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

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

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

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

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

    return root;
}

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

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

    if (key < root->key)
        root->left = deleteNode(root->left, key);
    else if (key > root->key)
        root->right = deleteNode(root->right, key);
    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 = minValueNode(root->right);
        root->key = temp->key;
        root->right = deleteNode(root->right, temp->key);
    }

    return root;
}

struct Node* search(struct Node* root, int key) {
    if (root == NULL || root->key == key)
        return root;

    if (key < root->key)
        return search(root->left, key);
    else
        return search(root->right, key);
}

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

int main() {
    struct Node* root = NULL;
    int choice, key;

    do {
        printf("\n1. Insert element\n");
        printf("2. Delete element\n");
        printf("3. Search for element\n");
        printf("4. Display BST (Inorder traversal)\n");
        printf("0. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter key to insert: ");
                scanf("%d", &key);
                root = insert(root, key);
                break;
            case 2:
                printf("Enter key to delete: ");
                scanf("%d", &key);
                root = deleteNode(root, key);
                break;
            case 3:
                printf("Enter key to search: ");
                scanf("%d", &key);
                struct Node* searchResult = search(root, key);
                if (searchResult != NULL)
                    printf("Key %d found in the BST.\n", key);
                else
                    printf("Key %d not found in the BST.\n", key);
                break;
            case 4:
                printf("Inorder traversal of BST: ");
                inorderTraversal(root);
                printf("\n");
                break;
            case 0:
                printf("Exiting the program.\n");
                break;
            default:
                printf("Invalid choice. Please enter a valid option.\n");
        }
    } while (choice != 0);

    return 0;
}
```