



# DSA LAB

## Assignment 7

**Name:** Madhav Jha

**Roll no.:** E3-48

**Branch:** CSE (AI & ML)

# Binary Search Tree

## Statement:

Implement a Binary search tree with operations – create, search, insert, inorder, preorder and postorder. Write a menu driven program that performs the above operations also counts the total nodes and total leaf nodes in the tree. int count(T) – returns the total number of nodes from BST int countLeaf(T) – returns the total number of leaf nodes from BST

## Code:

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

struct TreeNode {
    int val;
    struct TreeNode* left;
    struct TreeNode* right;
};

void addNode(struct TreeNode** root, int num) {
    struct TreeNode* node;
    node = (struct TreeNode*)malloc(sizeof(struct TreeNode));
    node->val = num;
    node->left = NULL;
    node->right = NULL;

    if (*root == NULL) {
        *root = node;
        return;
    }

    struct TreeNode* temp = *root;
    while (temp != NULL) {
        if (temp->val == num) {
            break;
        }
        if (temp->val < num) {
            if (temp->right != NULL) {
                temp = temp->right;
            }
        }
    }
}
```

```

        }
        else {
            temp->right = node;
            break;
        }
    }
    else {
        if (temp->left != NULL) {
            temp = temp->left;
        }
        else {
            temp->left = node;
            break;
        }
    }
}

void preOrder(struct TreeNode* node) {
    if (node == NULL) return;
    printf("%d ", node->val);
    preOrder(node->left);
    preOrder(node->right);
}

void inOrder(struct TreeNode* node) {
    if (node == NULL) return;
    inOrder(node->left);
    printf("%d ", node->val);
    inOrder(node->right);
}

void postOrder(struct TreeNode* node) {
    if (node == NULL) return;
    postOrder(node->left);
    postOrder(node->right);
    printf("%d ", node->val);
}

int searchBST(struct TreeNode** root, int num) {

```

```

    struct TreeNode* temp = *root;
    while (temp != NULL) {
        if (temp->val == num) {
            return 1;
        }
        else if (temp->val < num) {
            temp = temp->right;
        }
        else {
            temp = temp->left;
        }
    }
    return 0;
}

void countNodes(struct TreeNode* node, int* num) {
    if (node == NULL) return;
    *num = *num + 1;
    countNodes(node->left, num);
    countNodes(node->right, num);
}

int countAllNodes(struct TreeNode** root) {
    int count = 0;
    countNodes(*root, &count);
    return count;
}

void countLeaf(struct TreeNode* node, int* num) {
    if (node == NULL) return;
    if (node->left == NULL && node->right == NULL) {
        *num = *num + 1;
        return;
    }
    countLeaf(node->left, num);
    countLeaf(node->right, num);
}

int countAllLeaf(struct TreeNode** root) {
    int count = 0;

```

```

    countLeaf(*root, &count);
    return count;
}

int main(void) {

    struct TreeNode** root = (struct TreeNode**)malloc(sizeof(struct
TreeNode*));
    *root = NULL;

    int choice = 0;
    do {
        printf("\n0. Enter 0 to exit!");
        printf("\n1. Add node to tree.");
        printf("\n2. Show Pre-order traversal.");
        printf("\n3. Show In-order traversal.");
        printf("\n4. Show Post-order traversal.");
        printf("\n5. Count all nodes.");
        printf("\n6. Count leafs");
        printf("\n7. Search number in BST");

        printf("\nYour choice: ");
        scanf("%d", &choice);

