#include <iostream>

#include <stack>

#include <string>

using namespace std;

class Node {

public:

char data;

Node\* left;

Node\* right;

Node(char data) {

this->data = data;

this->left = this->right = nullptr;

}

};

Node\* Construct(const string& prefix) {

stack<Node\*> s;

for (int i = prefix.length() - 1; i >= 0; --i) {

char ch = prefix[i];

if (isalnum(ch)) {

Node\* node = new Node(ch);

s.push(node);

} else {

Node\* left = s.top();

s.pop();

Node\* right = s.top();

s.pop();

Node\* node = new Node(ch);

node->left = left;

node->right = right;

s.push(node);

}

}

return s.top();

}

void PostOT(Node\* root) {

if (root == nullptr) {

return;

}

stack<Node\*> s;

s.push(root);

Node\* prev = nullptr;

while (!s.empty()) {

Node\* curr = s.top();

if (prev == nullptr || prev->left == curr || prev->right == curr) {

if (curr->left != nullptr) {

s.push(curr->left);

}

} else if (curr->left == prev) {

if (curr->right != nullptr) {

s.push(curr->right);

}

} else {

cout << curr->data;

s.pop();

}

prev = curr;

}

}

void Delete(Node\* root) {

if (root == nullptr) {

return;

}

Delete(root->left);

Delete(root->right);

cout << "Deleting node: " << root->data << endl;

delete root;

}

int main() {

string prefix;

cout << "Enter the prefix expression: ";

getline(cin, prefix);

Node\* root = Construct(prefix);

cout << "Post Order Traversal: ";

PostOT(root);

cout << endl;

Delete(root);

cout << "Tree deleted successfully." << endl;

return 0;

}