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

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

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

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

    return root;
}

int findMin(struct Node* root) {
    if (root == NULL) {
        printf("BST is empty\n");
        return -1;
    }

    while (root->left != NULL)
        root = root->left;

    return root->data;
}

int findMax(struct Node* root) {
    if (root == NULL) {
```

```
        printf("BST is empty\n");
        return -1;
    }

    while (root->right != NULL)
        root = root->right;

    return root->data;
}

int main() {
    struct 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("Minimum value in BST: %d\n", findMin(root));
    printf("Maximum value in BST: %d\n", findMax(root));

    return 0;
}
```