Write a program

a) ToconstructabinarySearchtree.

b) To traverse the tree using all the methods i.e., in-order, preorder and post order

c) To display the elements in the tree.

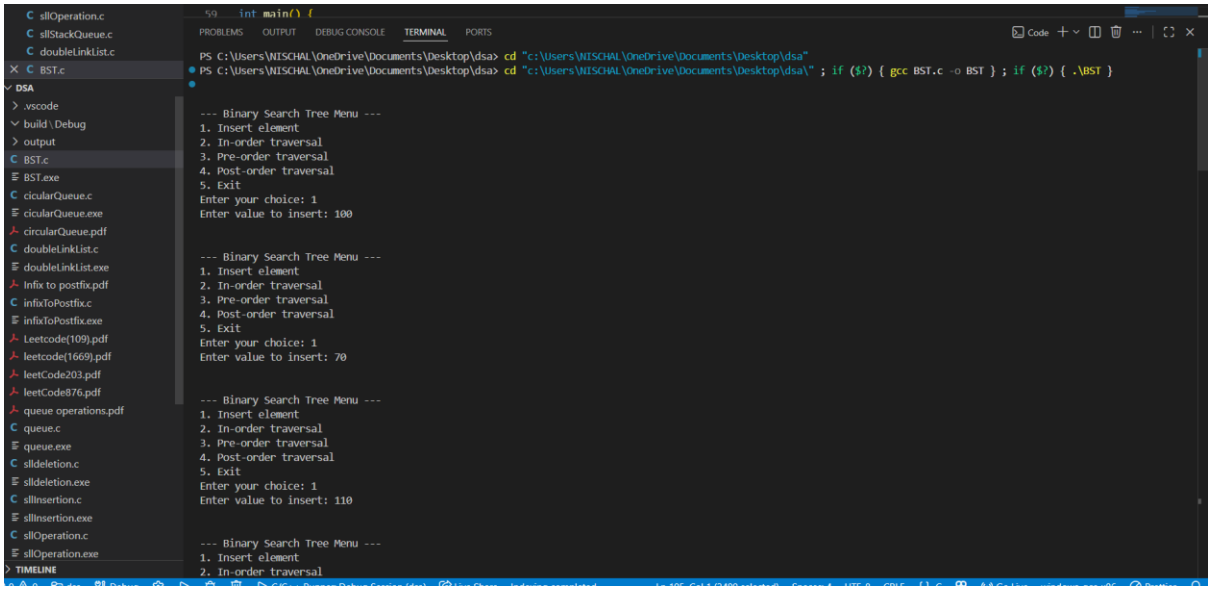
```
#include <stdio.h>
#include <stdlib.h>

/* Structure of a BST node */
struct node {
   int data;
   struct node *left, *right;
};

/* Create a new node */
struct node* createNode(int value) {
   struct node* newNode = (struct node*)malloc(sizeof(struct node));
   newNode->data = value;
   newNode->left = NULL;
   newNode->right = NULL;
   return newNode;
}

/* Insert a node into BST */
struct node* insert(struct node* root, int value) {
   if (root == NULL)
      return createNode(value);
```

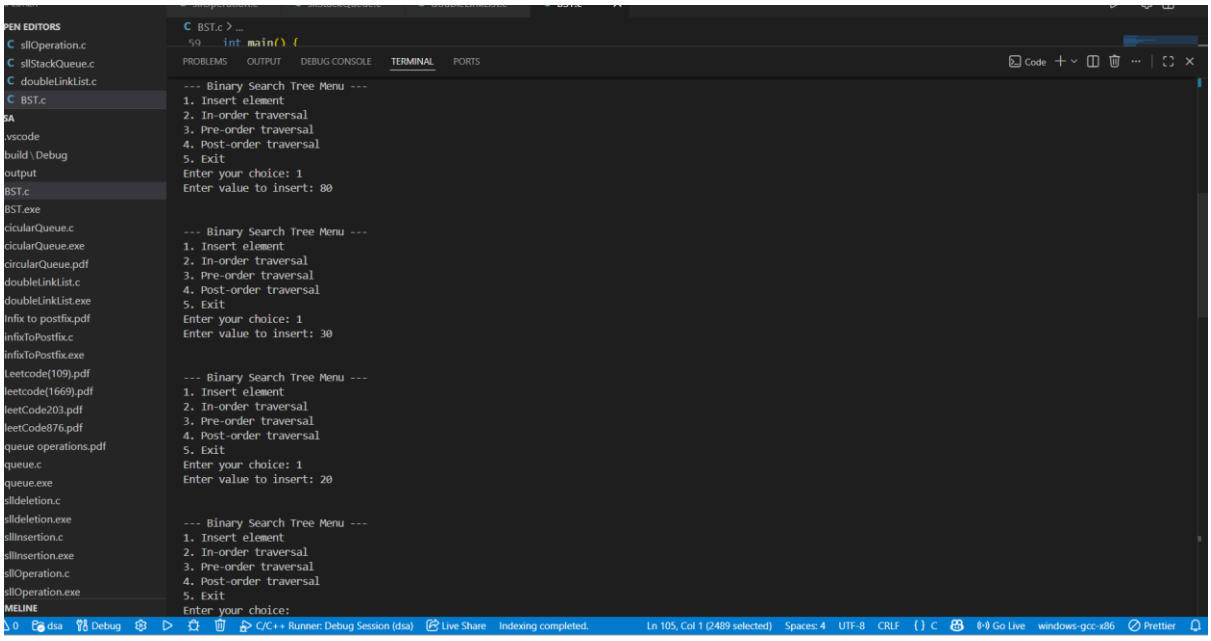
```
    if (value < root->data)
        root->left = insert(root->left, value);
    else if (value > root->data)
        root->right = insert(root->right, value);

    return root;
}

/* In-order Traversal */
void inorder(struct node* root) {
    if (root != NULL) {
        inorder(root->left);
        printf("%d ", root->data);
        inorder(root->right);
    }
}

/* Pre-order Traversal */
void preorder(struct node* root) {
    if (root != NULL) {
        printf("%d ", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}

/* Post-order Traversal */
void postorder(struct node* root) {
    if (root != NULL) {
```

