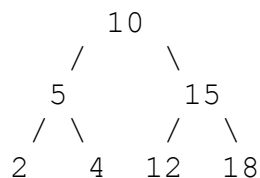


```

/*
    Estrutura simples - Árvore Binária
    MSc. Prof. Rafael S. Bressan
    2019
*/
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

// Definição da estrutura que será utilizada
/*

```



```

*/
struct no {
    int valor;
    struct no *filhoEsquerda;
    struct no *filhoDireita;
};
typedef struct no noh; //typedef permite ao programador definir um
novo nome para um determinado tipo.

// Cria uma árvore
void criarArvore(noh **n){
    (*n) = NULL;
}

void verificaEstado(noh **n){
    if (*n == NULL){
        printf("\n A árvore binária está vazia!");
    }else{
        printf("\n A árvore binária não está vazia!");
    }
}

void inserir (noh **n, int valor) {
    if (*n == NULL) {
        printf("\nEsta Null!");
        *n = (noh *)malloc(sizeof(noh));
        (*n)->filhoDireita = NULL;
        (*n)->filhoEsquerda = NULL;
        (*n)->valor = valor;
    }else if (valor < ((*n)->valor)) {
        inserir(&((*n)->filhoEsquerda), valor);
    }else{
        inserir(&((*n)->filhoDireita), valor);
    }
}

void exibirEmOrdem(noh **n){

```

```

        if(*n != NULL){
            exibirEmOrdem(&((*n)->filhoEsquerda));
            printf("\n%d", (*n)->valor);
            exibirEmOrdem(&((*n)->filhoDireita));
        }
    }

void exibirPreOrdem(noh **n){
    if(*n != NULL){
        printf("\n%d", (*n)->valor);
        exibirPreOrdem(&((*n)->filhoEsquerda));
        exibirPreOrdem(&((*n)->filhoDireita));
    }
}

void exibirPosOrdem(noh **n){
    if(*n != NULL){
        exibirPosOrdem(&((*n)->filhoEsquerda));
        exibirPosOrdem(&((*n)->filhoDireita));
        printf("\n%d", (*n)->valor);
    }
}

main(){
    int op,valor;
    noh p;

    // cria a árvore vazia
    criarArvore(&p);

    // Vamos verificar se está vazia.
    verificaEstado(&p);

    inserir(&p, 10); // raiz
    inserir(&p, 5);
    inserir(&p, 15);
    inserir(&p, 2);
    inserir(&p, 4);
    inserir(&p, 12);
    inserir(&p, 18);

    verificaEstado(&p);

    printf("\n Em ordem! \n");
    exibirEmOrdem(&p);
    printf("\n Pré ordem! \n");
    exibirPreOrdem(&p);
    printf("\n Pós ordem! \n");
    exibirPosOrdem(&p);

    return 0;
}

```