

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

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

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

```
struct Node* createNode(int val) {  
    struct Node* node = (struct Node*)malloc(sizeof(struct Node));  
    node->data = val;  
    node->left = node->right = NULL;  
    return node;  
}
```

```
struct Node* insert(struct Node* root, int val) {  
    if (root == NULL)  
        return createNode(val);  
  
    if (val < root->data)  
        root->left = insert(root->left, val);  
    else  
        root->right = insert(root->right, val);
```

```

    return root;
}

void inorder(struct Node* root, int k, int* count, int* result) {

    if (root == NULL || *count >= k) return;

    inorder(root->left, k, count, result);

    (*count)++;

    if (*count == k) {
        *result = root->data;

        return;
    }

    inorder(root->right, k, count, result);
}

int main() {

    struct Node* root = NULL;

    int arr[] = {20, 8, 22, 4, 12};

    int n = sizeof(arr)/sizeof(arr[0]);

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

        root = insert(root, arr[i]);

    int k = 3, count = 0, result = -1;

```

```
inorder(root, k, &count, &result);

if (result != -1)

    printf("%d-th minimum value is %d\n", k, result);

else

    printf("Tree has less than %d nodes\n", k);

return 0;
}
```

Output

```
3-th minimum value is 12
```

```
=== Code Execution Successful ===
```