

Name:Jadhav Prajakta Ganesh
Roll no: 2101043
Batch:A3

```
#include<iostream>
#include<stack>
#include<string>
using namespace std;
class node {
public:
    char val;
    node *left, *right;
};
node* newNode(char v) {
    node* temp = new node;
    temp->val = v;
    temp->left = NULL;
    temp->right = NULL;
    return temp;
}
bool isOperator(char c) {
    if (c == '+' || c == '-' || c == '*' || c == '/')
        return true;
    return false;
}
node* constructtree(string prefix) {
    stack<node*> st;
    node *t, *t1, *t2;
    for (int i = prefix.length() - 1; i >= 0; i--) {
        if (!isOperator(prefix[i])) {
            t = newNode(prefix[i]);
            st.push(t);
        } else {
            t = newNode(prefix[i]);
            t1 = st.top();
            st.pop();
            t2 = st.top();
            st.pop();
            t->left = t1;
            t->right = t2;
            st.push(t);
        }
    }
    t = st.top();
    st.pop();
    return t;
}
void postorder(node* ptr) {
    stack<node*> s1, s2;
    s1.push(ptr);
    node* temp;
    while (!s1.empty()) {
```

```

    temp = s1.top();
    s1.pop();
    s2.push(temp);
    if (temp->left != NULL)
        s1.push(temp->left);
    if (temp->right != NULL)
        s1.push(temp->right);
}
while (!s2.empty()) {
    temp = s2.top();
    s2.pop();
    cout << temp->val;
}
}
void deleteTree(node* root) {
    if (root == NULL)
        return;
    deleteTree(root->left);
    deleteTree(root->right);
    cout << endl << "Deleting node...." << root->val << endl;
    delete root;
    root = NULL;
}
int main() {
    string prefix;
    cout << "Enter the prefix expression: ";
    cin >> prefix;
    node* root = constructtree(prefix);
    cout << "Post-order traversal: ";
    postorder(root);
    cout << endl;
    deleteTree(root);
    root = NULL;
    return 0;
}

```

Output:

Enter the prefix expression: +*-/45678

Post-order traversal: 45-6/7*8+

Deleting node....4

Deleting node....5

Deleting node....-

Deleting node....6

Deleting node..../

Deleting node....7

Deleting node....*

Deleting node....8

Deleting node....+