Code:

```
#include <stack>
#include <iostream>
#include <string>
using namespace std;
struct node
{
  char data;
  node *l, *r;
};

class TREE
{
public:
  node *nnode, *root, *curr;
  stack<node *> s, s1;
  string exp;
  int n;
  TREE()
        {
    root = NULL;
  }

  void create()
        {
    cout << "\nEnter the expression: ";
    cin >> exp;
    n = exp.length();
    for (int i = n - 1; i >= 0; i--)
                {
      char ch = exp[i];
    nnode = new node;
    nnode->l = nnode->r = NULL;
    nnode->data = ch;
    if (ch >= 'a' && ch <= 'z')
                      {
      s.push(nnode);
    } else
                      {
                              if(s.size()<2)
                              {
                                      cout<<"\n Invalid expression";
                              }
                              else
      nnode->l = s.top();
      s.pop();
      nnode->r = s.top();
      s.pop();
      s.push(nnode);
    }
  }
  if (!s.empty())
              {
    root = s.top();
    s.pop();
  }
              else
              {
    root = NULL;
  }
}
```

```
void display(node *temp)
      {
  if (temp != NULL)
              {
    display(temp->l);
    display(temp->r);
    cout << temp->data << " ";
  }
}

void postorder(node *temp)
      {
  s.push(temp);
  while (!s.empty())
              {
    curr = s.top();
    s.pop();
    s1.push(curr);
    if (curr->l != NULL)
                      {
      s.push(curr->l);
    }
    if (curr->r != NULL)
                      {
      s.push(curr->r);
    }
  }
  while (!s1.empty())
              {
    curr = s1.top();
    s1.pop();
```

```
            cout << curr->data << " ";
        }
    }

    void deletet(node *temp)
            {
        if (temp != NULL)
                        {
            deletet(temp->l);
            deletet(temp->r);
            delete temp;
        }
    }
};

int main() {
    TREE ob;
    ob.create();
    cout << "\n Postorder traversal: ";
    ob.postorder(ob.root);
    cout << "\n Deleting tree...";
    ob.deletet(ob.root);
    cout << "\n Tree deleted successfully";
    return 0;
}
```

OUTPUT:

```
Enter the expression: +--a*bc/def

Postorder traversal: a b c * - d e / - f +

Deleting tree...

Tree deleted successfully
```