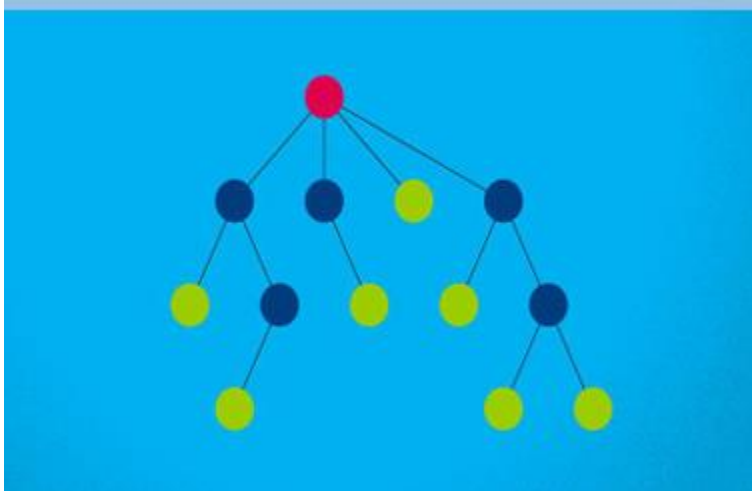


Name	Said Rasool
Sap	5691
sec	Se3-2

DATA STRUCTURE AND ARGORITRM:

INPUT:



CODE.1:

```

#include <iostream>

using namespace std;

struct Node {
    int data;
    Node* left;
    Node* right;
};

Node* createNode(int value) {
    Node* newNode = new Node();
    newNode->data = value;

```

```
newNode->left = newNode->right = NULL;

return newNode;

}
```

```
Node* insert(Node* root, int value) {

    if (root == NULL) {

        return createNode(value);

    }

    if (value < root->data) {

        root->left = insert(root->left, value);

    } else if (value > root->data) {

        root->right = insert(root->right, value);

    }

    return root;

}
```

```
void InOrderTraversal(Node* root) {

    if (root == NULL) return;

    InOrderTraversal(root->left);

    cout << root->data << " ";

    InOrderTraversal(root->right);

}
```

```
void PreOrderTraversal(Node* root) {
```

```
    if (root == NULL) return;

    cout << root->data << " ";

    PreOrderTraversal(root->left);

    PreOrderTraversal(root->right);

}
```

```
void PostOrderTraversal(Node* root) {

    if (root == NULL) return;

    PostOrderTraversal(root->left);

    PostOrderTraversal(root->right);

    cout << root->data << " ";

}
```

```
int main() {

    Node* root = NULL;

    root = insert(root, 50);

    root = insert(root, 30);

    root = insert(root, 10);

    root = insert(root, 20);

    root = insert(root, 40);

    root = insert(root, 60);

    root = insert(root, 80);

    cout << "In-Order traversal of the BST: ";
```

```
InOrderTraversal(root);

cout << endl;


cout << "Pre-Order traversal of the BST: ";

PreOrderTraversal(root);

cout << endl;


cout << "Post-Order traversal of the BST: ";

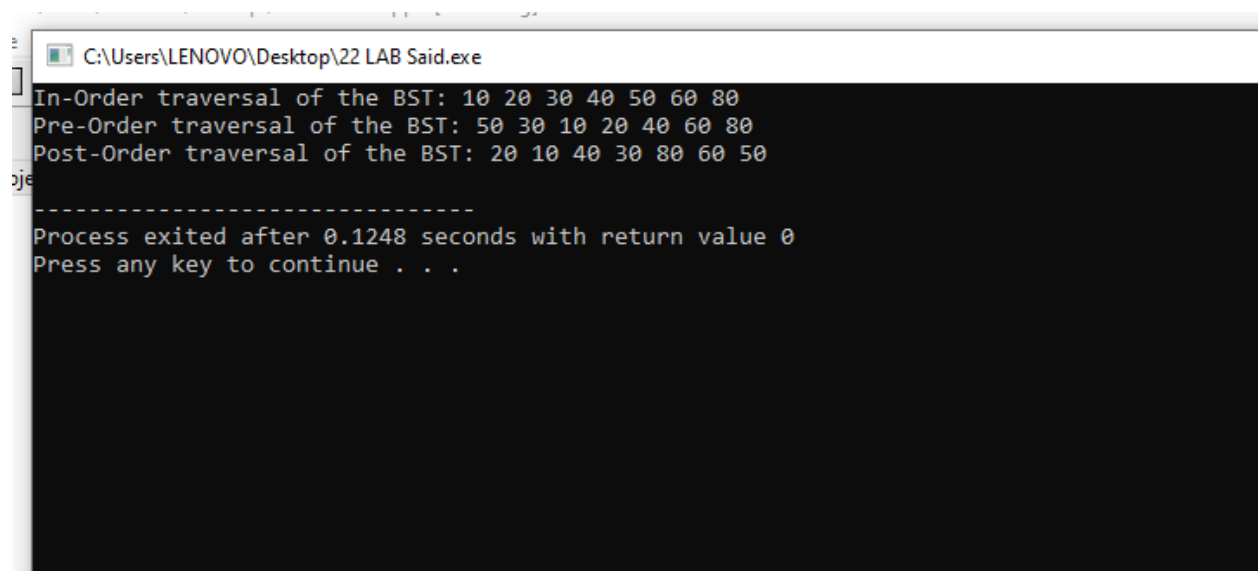
PostOrderTraversal(root);

cout << endl;


return 0;

}
```

