Assignment No. 5

Problem Statement: Construct an expression tree from the given prefix expression eg. +--a*bc/def and traverse it using postordertraversal (non recursive) and then delete the entire tree.

```
#include <iostream>
#include <stack>
#include <cctype>
using namespace std;
struct Node {
   char value;
   Node* left;
   Node* right;
   Node(char val) : value(val), left(nullptr), right(nullptr) {}
};
Node* newNode(char value) {
   return new Node(value);
}
Node* constructTree(const string& prefix) {
   stack<Node*> st;
   int n = prefix.length();
   for (int i = n - 1; i >= 0; i--) {
      Node* node = newNode(prefix[i]);

      if (isalpha(prefix[i])) {
         st.push(node);
      } else {
         node->left = st.top();
         st.pop();
```

```
            node->right = st.top();
            st.pop();
            st.push(node);
}
}
 return st.top();
}
void postOrderTraversal(Node* root) {
    if (!root) return;
    stack<Node*> s1, s2;
    s1.push(root);
    while (!s1.empty()) {
        Node* node = s1.top();
        s1.pop();
        s2.push(node);
        if (node->left) s1.push(node->left);
        if (node->right) s1.push(node->right);
    }
    while (!s2.empty()) {
        cout << s2.top()->value << " ";
        s2.pop();
    }
    cout << endl;
}
void deleteTree(Node* root) {
    if (!root) return;
    deleteTree(root->left);
    deleteTree(root->right);
```

```
    delete root;
}
int main() {
    string prefix = "+--a*bc/def";
    Node* root = constructTree(prefix);
    cout << "Postorder Traversal (Non-Recursive): ";
   postOrderTraversal(root);
    deleteTree(root);
    cout << "Tree deleted successfully." << endl;
    return 0;
}
```

Output:

```
Postorder Traversal (Non-Recursive): a b c * - d e f / - +
Tree deleted successfully.
```