        int inp, src;
        switch (choice) {
            case 0:
                printf("\n\nExit...!\n\n");
                break;
            case 1:
                printf("\n\nEnter number to add: ");
                scanf("%d", &inp);
                addNode(root, inp);
                printf("\nNode added!!\n");
                break;
            case 2:
                printf("\n\nPre-order: ");
                preOrder(*root);
                printf("\n\n");
                break;

```

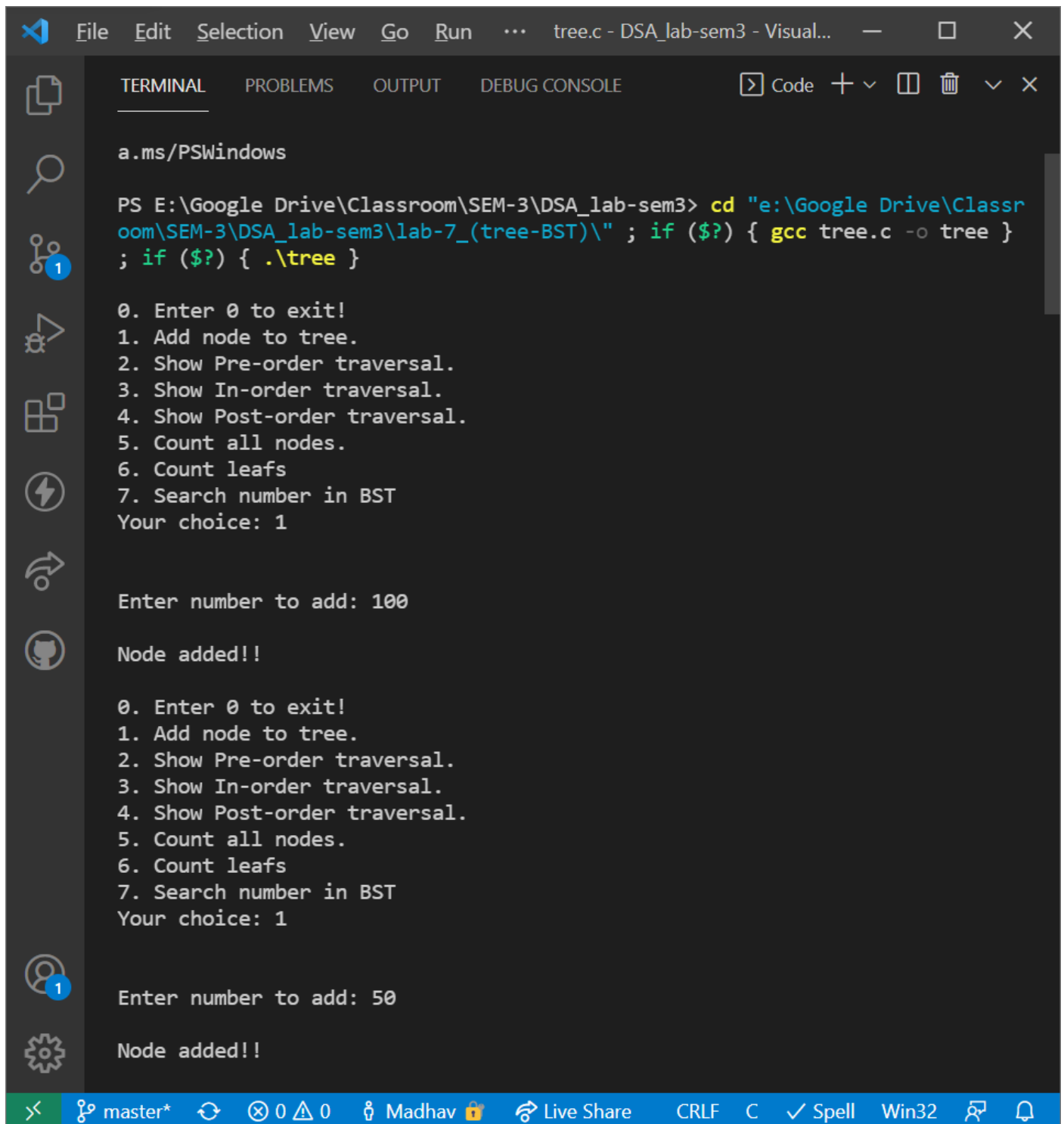
```

        case 3:
            printf("\n\nIN-order: ");
            inOrder(*root);
            printf("\n\n");
            break;
        case 4:
            printf("\n\nPost-order: ");
            postOrder(*root);
            printf("\n\n");
            break;
        case 5:
            printf("\n\nAll nodes count: %d\n", countAllNodes(root));
            break;
        case 6:
            printf("\n\nAll leaf nodes count: %d\n",
countAllLeaf(root));
            break;
        case 7:
            printf("\n\nEnter number to search: ");
            scanf("%d", &inp);
            if (searchBST(root, inp) == 0) {
                printf("\nNumber doesn't exist!!");
            }
            else {
                printf("\nNumber does exists!!");
            }
            printf("\n\n");
            break;
        default:
            printf("\nERROR: Invalid choice!!!");
            break;
    }
} while (choice != 0);

