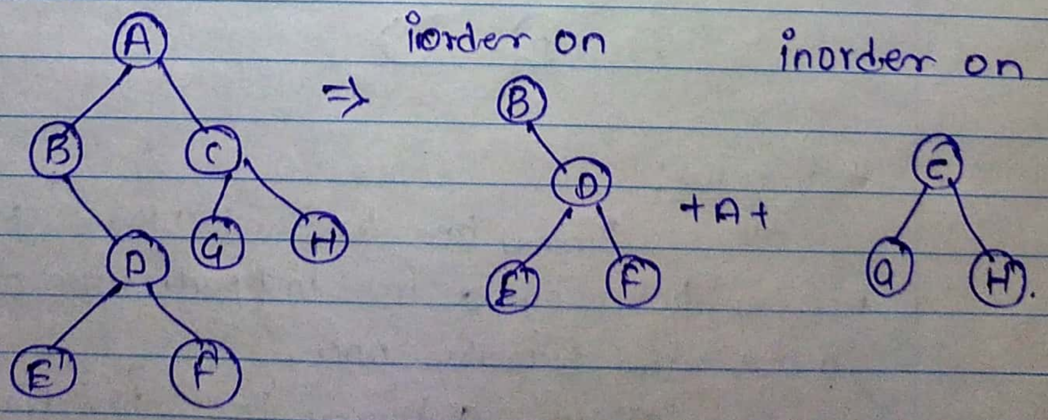
The functioning of inorder traversal of a non-empty . binary tree is as follows:-

i) firstly, traversal the left subtree in order.

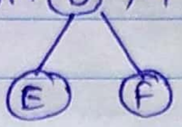
ii) Next ,Visit the root node.

iii) At lost, traversal the right of tree shown below

*inorder on :-



=(B+ inorder on D)+A+(inorder on G) +C+



(inorder on H)).

= (B+ (inorder on E)+D+ (inorder on F)) +A+(G(H))

= BEDFAGCH.

* postorder traversal -:

The functioning of postorder traversal of a non-empty binary tree is as follows:-

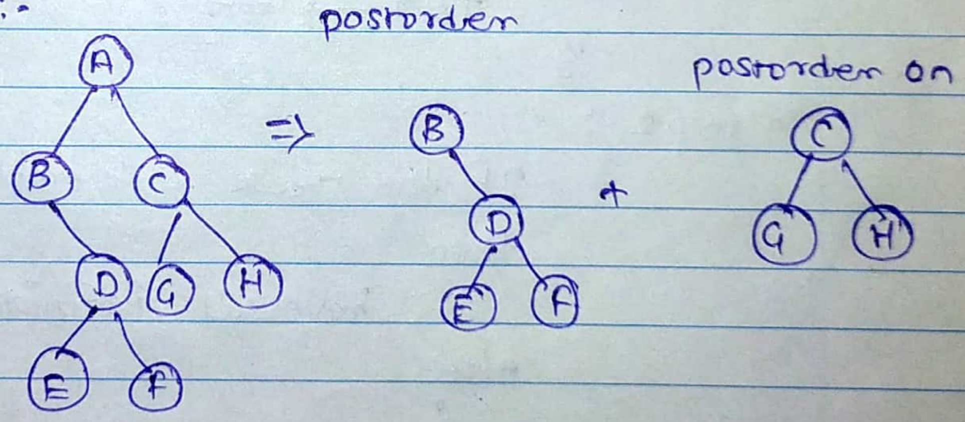
i) firstly, traversal the left subtree in postorders.

ii) Next. traverse the right subtree in postorder.

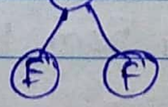
iii) At last, visit the root node.

Stepwise postorder traversal of tree shown below.

postorder on :-



=> ((postorder on B +B) +((postorder on G)+(postorder on H+C) +A.



=> EFDBGHCA.

Algorithm for non-recursive inorder traversal-:

Algorithm for non-recursive inorder traversal of binary tree too works similar to non-recursive preorder traversal , a node is visited before it is pushed into the visited immediately after, it is poped from the stock. In non-recursive inorder traversal a node is visited before it is pushed into the visited immediatly after. it is poped from the stock.

- step 1-: Start traversing from root (say T), traverse left and continue traversing left All the traversed are pused into the stack (say S).

```
while (T != NULL)
{
    S.push(T);
    T = T → left.
}
```

- Step 2-:

```
    If the stack S is empty
    then
        traversal is finished
    else
      {
        Visited the right subtree of the node
        poped from the stack.

      T = S.pop() ,Visit()
       T = T → right();
    while (T != NULL)
```

```
{
    S.push(T);
    T = T→left();
  }.
}.
```

Step3-: GotoStep 3.