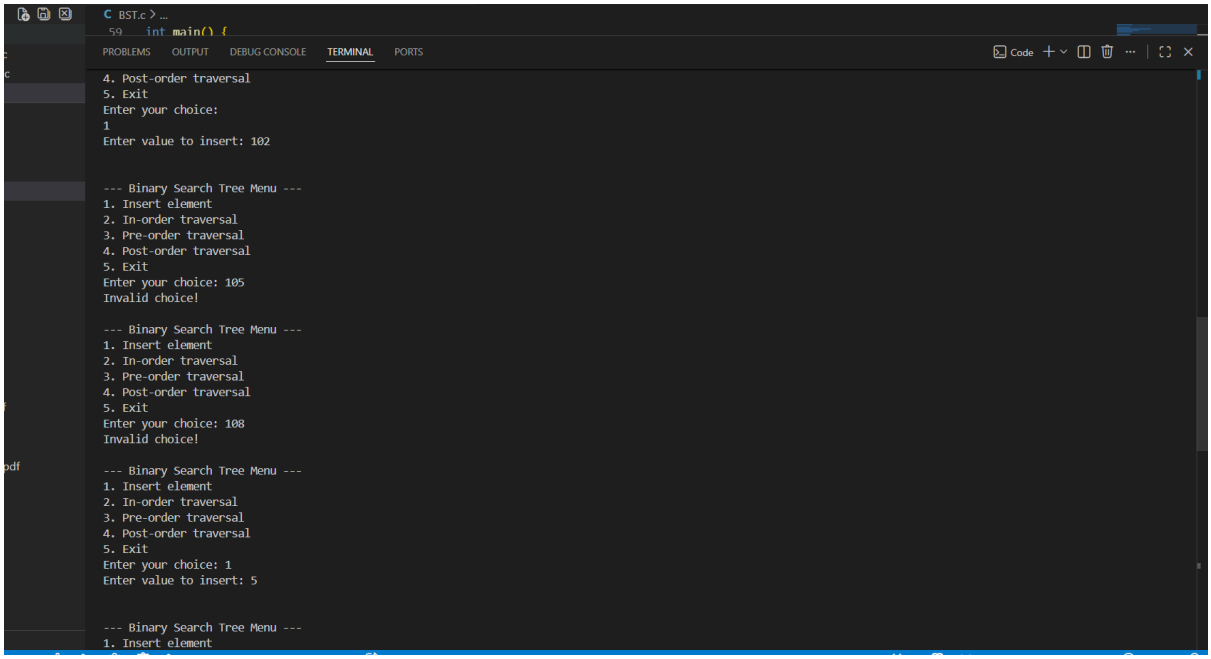
```
    postorder(root->left);
    postorder(root->right);
    printf("%d ", root->data);
  }
}

int main() {
  struct node* root = NULL;
  int choice, value;

  while (1) {
    printf("\n\n--- Binary Search Tree Menu ---");
    printf("\n1. Insert element");
    printf("\n2. In-order traversal");
    printf("\n3. Pre-order traversal");
    printf("\n4. Post-order traversal");
    printf("\n5. Exit");
    printf("\nEnter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
      case 1:
        printf("Enter value to insert: ");
        scanf("%d", &value);
        root = insert(root, value);
        break;

      case 2:
        printf("In-order traversal: ");
```