OUTPUT:



```
C:\Users\LENOVO\Desktop\22 LAB Said.exe
In-Order traversal of the BST: 10 20 30 40 50 60 80
Pre-Order traversal of the BST: 50 30 10 20 40 60 80
Post-Order traversal of the BST: 20 10 40 30 80 60 50
-----
Process exited after 0.1248 seconds with return value 0
Press any key to continue . . .
```

Code.2:

Input:

```
#include <iostream>
```

```
using namespace std;
```

```
struct Node {
```

```
    int data;
```

```
    Node* left;
```

```
    Node* right;
```

```
};
```

```
Node* createNode(int value) {
```

```
    Node* newNode = new Node();
```

```
    newNode->data = value;
```

```
    newNode->left = newNode->right = NULL;
```

```
    return newNode;
```

```
}
```

```
Node* insert(Node* root, int value) {
```

```
    if (root == NULL) {
```

```
        return createNode(value);
```

```
    }
```

```
    if (value < root->data) {
```

```
        root->left = insert(root->left, value);
```

```
    } else if (value > root->data) {
```

```
        root->right = insert(root->right, value);
```

```
    }  
    return root;  
}
```

```
void InOrderTraversal(Node* root) {  
    if (root == NULL) return;  
    InOrderTraversal(root->left);  
    cout << root->data << " ";  
    InOrderTraversal(root->right);  
}
```

```
void PreOrderTraversal(Node* root) {  
    if (root == NULL) return;  
    cout << root->data << " ";  
    PreOrderTraversal(root->left);  
    PreOrderTraversal(root->right);  
}
```

```
void PostOrderTraversal(Node* root) {  
    if (root == NULL) return;  
    PostOrderTraversal(root->left);  
    PostOrderTraversal(root->right);  
    cout << root->data << " ";  
}
```

```
int main() {  
  
    Node* root = NULL;  
  
    int value;  
  
    char choice;  
  
    do {  
  
        cout << "Enter a value to insert into the BST: ";  
  
        cin >> value;  
  
        root = insert(root, value);  
  
        cout << "Do you want to insert another value? (y/n): ";  
  
        cin >> choice;  
    } while (choice == 'y' || choice == 'Y');  
  
    cout << "In-Order traversal of the BST: ";  
  
    InOrderTraversal(root);  
  
    cout << endl;  
  
    cout << "Pre-Order traversal of the BST: ";  
  
    PreOrderTraversal(root);  
  
    cout << endl;  
  
    cout << "Post-Order traversal of the BST: ";
```

```

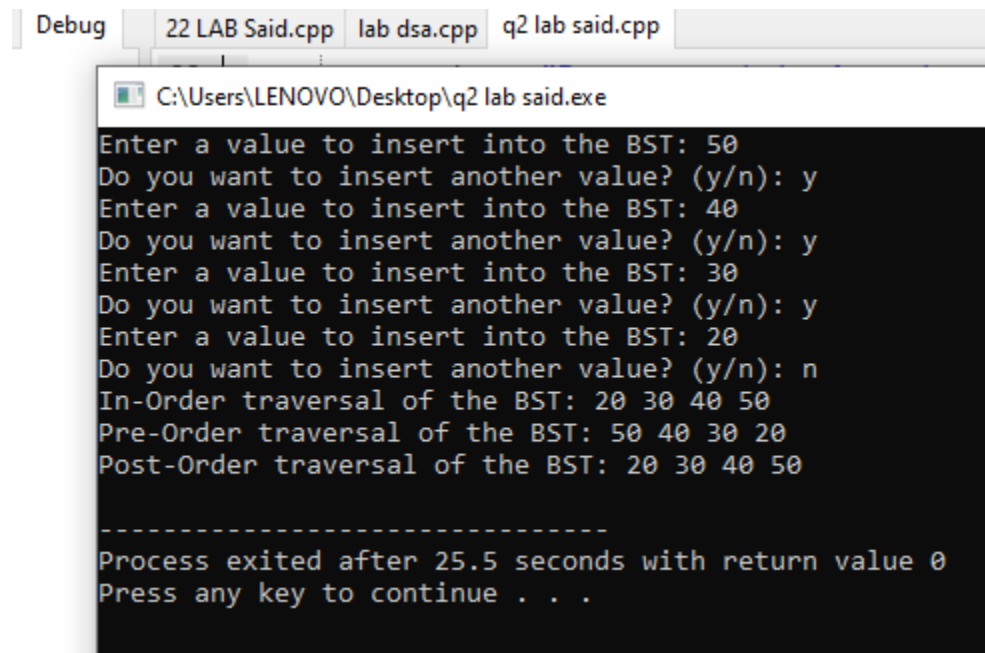
    PostOrderTraversal(root);

    cout << endl;

    return 0;
}