* Algorithm for non-recursive postorder traversal.

Non-Recursive postorder traverse works in slightly different way. In positive traversal, element is visited after the right subtree is traversed. Thus, the address of the node should be pressered in stack until it has been printed. Non-Recurrise algorithm encounter a node three times:

1) while going to the left.
2) During backtrack to traversed to right subtree.
3) while returing from the right.

An additonal field 'flag' in stack is used to differentiate between the two students/situations.

- while going to left, address of the treenode along with flag=0 is pushed onto back with flag to 1

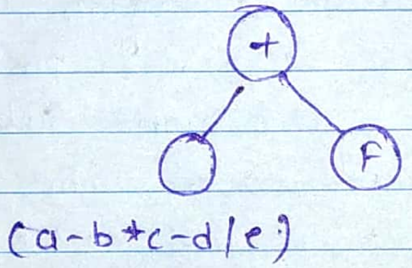
- During back tracking to traverse the right subtree. an element is poped from the stack and if flag field is found to be 0, it is pused back with flag to 1.

∘When we return from the right subtree and poped element from the steek ,the flag field will be 1 This is the time when we visit the node.

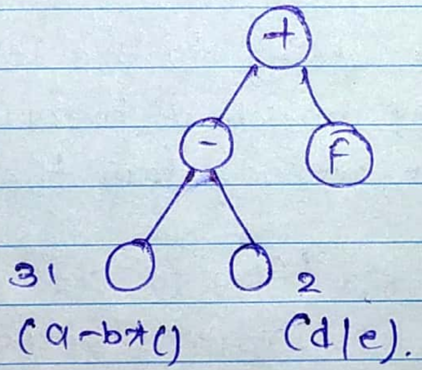
for given expression e.g a-b*c-d/e+f construct inorder sequence & traverse it using sequence & traverse it using postorder traversal.

The operator which evaluate the last make it as root node. Divide the Expression.

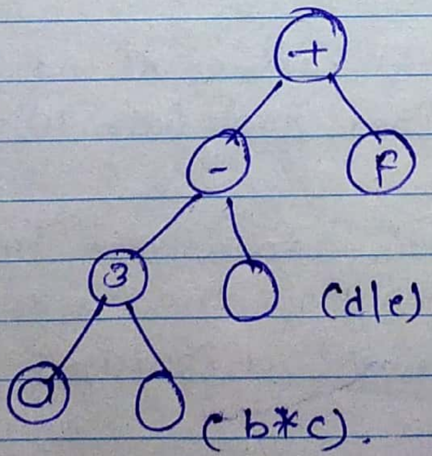
Step1



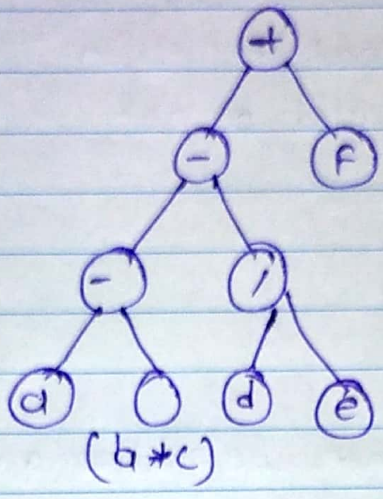
Step 2-:



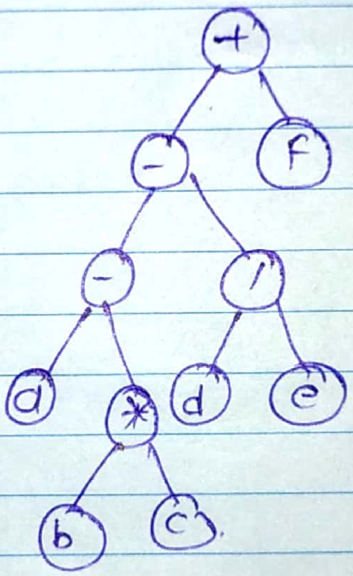
Step3-:



∘ Step :4



* Step 5-:



∘ postorder traversal -! ∴ abc*-de/-F+.

Test case -:

Test Id.	Test Scenario.	Test step.
TC-1.	i) check wheather tree is empty or not	check tree is (T==NULL) then it is empty

TC-2	ii) Check wheather externnal expression isvalid or not	check tree. expression check contain the valid Operand with valid operator.

* Conclusion :-

Hence, we studies and implementation the program for construct inorder sequence Epression using postorder traversal.