|  |  |
| --- | --- |
| #include<bits/stdc++.h>  using namespace std;  struct BST  {  int data;  BST \*left\_child, \*right\_child;  BST(int value)  {  data = value;  left\_child = NULL;  right\_child = NULL;  }  };  BST\* InsertNODE(BST\* root, int inserting\_value)  {  if (root == NULL)  return new BST(inserting\_value);  else if (inserting\_value < root->data)  root->left\_child = InsertNODE(root->left\_child, inserting\_value);  else  root->right\_child = InsertNODE(root->right\_child, inserting\_value);  return root;  }  bool Search(BST\* root, int value)  {  if (root == NULL)  return false;  else if (root->data == value)  return true;  else if (value < root->data)  return Search(root->left\_child, value);  else  return Search(root->right\_child, value);  }  void PrintInorder(BST\* root)  {  if (root == NULL)  return;  PrintInorder(root->left\_child);  cout << root->data << " ";  PrintInorder(root->right\_child);  }  cout << searchValue << " is found in the BST." << endl;  else | void PrintPreorder(BST\* root)  {  if (root == NULL)  return;  cout << root->data << " ";  PrintPreorder(root->left\_child);  PrintPreorder(root->right\_child);  }  void PrintPostorder(BST\* root)  {  if (root == NULL)  return;  PrintPostorder(root->left\_child);  PrintPostorder(root->right\_child);  cout << root->data << " ";  }  int main()  {  BST\* root = NULL;  cout << "Enter the number of inserting values: ";  int n;  cin >> n;  while (n--)  {  int value;  cin >> value;  root = InsertNODE(root, value);  }  cout << "In-order traversal of the BST: ";  PrintInorder(root);  cout << endl;  cout << "Pre-order traversal of the BST: ";  PrintPreorder(root);  cout << endl;  cout << "Post-order traversal of the BST: ";  PrintPostorder(root);  cout << endl;  cout << "\nEnter a value to search for: ";  int searchValue;  cin >> searchValue;  if (Search(root, searchValue))  cout << searchValue << " is not found in the BST." << endl;  return 0;  } |