return 0;
}

```

## Output:



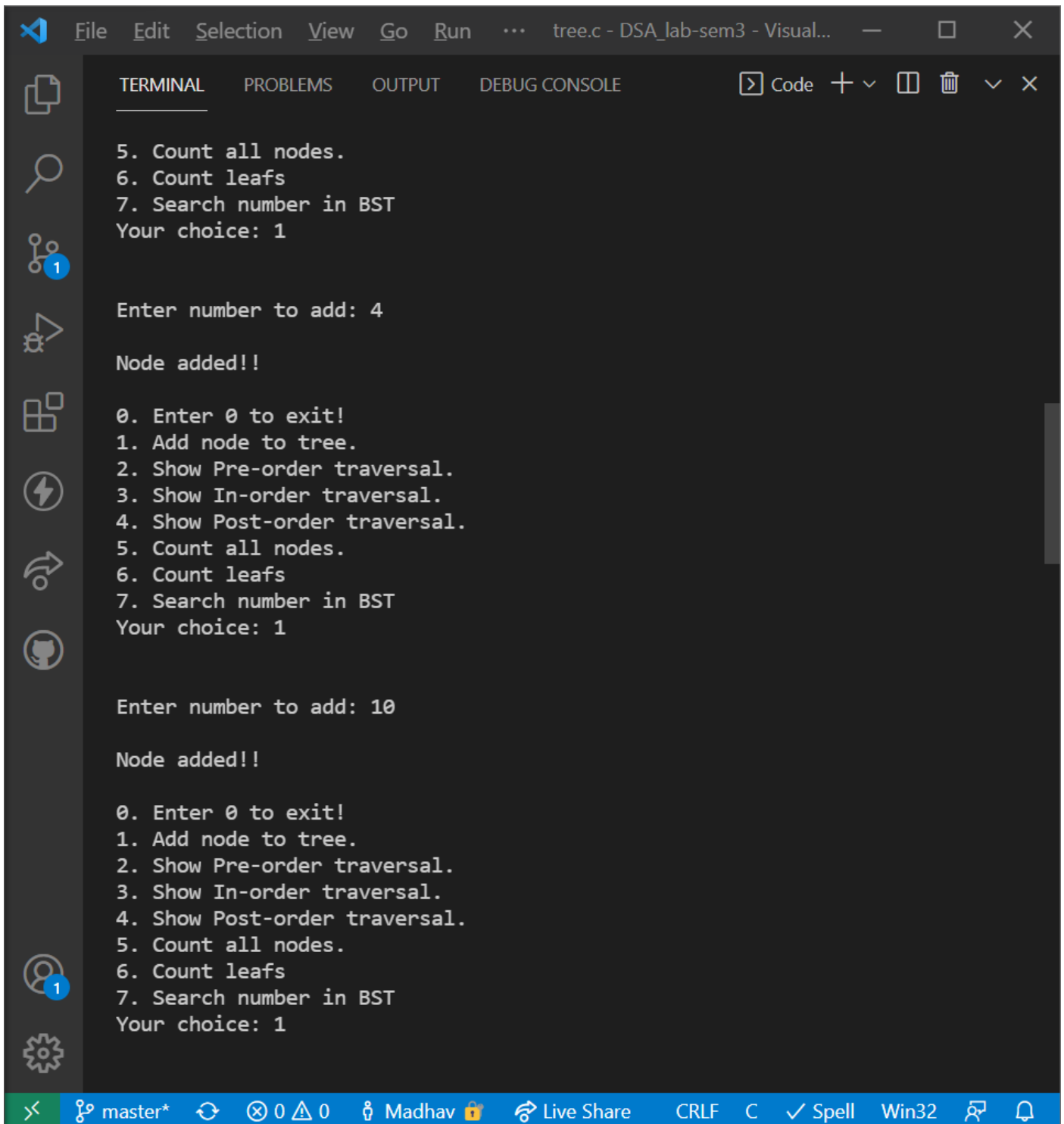
```
tree.c - DSA_lab-sem3 - Visual...  
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE  
a.ms/PSWindows  
PS E:\Google Drive\Classroom\SEM-3\DSA_lab-sem3> cd "e:\Google Drive\Classr  
oom\SEM-3\DSA_lab-sem3\lab-7_(tree-BST)\\" ; if ($?) { gcc tree.c -o tree }  
; if ($?) { .\tree }  
  
