```
#include<bits/stdc++.h>
using namespace std;
                                      void PrintPreorder(BST* root)
struct BST
                                          if (root == NULL)
                                               return;
    int data;
                                          cout << root->data << " ";</pre>
    BST *left_child, *right_child;
                                          PrintPreorder(root-
    BST(int value)
                                      >left child);
                                          PrintPreorder(root-
        data = value;
                                      >right child);
        left child = NULL;
        right child = NULL;
                                      void PrintPostorder(BST* root)
    }
};
                                           if (root == NULL)
                                               return;
BST* InsertNODE(BST* root, int
                                          PrintPostorder(root-
inserting value)
                                      >left child);
                                          PrintPostorder(root-
{
    if (root == NULL)
                                      >right_child);
                                          cout << root->data << " ";</pre>
        return new
BST(inserting value);
                                      }
   else if (inserting value <
root->data)
                                      int main()
        root->left child =
                                      {
InsertNODE(root->left child,
                                          BST* root = NULL;
inserting value);
                                          cout << "Enter the number of</pre>
    else
                                      inserting values: ";
        root->right_child =
                                           int n;
InsertNODE(root->right child,
                                          cin >> n;
inserting value);
                                          while (n--)
    return root;
                                          {
}
                                               int value;
                                               cin >> value;
bool Search(BST* root, int value)
                                               root = InsertNODE(root,
                                      value);
    if (root == NULL)
                                           }
        return false;
  else if (root->data == value)
                                          cout << "In-order traversal of</pre>
        return true;
                                      the BST: ";
  else if (value < root->data)
                                          PrintInorder(root);
        return Search(root-
                                          cout << endl;</pre>
>left child, value);
    else
                                           cout << "Pre-order traversal</pre>
        return Search(root-
                                      of the BST: ";
>right child, value);
                                          PrintPreorder(root);
                                          cout << endl;</pre>
```

```
void PrintInorder(BST* root)
                                            cout << "Post-order traversal</pre>
                                        of the BST: ";
    if (root == NULL)
                                            PrintPostorder(root);
                                            cout << endl;</pre>
         return;
    PrintInorder(root-
                                            cout << "\nEnter a value to</pre>
>left child);
                                        search for: ";
    cout << root->data << " ";</pre>
    PrintInorder(root-
                                            int searchValue;
>right child);
                                            cin >> searchValue;
                                            if (Search(root, searchValue))
        cout << searchValue << "</pre>
is found in the BST." << endl;
                                                 cout << searchValue << "</pre>
                                        is not found in the BST." << endl;
    else
                                            return 0;
```

```
iearch

"D:\CSE\Algorithm_Lab\A_N_S_30_383\Binary Search Tree.exe"

inclEnter the number of inserting values: 10

Ising 1 5 9 3 5 7 4 6 2 8

In-order traversal of the BST: 1 2 3 4 5 5 6 7 8 9

Pre-order traversal of the BST: 1 5 3 2 4 9 5 7 6 8

truc

Post-order traversal of the BST: 2 4 3 6 8 7 5 9 5 1

intEnter a value to search for: 9

BS9 is found in the BST.

BS

Process returned 0 (0x0) execution time: 26.069 s

Press any key to continue.
```