Institute of Computer Technology
B. Tech Computer Science and Engineering
Subject: DS (2CSE302)

PRACTICAL-19

AIM: - Implement the real-life scenario using binary search tree.

19. McAfee Corp. is an American global computer security software company headquartered in San Jose, CA. Kiara is working at McAfee, Ahmedabad and she wants to capture all orders of security software equipment in the form of a binary search tree as each order is linked to the previous one in a parent-child relationship. Kindly perform the below operation using C language also create the binary search tree structure using paper and pen:

a. Create BST from the below list of elements

int array [7] = {20, 15, 30, 10, 19, 21, 42}

b. Print elements using pre-order, post-order, and in-order traversal.

Input:

20, 15, 30, 10, 19, 21, 42

Output:

Pre-order Traversal: 20, 15, 10, 19, 30, 21, 42
Post-order Traversal: 10, 19, 15, 21, 42, 30, 20
In-order Traversal: 10, 15, 19, 20, 21, 30, 42

c. Search elements 31 and 15 in the created BST.

Output:

Enter Elements to search: 15
Element exists
Enter Elements to search: 31
Element does not exist

SOLUTION

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
struct yash
{
    int data;
    struct yash *left;
    struct yash *right;
};
struct yash *root = NULL;
void insert(int item)
```

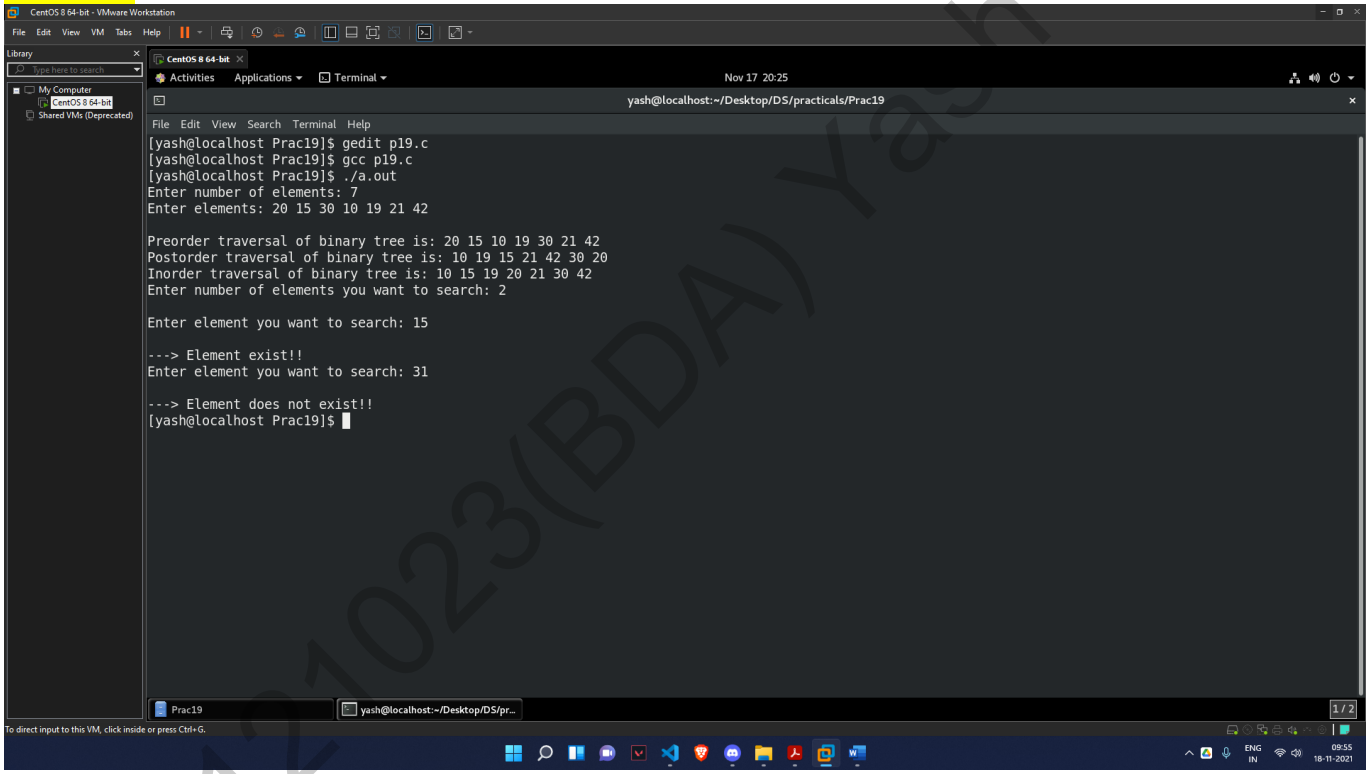
```
{
        struct yash *new_node, *parent, *current;
        new_node = (struct yash *)malloc(sizeof(struct yash));
        new_node->data = item;
        new_node->left = NULL;
        new_node->right = NULL;
        parent = root;
        if (root == NULL)
        {
                root = new_node;
        }
        else
        {
                current = root;
                while (current != NULL)
                {
                        parent = current;
                        if (new_node->data >= current->data)
                        {
                                current = current->right;
                        }
                        else
                        {
                                current = current->left;
                        }
                }
                if (new_node->data >= parent->data)
                {
                        parent->right = new_node;
                }
                else
                {
                        parent->left = new_node;
                }
        }
}
void preorder(struct yash *root)
{
```

```
        if (root != NULL)
        {
                printf("%d ", root->data);
                preorder(root->left);
                preorder(root->right);
        }
}
void postorder(struct yash *root)
{
        if (root != NULL)
        {
                postorder(root->left);
                postorder(root->right);
                printf("%d ", root->data);
        }
}
void inorder(struct yash *root)
{
        if (root != NULL)
        {
                inorder(root->left);
                printf("%d ", root->data);
                inorder(root->right);
        }
}
void search(int element)
{
        struct yash *current;
        current = root;
        bool flag = false;
        while (current != NULL && flag == false)
        {
                if (current->data == element)
                {
                        printf("\n---> Element exist!!");
                        flag = true;
                }
                else
```

```
            {
                if (element >= current->data)
                {
                    current = current->right;
                }
                else
                {
                    current = current->left;
                }
            }
        }
        if (current == NULL)
        {
            printf("\n---> Element does not exist!!");
        }
}
int main()
{
        int i, n, n1;
        printf("Enter number of elements: ");
        scanf("%d", &n);
        int arr[n];
        printf("Enter elements: ");
        for (i = 0; i < n; i++)
        {
            scanf("%d", &arr[i]);
            insert(arr[i]);
        }
        printf("\nPreorder traversal of binary tree is: ");
        preorder(root);

        printf("\nPostorder traversal of binary tree is: ");
        postorder(root);

        printf("\nInorder traversal of binary tree is: ");
        inorder(root);

        printf("\nEnter number of elements you want to search: ");
```

```
        scanf("%d", &n1);
        int a[n1];
        for (i = 0; i < n1; i++)
        {
                printf("\nEnter element you want to search: ");
                scanf("%d", &a[i]);
                search(a[i]);
        }
        printf("\n");
        return 0;
}
```

OUTPUT



USING PEN AND PAPER