**Code:-**

# **Name : Rajkumar B L**

# **Reg.No : 2047120**

# **Course : MCS 271 Data Structure (Lab 12 – BST)**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 \* Name : Rajkumar B L

 \* Reg  : 2047120

 \* Lab  : 12

 \* Program : BST

 \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

struct node

{

    int key;

    struct node \*left, \*right;

};

struct node \*newNode(int item)

{

    struct node \*temp = (struct node \*)malloc(sizeof(struct node));

    temp->key = item;

    temp->left = temp->right = NULL;

    return temp;

}

void traversetree(struct node \*root)

{

    if (root != NULL)

    {

        traversetree(root->left);

        printf("%d \t", root->key);

        traversetree(root->right);

    }

}

struct node \*search(struct node \*root, int key)

{

    if (root == NULL || root->key == key)

        return root;

    if (root->key < key)

        return search(root->right, key);

    return search(root->left, key);

}

struct node \*insert(struct node \*node, int key)

{

    if (node == NULL)

        return newNode(key);

    if (key < node->key)

        node->left = insert(node->left, key);

    else if (key > node->key)

        node->right = insert(node->right, key);

    return node;

}

int main()

{

    int ch, i, num;

    struct node \*root = NULL;

    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\*  Name : Rajkumar B L     \*\n\*  Reg  : 2047120          \*\n\*  Lab  : 12               \*\n\*  Prg  : BST              \*\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

    do

    {

        printf("\n=========================\n\tMenu\n=========================\n");

        printf("1. Insert a node. \n");

        printf("2. Search for a node. \n");

        printf("3. Traverse the BST. \n");

        printf("4. Exit \n");

        printf("=========================\n");

        printf("Enter your choice: ");

        scanf("%d", &ch);

        switch (ch)

        {

        case 1:

            printf("Enter the number to be inserted: ");

            scanf("%d", &num);

            if (root == NULL){

                root = insert(root, num);

                printf("%d inserted successfully!\n",num);

            }

            else{

                insert(root, num);

                printf("%d inserted successfully!\n", num);

            }

            break;

        case 3:

            printf("The tree is :\n");

            traversetree(root);

            printf("\n");

            break;

        case 2:

            printf("Enter the number to be searched: ");

            scanf("%d", &num);

            printf("Searching for %d in the tree ", num);

            if (search(root, num))

                printf("\nElement found!\n");

            else

                printf("\nElement not found!\n");

            break;

        case 4:

            printf("Bye!\n\n");

            exit(0);

        default:

            printf("Invalid Input!\n");

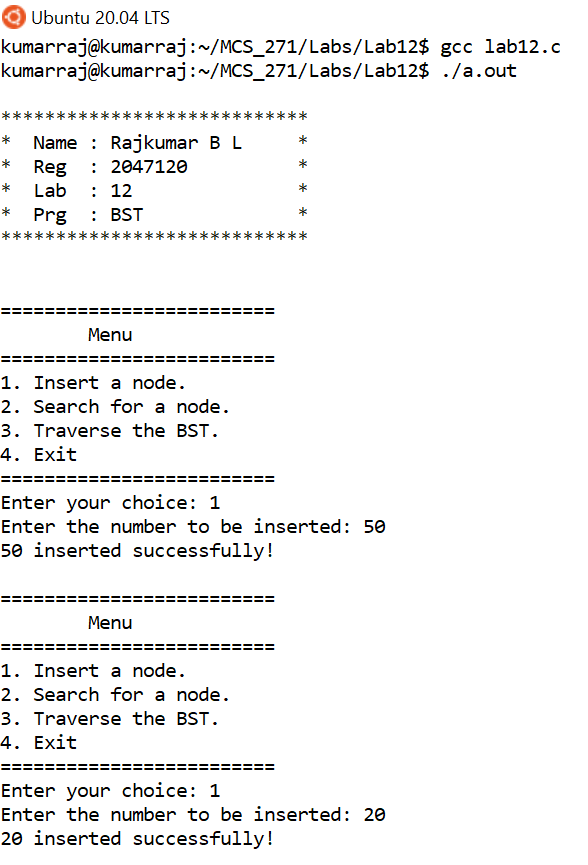
        }

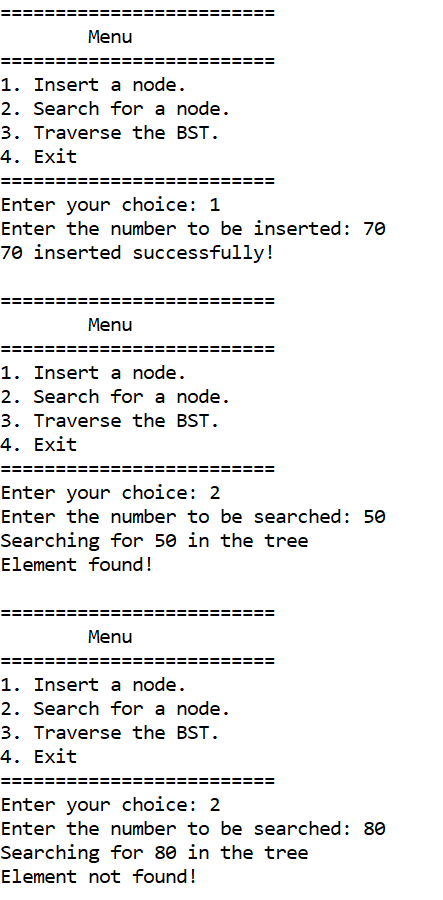
    } while (ch != 4);

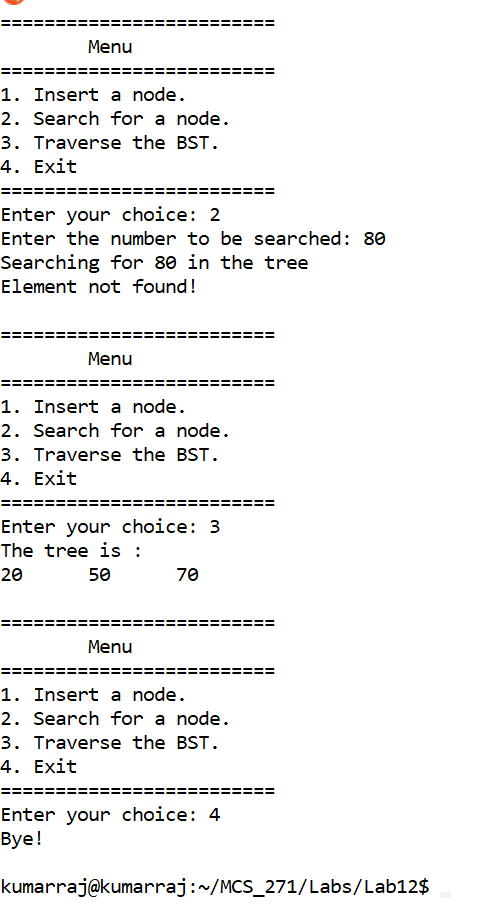
    return 0;

}

**Output:**

****

****

****