LAB 8.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management

Projects Files FSymbol

Workspace

Start here | LAB 8.c

```c
    }
}
void displayAll(struct Node* root) {
    if (root == NULL) {
        printf("Tree is empty!\n");
        return;
    }

    struct Node* queue[100];
    int front = 0, rear = 0;

    queue[rear++] = root;

    printf("Elements in the tree: ");
    while (front < rear) {
        struct Node* temp = queue[front++];
        printf("%d ", temp->data);

        if (temp->left)
            queue[rear++] = temp->left;
        if (temp->right)
            queue[rear++] = temp->right;
    }
    printf("\n");
}

int main() {
    struct Node* root = NULL;
    int n, value, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    for (i = 0; i < n; i++) {
        printf("Enter value %d: ", i + 1);
        scanf("%d", &value);
        root = insert(root, value);
    }

    printf("\nInorder Traversal : ");
    inorder(root);

    printf("\nPreorder Traversal : ");
    preorder(root);

    printf("\nPostorder Traversal : ");
    postorder(root);

    printf("\n");
    displayAll(root);

    return 0;
}
```


C:\Users\Admin\Desktop\1BM24CS164\LAB 8.c | C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 7, Col 3, Pos 107 | Insert | Read/Write | default

Type here to search

Humid ENG 12.00.07 PM 01-12-2025

LAB 8.c - Code::Blocks 20.03

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

Management

Projects Files FSymbol

Workspace

Start here | LAB 8.c

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

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

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

    return root;
}
void inorder(struct Node* root) {
    if (root != NULL) {
        inorder(root->left);
        printf("%d ", root->data);
        inorder(root->right);
    }
}
void preorder(struct Node* root) {
    if (root != NULL) {
        printf("%d ", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}
void postorder(struct Node* root) {
    if (root != NULL) {
        postorder(root->left);
        postorder(root->right);
        printf("%d ", root->data);
    }
}
void displayAll(struct Node* root) {
    if (root == NULL) {
        printf("Tree is empty!\n");
        return;
    }

    struct Node* queue[100];
    int front = 0, rear = 0;
```


C:\Users\Admin\Desktop\1BM24CS164\LAB 8.c | C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 7, Col 3, Pos 107 | Insert | Read/Write | default

Type here to search

Humid ENG 11.59.48 AM 01-12-2025