0. Enter 0 to exit!  
1. Add node to tree.  
2. Show Pre-order traversal.  
3. Show In-order traversal.  
4. Show Post-order traversal.  
5. Count all nodes.  
6. Count leafs  
7. Search number in BST  
Your choice: 1  
  
Enter number to add: 100  
  
Node added!!  
  
0. Enter 0 to exit!  
1. Add node to tree.  
2. Show Pre-order traversal.  
3. Show In-order traversal.  
4. Show Post-order traversal.  
5. Count all nodes.  
6. Count leafs  
7. Search number in BST  
Your choice: 1  
  
Enter number to add: 50  
  
Node added!!
```

master\* 0 0 Madhav Live Share CRLF C ✓ Spell Win32

The screenshot shows a Visual Studio Code window with a terminal open. The terminal displays a menu for a Binary Search Tree (BST) application. The menu options are: 0. Enter 0 to exit!, 1. Add node to tree., 2. Show Pre-order traversal., 3. Show In-order traversal., 4. Show Post-order traversal., 5. Count all nodes., 6. Count leafs, 7. Search number in BST. The user has entered '1' to add a node. The terminal then prompts 'Enter number to add: 200' and 'Node added!!'. The menu is shown again, and the user enters '1' again. The terminal prompts 'Enter number to add: 4' and 'Node added!!'. The menu is shown a third time, and the user enters '4' to show the post-order traversal.

```
tree.c - DSA_lab-sem3 - Visual...  
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE  
0. Enter 0 to exit!  
1. Add node to tree.  
2. Show Pre-order traversal.  
3. Show In-order traversal.  
4. Show Post-order traversal.  
5. Count all nodes.  
6. Count leafs  
7. Search number in BST  
Your choice: 1  
  
Enter number to add: 200  
  
Node added!!  
  
0. Enter 0 to exit!  
1. Add node to tree.  
2. Show Pre-order traversal.  
3. Show In-order traversal.  
4. Show Post-order traversal.  
5. Count all nodes.  
6. Count leafs  
7. Search number in BST  
Your choice: 1  
  
Enter number to add: 4  
  
Node added!!  
  
0. Enter 0 to exit!  
1. Add node to tree.  
2. Show Pre-order traversal.  
3. Show In-order traversal.  
4. Show Post-order traversal.
```





The image shows a Visual Studio Code window with a terminal open. The terminal displays a menu-driven program for a Binary Search Tree (BST). The menu options are: 0. Enter 0 to exit!, 1. Add node to tree., 2. Show Pre-order traversal., 3. Show In-order traversal., 4. Show Post-order traversal., 5. Count all nodes., 6. Count leafs, 7. Search number in BST. The user has chosen option 1 and entered 4 and 10 as numbers to add, with the program confirming "Node added!!" for each. The terminal output is as follows:

```
5. Count all nodes.
6. Count leafs
7. Search number in BST
Your choice: 1

Enter number to add: 4

Node added!!

0. Enter 0 to exit!
1. Add node to tree.
2. Show Pre-order traversal.
3. Show In-order traversal.
4. Show Post-order traversal.
5. Count all nodes.
6. Count leafs
7. Search number in BST
Your choice: 1

Enter number to add: 10

Node added!!

