#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 = NULL;

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\* findMin(struct Node\* root) {

struct Node\* current = root;

while (current && current->left != NULL) {

current = current->left;

}

return current;

}

struct Node\* findMax(struct Node\* root) {

struct Node\* current = root;

while (current && current->right != NULL) {

current = current->right;

}

return current;

}

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 inorder(struct Node\* root) {

if (root != NULL) {

inorder(root->left);

printf("%d ", root->key);

inorder(root->right);

}

}

int main() {

struct Node\* root = NULL;

int n, key, choice, searchKey;

printf("Enter the number of keys to insert: ");

scanf("%d", &n);

printf("Enter %d keys: ", n);

for (int i = 0; i < n; i++) {

scanf("%d", &key);

root = insert(root, key);

}

do {

printf("\nMenu:\n");

printf("1. Find Minimum Key\n");

printf("2. Find Maximum Key\n");

printf("3. Search for a Key\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

if (root) {

struct Node\* minNode = findMin(root);

printf("Minimum Key: %d\n", minNode->key);

} else {

printf("The tree is empty.\n");

}

break;

case 2:

if (root) {

struct Node\* maxNode = findMax(root);

printf("Maximum Key: %d\n", maxNode->key);

} else {

printf("The tree is empty.\n");

}

break;

case 3:

printf("Enter the key to search: ");

scanf("%d", &searchKey);

struct Node\* foundNode = search(root, searchKey);

if (foundNode) {

printf("Key %d found in the BST.\n", searchKey);

} else {

printf("Key %d not found in the BST.\n", searchKey);

}

break;

case 4:

printf("Exiting program.\n");

break;

default:

printf("Invalid choice! Please try again.\n");

}

} while (choice != 4);

return 0;

}