```

Output:



The screenshot shows a Windows IDE with three tabs: '22 LAB Said.cpp', 'lab dsa.cpp', and 'q2 lab said.cpp'. The active window is 'q2 lab said.exe' at the path 'C:\Users\LENOVO\Desktop\q2 lab said.exe'. The output is as follows:

```

Enter a value to insert into the BST: 50
Do you want to insert another value? (y/n): y
Enter a value to insert into the BST: 40
Do you want to insert another value? (y/n): y
Enter a value to insert into the BST: 30
Do you want to insert another value? (y/n): y
Enter a value to insert into the BST: 20
Do you want to insert another value? (y/n): n
In-Order traversal of the BST: 20 30 40 50
Pre-Order traversal of the BST: 50 40 30 20
Post-Order traversal of the BST: 20 30 40 50

-----
Process exited after 25.5 seconds with return value 0
Press any key to continue . . .

```

Code.3:

Input:

```

#include <iostream>

using namespace std;

struct Node {
    int data;

    Node* left;

    Node* right;
}

```



```
};
```

```
Node* createNode(int value) {  
    Node* newNode = new Node();  
    newNode->data = value;  
    newNode->left = newNode->right = NULL;  
    return newNode;  
}
```

```
Node* insert(Node* root, int value) {  
    if (root == NULL) {  
        return createNode(value);  
    }  
    if (value < root->data) {  
        root->left = insert(root->left, value);  
    } else if (value > root->data) {  
        root->right = insert(root->right, value);  
    }  
    return root;  
}
```

```
bool search(Node* root, int value) {  
    if (root == NULL) {  
        return false;  
    }
```

```
}  
  
if (value == root->data) {  
    return true;  
} else if (value < root->data) {  
    return search(root->left, value);  
} else {  
    return search(root->right, value);  
}  
}
```

```
int main() {  
    Node* root = NULL;  
  
    root = insert(root, 50);  
    root = insert(root, 30);  
    root = insert(root, 10);  
    root = insert(root, 20);  
    root = insert(root, 40);  
    root = insert(root, 60);  
    root = insert(root, 80);  
  
    int value;  
  
    cout << "Enter a value to search in the BST: ";  
  
    cin >> value;  
  
    if (search(root, value)) {
```

```

        cout << "Value " << value << " found in the BST." << endl;
    } else {
        cout << "Value " << value << " not found in the BST." << endl;
    }

    return 0;
}

```

Output:

```

cpp lab dsa.cpp q2 lab said.cpp q3dsa said.cpp q4said dsa.cpp
root = insert(root, 30);
root = insert(root, 10);
C:\Users\LENOVO\Desktop\q4said dsa.exe
Enter a value to search in the BST: 40
Value 40 found in the BST.
Process exited after 5.894 seconds with return value 0
Press any key to continue . . .

