

# TRAVERSING BINARY TREE

## INPUT:

```
#include <stdlib.h>

#include <iostream>

using namespace std;

struct node {

    int data;

    struct node *left;

    struct node *right;

};

// New node creation

struct node *newNode(int data) {

    struct node *node = (struct node *)malloc(sizeof(struct node));

    node->data = data;

    node->left = NULL;

    node->right = NULL;

    return (node);

}

// Traverse Preorder

void traversePreOrder(struct node *temp) {

    if (temp != NULL) {

        cout << " " << temp->data;

        traversePreOrder(temp->left);

        traversePreOrder(temp->right);

    }

}
```

```
}  
}
```

```
// Traverse Inorder
```

```
void traverseInOrder(struct node *temp) {  
    if (temp != NULL) {  
        traverseInOrder(temp->left);  
        cout << " " << temp->data;  
        traverseInOrder(temp->right);  
    }  
}
```

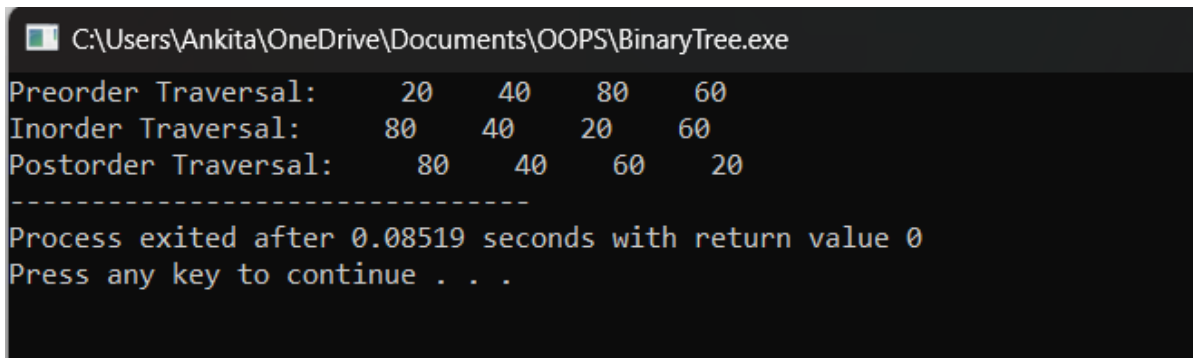
```
// Traverse Postorder
```

```
void traversePostOrder(struct node *temp) {  
    if (temp != NULL) {  
        traversePostOrder(temp->left);  
        traversePostOrder(temp->right);  
        cout << " " << temp->data;  
    }  
}
```

```
int main() {  
    struct node *root = newNode(20);  
    root->left = newNode(40);  
    root->right = newNode(60);  
    root->left->left = newNode(80);  
    cout << "Preorder Traversal: ";  
    traversePreOrder(root);  
    cout << "\nInorder Traversal: ";
```

```
traverseInOrder(root);  
cout << "\nPostorder Traversal: ";  
traversePostOrder(root);  
return 0;  
}
```

### **OUTPUT:**



```
C:\Users\Ankita\OneDrive\Documents\OOPS\BinaryTree.exe  
Preorder Traversal:    20    40    80    60  
Inorder Traversal:     80    40    20    60  
Postorder Traversal:   80    40    60    20  
-----  
Process exited after 0.08519 seconds with return value 0  
Press any key to continue . . .
```