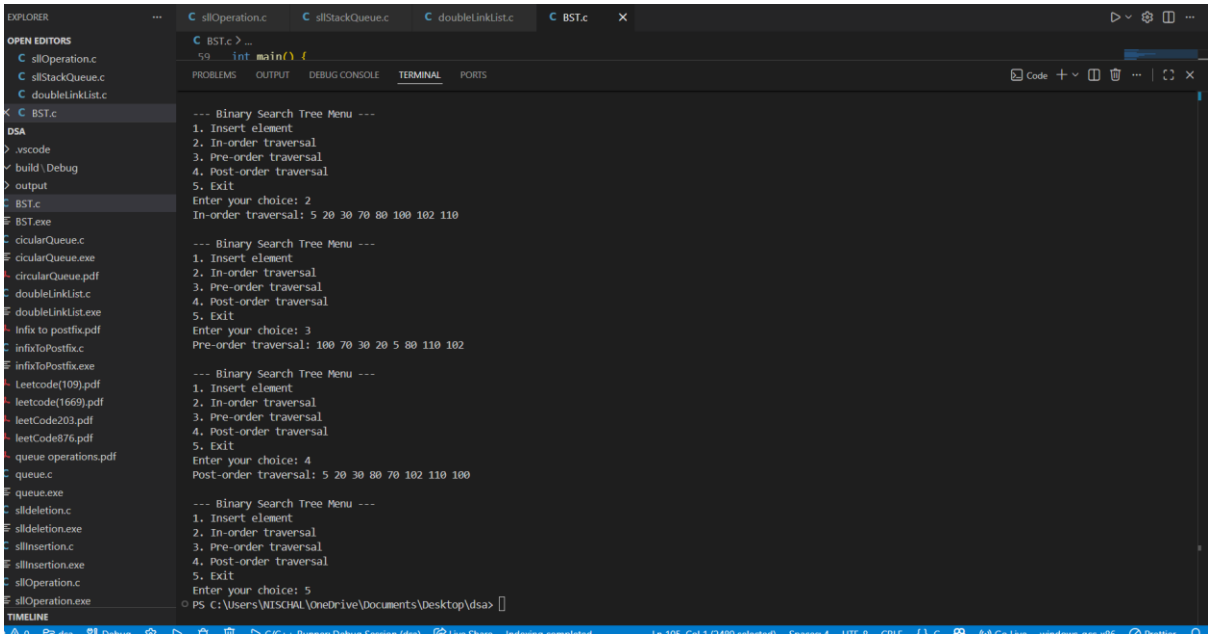
```
            inorder(root);
            break;

        case 3:
            printf("Pre-order traversal: ");
            preorder(root);
            break;

        case 4:
            printf("Post-order traversal: ");
            postorder(root);
            break;

        case 5:
            exit(0);

        default:
            printf("Invalid choice!");
      }
   }

   return 0;
}
```

OUTPUT:


int main() {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa> cd "c:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa"
PS C:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa> cd "c:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa\" ; if ($?) { gcc BST.c -o BST } ; if ($?) { .\BST }
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 1
Enter value to insert: 100
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 1
Enter value to insert: 70
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 1
Enter value to insert: 110
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 1
Enter value to insert: 80
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 1
Enter value to insert: 30
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 1
Enter value to insert: 20
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice:


BST.c > ...
59 int main() {
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
4. Post-order traversal
5. Exit
Enter your choice:
1
Enter value to insert: 102
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 105
Invalid choice!
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 108
Invalid choice!
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 1
Enter value to insert: 5
--- Binary Search Tree Menu ---
1. Insert element


EXPLORER
OPEN EDITORS
sllOperation.c
sllStackQueue.c
doubleLinkList.c
BST.c
DSA
.vscode
build\Debug
output
BST.c
BST.exe
cicularQueue.c
cicularQueue.exe
circularQueue.pdf
doubleLinkList.c
doubleLinkList.exe
Infix to postfix.pdf
infixToPostfix.c
infixToPostfix.exe
Leetcode(109).pdf
leetcode(1669).pdf
leetCode203.pdf
leetCode876.pdf
queue operations.pdf
queue.c
queue.exe
sllDeletion.c
sllDeletion.exe
sllInsertion.c
sllInsertion.exe
sllOperation.c
sllOperation.exe
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 2
In-order traversal: 5 20 30 70 80 100 102 110
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 3
Pre-order traversal: 100 70 30 20 5 80 110 102
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 4
Post-order traversal: 5 20 30 80 70 102 110 100
--- Binary Search Tree Menu ---
1. Insert element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. Exit
Enter your choice: 5
PS C:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa>