Jawahar Education Society's
A. C. Patil College of Engineering, Kharghar
Navi Mumbai 410210

Student Name: Chetan Ingale

Course Name: C.S.E. (IoT CS BC)

Year: S.E.

Roll No.: 17

PRN No.: 221111030

Course code: CSL301

Semester: 3

Experiment Evaluation Sheet

Experiment No.: 8

Experiment Name:
Write a program to implement Binary Search Tree ADT using Linked List.

Sr No.	Evaluation Criteria	Marks (Out of 9)	Performance Date	Correction Date and Signature of Instructor
1	Experiment Performance			
2	Journal Performance			
3	Punctuality			
Total				

Code :

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

typedef struct Node {
    int data;
    struct Node *left;
    struct Node *right;
} Node;

Node* createNode(int item) {
    Node* temp = (Node*)malloc(sizeof(Node));
    temp->data = item;
    temp->left = temp->right = NULL;
    return temp;
}

Node* insert(Node* node, int key) {
    if (node == NULL) return createNode(key);

    if (key < node->data)
        node->left = insert(node->left, key);
    else if (key > node->data)
        node->right = insert(node->right, key);

    return node;
}

Node* minValueNode(Node* node) {
    Node* current = node;
    while (current && current->left != NULL)
        current = current->left;
    return current;
}

Node* deleteNode(Node* root, int key) {
    if (root == NULL) return root;

    if (key < root->data)
        root->left = deleteNode(root->left, key);
    else if (key > root->data)
        root->right = deleteNode(root->right, key);
    else {
        if (root->left == NULL) {
            Node* temp = root->right;
            free(root);
            return temp;
        } else if (root->right == NULL) {
            Node* temp = root->left;
            free(root);
            return temp;
        }

        Node* temp = minValueNode(root->right);
```

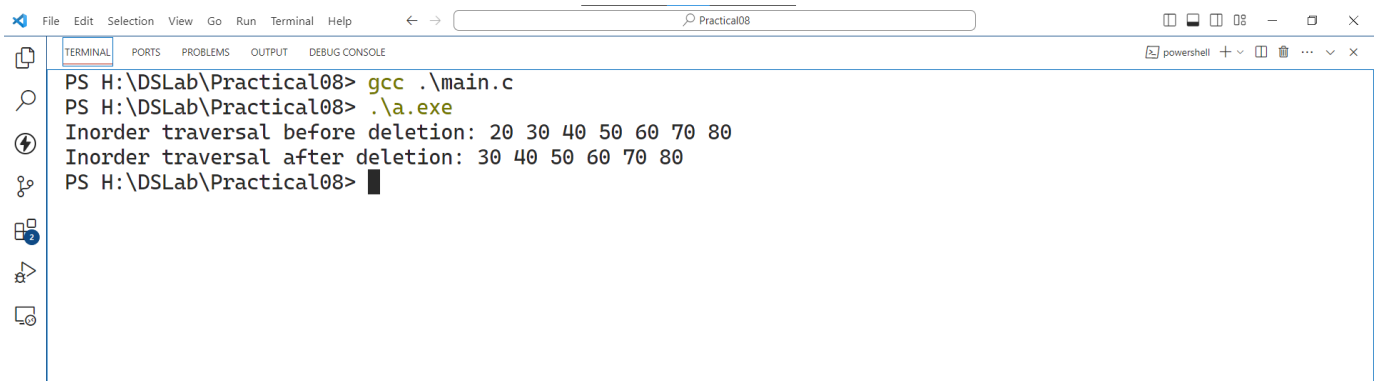
Code :

```
        root->data = temp->data;
        root->right = deleteNode(root->right, temp->data);
    }
    return root;
}

void inorder(Node* root) {
    if (root != NULL) {
        inorder(root->left);
        printf("%d ", root->data);
        inorder(root->right);
    }
}

int main() {
    Node* root = NULL;
    root = insert(root, 50);
    insert(root, 30);
    insert(root, 20);
    insert(root, 40);
    insert(root, 70);
    insert(root, 60);
    insert(root, 80);
    printf("Inorder traversal before deletion: ");
    inorder(root);
    root = deleteNode(root, 20);
    printf("\nInorder traversal after deletion: ");
    inorder(root);
    return 0;
}
```

Output :



Conclusion :

Through this experiment we have learnt about how to implement a Binary Search Tree using the C language. Various operations like insertion, deletion and traversal are applied on the BST.
This experiment helps us in using BST as a data structure for further reference.