```

Code.4:

Input:

```

#include <iostream>

using namespace std;

struct Node {
    int data;
    Node* left;

```

```
Node* right;  
};
```

```
Node* createNode(int value) {  
    Node* newNode = new Node();  
    newNode->data = value;  
    newNode->left = newNode->right = NULL;  
    return newNode;  
}
```

```
Node* insert(Node* root, int value) {  
    if (root == NULL) {  
        return createNode(value);  
    }  
    if (value < root->data) {  
        root->left = insert(root->left, value);  
    } else if (value > root->data) {  
        root->right = insert(root->right, value);  
    }  
    return root;  
}
```

```
bool search(Node* root, int value) {  
    if (root == NULL) {
```

```
        return false;
    }
    if (value == root->data) {
        return true;
    } else if (value < root->data) {
        return search(root->left, value);
    } else {
        return search(root->right, value);
    }
}
```

```
int main() {
    Node* root = NULL;
    int value;
    char choice;

    do {
        cout << "Enter a value to insert into the BST: ";
        cin >> value;
        root = insert(root, value);

        cout << "Do you want to insert another value? (y/n): ";
        cin >> choice;
    } while (choice == 'y' || choice == 'Y');
```

```
cout << "Enter a value to search in the BST: ";  
  
cin >> value;  
  
if (search(root, value)) {  
    cout << "Value " << value << " found in the BST." << endl;  
} else {  
    cout << "Value " << value << " not found in the BST." << endl;  
}  
  
return 0;  
}
```

Output:

```
lab dsa.cpp  q2 lab said.cpp  q3dsa said.cpp  q4said dsa.cpp
cin >> value;
root = insert(root, value);

cout << "
cin
while
cout <<
cin >> v
if (search
cout
else {
cout
Process exited after 24.43 seconds with return value
Press any key to continue . . .
return 0
```

Code.5:

Input:

```
#include <iostream>
```

```
using namespace std;
```

```
struct Node
```

```
{
```

```
int data;    // Data stored in the node
```

```
Node *left;  // Pointer to the left child node
```

```
Node *right; // Pointer to the right child node
```

```
};
```

```
Node *createNode(int value)
```

```
{
```

```
    Node *newNode = new Node();
```

```
    newNode->data = value;
```

```
    newNode->left = NULL;
```

```
    newNode->right = NULL;
```

```
    return newNode;
```

```
}
```

```
// Function to insert a node into the BST
```

```
Node *insert(Node *root, int value)
```

```
{
```

```
    if (root == NULL)
```

```
    {
```

```
        return createNode(value);
```

```
    }
```

```
    if (value < root->data)
```

```
    {
```

```
        root->left = insert(root->left, value);
```

```
    }
```

```
    else if (value > root->data)
```

```
    {
```



```
        root->right = insert(root->right, value);
    }

    return root;
}
```

```
void InOrderTraversal(Node *root)
{
    if (root == NULL) return;
    InOrderTraversal(root->left);
    cout << root->data << " ";
    InOrderTraversal(root->right);
}
```

```
void PreOrderTraversal(Node *root)
{
    if (root == NULL) return;
    cout << root->data << " ";
    PreOrderTraversal(root->left);
    PreOrderTraversal(root->right);
}
```

```
void PostOrderTraversal(Node *root)
{
    if (root == NULL) return;
    PostOrderTraversal(root->left);
```

```
    PostOrderTraversal(root->right);  
    cout << root->data << " ";  
}
```

```
void deleteTree(Node* root) {  
    if (root != NULL) {  
        deleteTree(root->left);  
        deleteTree(root->right);  
        delete root;  
    }  
}
```

```
int main()  
{  
    Node *root = NULL;  
    int numNodes, value;  
  
    cout << "Enter the number of nodes to insert: ";  
    cin >> numNodes;  
  
    for (int i = 0; i < numNodes; i++)  
    {  
        cout << "Enter value for Node " << (i + 1) << ": ";
```

```
    cin >> value;

    root = insert(root, value);
}
```

```
cout << "\nIn-order Traversal: ";

InOrderTraversal(root);

cout << endl;
```

```
cout << "Pre-order Traversal: ";

PreOrderTraversal(root);

cout << endl;
```

```
cout << "Post-order Traversal: ";

PostOrderTraversal(root);

cout << endl;
```

```
deleteTree(root);
```

```
return 0;

}
```

Output:

22 LAB Said.cpp lab dsa.cpp q2 lab said.cpp q3dsa said.cpp q4said dsa.cpp q5 said dsa.cpp [*] q55d:

87

88

cout << "\nIn-order Traversal: ";

C:\Users\LENOVO\Desktop\q55dsa said.exe

Enter the number of nodes to insert: 4

Enter value for Node 1: 3

Enter value for Node 2: 4

Enter value for Node 3: 5

Enter value for Node 4: 7

In-order Traversal: 3 4 5 7

Pre-order Traversal: 3 4 5 7

Post-order Traversal: 7 5 4 3

Process exited after 21.01 seconds with return value 0

Press any key to continue . . .