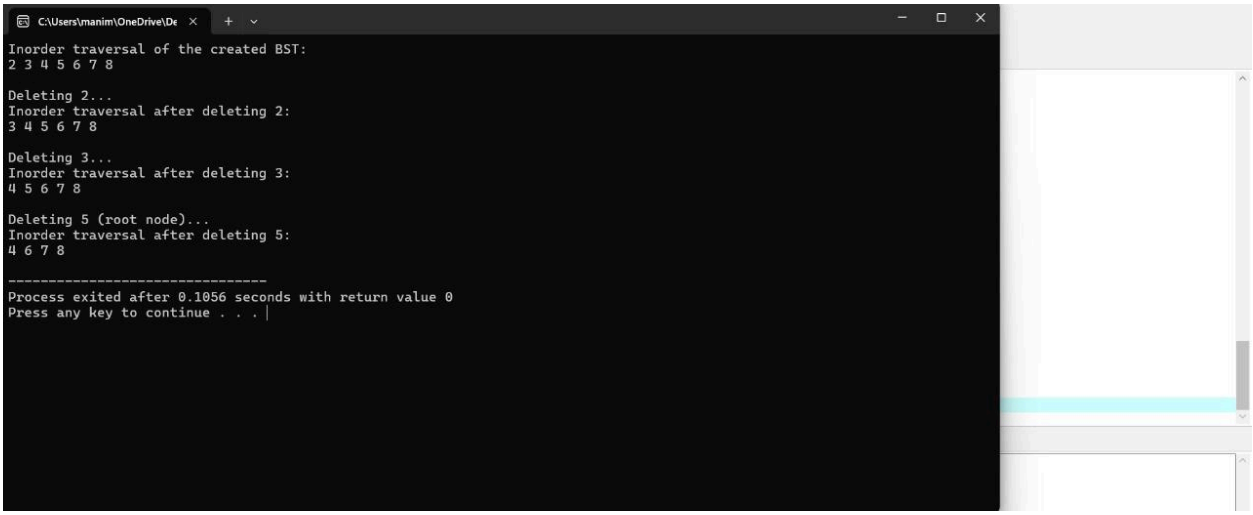
```
#include <stdio.h>
#include <stdlib.h>

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

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

struct Node* insert(struct Node* root, int data) {
    if (root == NULL) {
        return createNode(data);
    }

    if (data < root->data) {
        root->left = insert(root->left, data);
    } else if (data > root->data) {
        root->right = insert(root->right, data);
    }

    return root;
}

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

struct Node* deleteNode(struct Node* root, int data) {
    if (root == NULL) {
        return root;
    }

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

        root->data = temp->data;

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

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

int main() {
    struct Node* root = NULL;
    root = insert(root, 5);
    insert(root, 3);
    insert(root, 2);
    insert(root, 4);
    insert(root, 7);
    insert(root, 6);
    insert(root, 8);

    printf("Inorder traversal of the created BST: \n");
    inorderTraversal(root);
    printf("\n");
```

```c
    printf("\nDeleting 2...\n");
    root = deleteNode(root, 2);
    printf("Inorder traversal after deleting 2: \n");
    inorderTraversal(root);
    printf("\n");

    printf("\nDeleting 3...\n");
    root = deleteNode(root, 3);
    printf("Inorder traversal after deleting 3: \n");
    inorderTraversal(root);
    printf("\n");

    printf("\nDeleting 5 (root node)...\n");
    root = deleteNode(root, 5);
    printf("Inorder traversal after deleting 5: \n");
    inorderTraversal(root);
    printf("\n");

    return 0;
}
```



```
Inorder traversal of the created BST:
2 3 4 5 6 7 8

Deleting 2...
Inorder traversal after deleting 2:
3 4 5 6 7 8

Deleting 3...
Inorder traversal after deleting 3:
4 5 6 7 8

Deleting 5 (root node)...
Inorder traversal after deleting 5:
4 6 7 8

--------------------------------
Process exited after 0.1056 seconds with return value 0
Press any key to continue . . .
```