0. Enter 0 to exit!
1. Add node to tree.
2. Show Pre-order traversal.
3. Show In-order traversal.
4. Show Post-order traversal.
5. Count all nodes.
6. Count leafs
7. Search number in BST
Your choice: 1
```

The Visual Studio Code interface includes a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Docker, GitHub, and Settings. The top bar shows the file name "tree.c - DSA\_lab-sem3 - Visual..." and standard window controls. The bottom status bar displays "master\*", "0 0", "Madhav", "Live Share", "CRLF", "C", "✓ Spell", "Win32", and a notification bell.

The screenshot shows a Visual Studio Code window with a terminal running a C program for a Binary Search Tree (BST). The terminal output is as follows:

```
Enter number to add: 30

Node added!!

0. Enter 0 to exit!
1. Add node to tree.
2. Show Pre-order traversal.
3. Show In-order traversal.
4. Show Post-order traversal.
5. Count all nodes.
6. Count leafs
7. Search number in BST
Your choice: 2

Pre-order: 100 50 4 10 30 200

0. Enter 0 to exit!
1. Add node to tree.
2. Show Pre-order traversal.
3. Show In-order traversal.
4. Show Post-order traversal.
5. Count all nodes.
6. Count leafs
7. Search number in BST
Your choice: 3

IN-order: 4 10 30 50 100 200

0. Enter 0 to exit!
1. Add node to tree.
2. Show Pre-order traversal.
3. Show In-order traversal.
```

The interface includes a sidebar with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Testing, and Settings. The top menu bar shows File, Edit, Selection, View, Go, Run, and a dropdown menu. The bottom status bar displays the current branch (master\*), file encoding (UTF-8), line and column counts (0/0), the user's name (Madhav), and various tool icons like Live Share, CRLF, C, Spell, Win32, and a notification bell.

```
File Edit Selection View Go Run ... tree.c - DSA_lab-sem3 - Visual...
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE
4. Show Post-order traversal.
5. Count all nodes.
6. Count leafs
7. Search number in BST
Your choice: 4

Post-order: 30 10 4 50 200 100

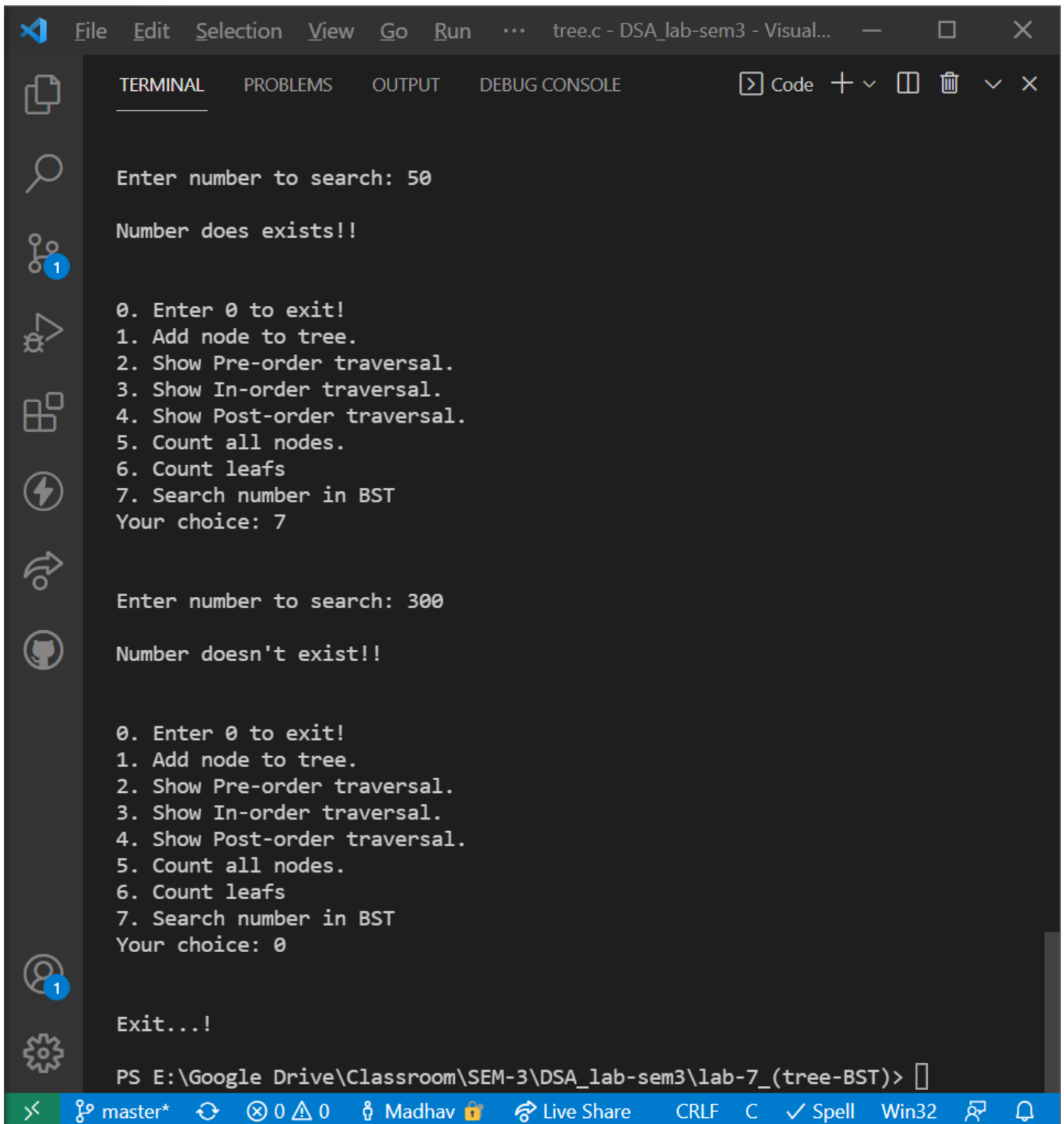
0. Enter 0 to exit!
1. Add node to tree.
2. Show Pre-order traversal.
3. Show In-order traversal.
4. Show Post-order traversal.
5. Count all nodes.
6. Count leafs
7. Search number in BST
Your choice: 5

