

INF1007 - PROGRAMAÇÃO II

LISTA DE EXERCÍCIOS 13

1. Implemente um TAD *Árvore* para armazenar números inteiros com as seguintes funções:

```
NoArvore *arvore_criaVazia(void);  
NoArvore *arvore_cria(int info, NoArvore *sae, NoArvore *sad);  
int arvore_vazia(NoArvore *a);  
void arvore_libera(NoArvore *a);  
void arvore_imprime(NoArvore *a, int nivel);
```

Além das funções normais, o TAD *Árvore* também deve implementar e exportar as seguintes funções:

- `arvore_folhas` – A função recebe uma árvore binária e retorna o número de folhas dessa árvore (o total de nós que não tem nenhum filho). Essa função tem o seguinte protótipo:

```
int arvore_folhas(NoArvore* a);
```

- `arvore_intermediarios` – A função recebe uma árvore binária e retorna a quantidade de nós que não são folhas (o total de nós que tem pelo menos um filho). Essa função tem o seguinte protótipo:

```
int arvore_intermediarios(NoArvore* a);
```

- `arvore_pares` – A função recebe uma árvore binária e retorna a quantidade de nós que guardam números pares. Essa função tem o seguinte protótipo:

```
int arvore_pares(NoArvore* a);
```

- `arvore_iguais` – A função recebe duas árvores binárias e compara se elas são iguais. A função deve retornar 1 se as árvores são iguais e 0 caso contrário. Árvores iguais devem ter a mesma estrutura hierárquica e armazenar a mesma informação nos nós de mesma posição nas duas árvores. A função tem o seguinte protótipo:

```
int arvore_iguais(NoArvore *a1, NoArvore *a2);
```

- `arvore_alturaMaior` – A função recebe uma árvore binária e um valor inteiro. A função deve verificar se a altura da árvore é maior que o valor inteiro recebido por parâmetro. A função deve retornar 1 se a altura é maior, e 0 caso contrário. A função tem o seguinte protótipo:

```
int arvore_alturaMaior(NoArvore *a, int altura);
```

Após implementar as funções, utilize a seguinte função principal para testar o seu programa:

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

int main()
{
    int n;
    NoArvore *arvore1 = arvore_Cria(1,
                                    arvore_Cria(2,
                                                  arvore_Cria(4,arvore_criaVazia(),arvore_criaVazia()),
                                                  arvore_Cria(5,arvore_criaVazia(),arvore_criaVazia())),
                                    arvore_Cria(3,
                                                  arvore_criaVazia(),
                                                  arvore_Cria(6,arvore_criaVazia(),arvore_criaVazia())));

    NoArvore *arvore2 = arvore_Cria(1,
                                    arvore_Cria(2,
                                                  arvore_Cria(4,arvore_criaVazia(),arvore_criaVazia()),
                                                  arvore_Cria(5,
                                                              arvore_Cria(6,arvore_criaVazia(),arvore_criaVazia()),
                                                              arvore_criaVazia())),
                                    arvore_Cria(3,
                                                  arvore_criaVazia(),arvore_criaVazia()));

    printf("Arvore 1:\n\n");
    arvore_imprime(arvore1,0);
    printf("\n");

    printf("Arvore 2:\n\n");
    arvore_imprime(arvore2,0);
    printf("\n");

    n = arvore_folhas(arvore1);
    printf("Total Folhas Arvore 1: %d\n", n);
    n = arvore_folhas(arvore2);
    printf("Total Folhas Arvore 2: %d\n", n);

    n = arvore_intermediarios(arvore1);
    printf("Total Intermediarios Arvore 1: %d\n", n);
    n = arvore_intermediarios(arvore2);
    printf("Total Intermediarios Arvore 2: %d\n", n);

    n = arvore_pares(arvore1);
    printf("Total Pares Arvore 1: %d\n", n);
    n = arvore_pares(arvore2);
    printf("Total Pares Arvore 2: %d\n", n);

    n = arvore_iguais(arvore1, arvore2);
    printf("Arvore 1 igual a arvore 2? %s\n", n == 1? "Sim":"Nao");
    n = arvore_iguais(arvore1, arvore1);
    printf("Arvore 1 igual a arvore 1? %s\n", n == 1? "Sim":"Nao");

    n = arvore_alturaMaior(arvore1, 2);
    printf("Altura da arvore 1 eh maior que 2? %s\n", n == 1? "Sim":"Nao");
    n = arvore_alturaMaior(arvore2, 2);
    printf("Altura da arvore 2 eh maior que 2? %s\n\n", n == 1? "Sim":"Nao");

    return 0;
}
```