

Giullio Emmanuel da Cruz Di Gerolamo

RA: 790965

Frequência F8

Arquivo ArvoreBB.cpp

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

typedef struct TreeNode {
    int data;
    struct TreeNode* esquerda, *direita;
    int eh_folha;
}TreeNode;

TreeNode* cria(int data) { //cria a arvore
    TreeNode* node = (TreeNode*) calloc (1, sizeof(TreeNode));
    node->data = data;
    node->esquerda = node->direita = NULL;
    node->eh_folha = 1;
    return node;
}

TreeNode* insere(TreeNode* raiz, int data) { //insere valor na arvore
    if (!raiz) {
        raiz = cria(data);
        return raiz;
    }
    else {
        TreeNode* node = cria(data);
        TreeNode* temp = raiz;
        while (temp) {
            if (temp->eh_folha) {
                if (temp->data > data) {
                    temp->esquerda = node;
                    temp->eh_folha = 0;
                    break;
                }
                else {
                    temp->direita = node;
                }
            }
        }
    }
}
```

```

        temp->eh_folha = 0;
        break;
    }
}
else {
    if (temp->data > data) {
        if (temp->esquerda == NULL) {
            temp->esquerda = node;
            break;
        }
        temp = temp->esquerda;
    }
    else {
        if (temp->direita == NULL) {
            temp->direita = node;
            break;
        }
        temp = temp->direita;
    }
}
}
}
return raiz;
}

int esta_na_arvore(TreeNode* raiz, int valor) { //verifica se esta na arvore
    if (!raiz)
        return 0;
    if (raiz->data == valor)
        return 1;
    else if (raiz->data > valor)
        return esta_na_arvore(raiz->esquerda, valor);
    else
        return esta_na_arvore(raiz->direita, valor);
    return 0;
}

void destroi(TreeNode* raiz) { //apaga a arvore
    if (!raiz)
        return;
    destroi(raiz->esquerda);
    destroi(raiz->direita);
    free(raiz);
}

void print_EnA(TreeNode* raiz, int valor) { //printa se esta na arvore
    if (esta_na_arvore(raiz, valor) == 1) {
        printf("Value: %d esta na arvore!\n", valor);
    }
}

```

```

    }
    else {
        printf("Value: %d nao esta na arvore.\n", valor);
    }
}

void imprime_todos(TreeNode* raiz) { //printa a arvore
    if (!raiz)
        return;
    imprime_todos(raiz->esquerda);
    printf("Node: %d -> ", raiz->data);
    imprime_todos(raiz->direita);
}

int soma(TreeNode* raiz) //soma os valores da arvore
{
    if (raiz == NULL)
        return 0;
    return (raiz->data + soma(raiz->esquerda) + soma(raiz->direita));
}

int main() {
    TreeNode* raiz = cria(45);
    raiz = insere(raiz, 20);
    raiz = insere(raiz, 15);
    raiz = insere(raiz, 60);
    raiz = insere(raiz, 40);
    raiz = insere(raiz, 50);
    raiz = insere(raiz, 70);
    imprime_todos(raiz);
    printf("\n");
    int sum = soma(raiz);
    printf("soma: %d ", sum);
    printf("\n");
    print_EnA(raiz, 15);
    print_EnA(raiz, 70);
    print_EnA(raiz, 35);
    destroi(raiz);
    return 0;
}

```

Prints da execução

The image displays two screenshots of a C++ IDE (Dev-C++ 5.11) showing the execution of a program that constructs and traverses a binary search tree.

Top Screenshot: The IDE shows the source code in `main.cpp`. The code defines a `TreeNode` structure and implements functions for inserting nodes and performing a pre-order traversal. The compiler output at the bottom indicates a successful compilation with no errors or warnings. The output filename is `C:\Users\Administrator\Desktop\work\AED1\F8\Project2.exe`, and the compilation time is 0.31s.

Bottom Screenshot: The IDE shows the program's output during execution. The output displays the traversal of the tree nodes, showing the values 15, 28, 40, 45, 50, 60, and 70. The program exits after 0.01364 seconds with a return value of 0. The output also shows the values 15, 28, 40, 45, 50, 60, and 70, indicating the traversal of the tree nodes.