All nodes count: 6

0. Enter 0 to exit!
1. Add node to tree.
2. Show Pre-order traversal.
3. Show In-order traversal.
4. Show Post-order traversal.
5. Count all nodes.
6. Count leafs
7. Search number in BST
Your choice: 6

All leaf nodes count: 2
```

[illegible]



The screenshot shows a Visual Studio Code window with a terminal open. The terminal has tabs for TERMINAL, PROBLEMS, OUTPUT, and DEBUG CONSOLE. The terminal output shows a program for a Binary Search Tree (BST) with the following menu:

- 0. Enter 0 to exit!
- 1. Add node to tree.
- 2. Show Pre-order traversal.
- 3. Show In-order traversal.
- 4. Show Post-order traversal.
- 5. Count all nodes.
- 6. Count leafs
- 7. Search number in BST

The user enters '7' as their choice. The program prompts 'Enter number to search: 50' and then 'Number does exists!!'. The user enters '300' and the program outputs 'Number doesn't exist!!'. The user enters '0' and the program outputs 'Exit...!'. The terminal prompt is 'PS E:\Google Drive\Classroom\SEM-3\DSA\_lab-sem3\lab-7\_(tree-BST)>'. The status bar at the bottom shows 'master\*', '0 0', 'Madhav', 'Live Share', 'CRLF', 'C', 'Spell', 'Win32', and a bell icon.

```
File Edit Selection View Go Run ... tree.c - DSA_lab-sem3 - Visual...  
TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE Code + - [ ] [ ] [ ] [ ] [ ] [ ]  
Enter number to search: 50  
Number does exists!!  
0. Enter 0 to exit!  
1. Add node to tree.  
2. Show Pre-order traversal.  
3. Show In-order traversal.  
4. Show Post-order traversal.  
5. Count all nodes.  
6. Count leafs  
7. Search number in BST  
Your choice: 7  
Enter number to search: 300  
Number doesn't exist!!  
0. Enter 0 to exit!  
1. Add node to tree.  
2. Show Pre-order traversal.  
3. Show In-order traversal.  
4. Show Post-order traversal.  
5. Count all nodes.  
6. Count leafs  
7. Search number in BST  
Your choice: 0  
Exit...!  
PS E:\Google Drive\Classroom\SEM-3\DSA_lab-sem3\lab-7_(tree-BST)> [ ]  
master* 0 0 Madhav Live Share CRLF C Spell Win32
```