

```

#include <stdio.h>
#include <stdlib.h>
// Structure for a node in BST
struct node {
    int data;
    struct node* left;
    struct node* right;
};
// Function to create a new node
struct node* newNode(int item) {
    struct node* temp = (struct node*)malloc(sizeof(struct node));
    temp->data = item;
    temp->left = temp->right = NULL;
    return temp;
}
// Function to insert a new node in BST
struct node* insert(struct node* node, int data) {
    if (node == NULL)
        return newNode(data);
    if (data < node->data)
        node->left = insert(node->left, data);
    else if (data > node->data)
        node->right = insert(node->right, data);
    return node;
}
// Function to find the minimum element in BST
struct node* minValue(struct node* node) {
    struct node* current = node;
    while (current && current->left != NULL)
        current = current->left;
    return current;
}
// Function to find the maximum element in BST
struct node* maxValue(struct node* node) {
    struct node* current = node;
    while (current && current->right != NULL)
        current = current->right;
    return current;
}
// Driver program to test above functions
int main() {
    struct node* root = NULL;
    int num, n;
    printf("Enter the number of elements to insert: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for (int i = 0; i < n; i++) {

```

```
    scanf("%d", &num);  
    root = insert(root, num);  
}  
printf("Minimum element in BST: %d\n", minValue(root)->data);  
printf("Maximum element in BST: %d\n", maxValue(root)->data);  
return 0;  
}
```