```c
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node *left, *right;
} Node;
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = data;
    newNode->left = newNode->right = NULL;
    return newNode;
}
Node* insert(Node* root, int data) {
    if (root == NULL)
        return createNode(data);
    if (data < root->data)
        root->left = insert(root->left, data);
    else
        root->right = insert(root->right, data);
    return root;
}
void kthMinUtil(Node* root, int k, int* count, int* result) {
    if (root == NULL || *count >= k)
        return;
    kthMinUtil(root->left, k, count, result);
    (*count)++;
    if (*count == k) {
        *result = root->data;
        return;
    }
    kthMinUtil(root->right, k, count, result);
}
int kthMin(Node* root, int k) {
    int count = 0, result = -1;
    kthMinUtil(root, k, &count, &result);
    return result;
}
int main() {
    Node* root = NULL;
    int elements[] = {20, 8, 22, 4, 12, 10, 14};
    int n = sizeof(elements) / sizeof(elements[0]);
    for (int i = 0; i < n; i++)
        root = insert(root, elements[i]);
    int k = 3;
```

```
    int result = kthMin(root, k);
    if (result != -1)
        printf("The %d-th minimum value in BST is: %d\n", k, result);
    else
        printf("Less than %d nodes in the BST.\n", k);
    return 0;
}
```