

## ***Disciplina: Estrutura de Dados II***

***Professor: Adilso Nunes de Souza***

### **Lista de exercícios 3**

**1** – Seguindo o modelo implementado de árvore crie as funcionalidades solicitadas:

1.1 Função que retorne a quantidade de nós folhas existentes na árvore.

```
int qtd_folhas(arvore *t)
{
    if(!t)
        return 0;
    else if((t->sae == NULL) && (t->sad == NULL))
        return 1;
    return(qtd_folhas(t->sae) + qtd_folhas(t->sad));
}
```

1.2 Função que retorne a quantidade de nós que guardam valores maiores que um determinado valor x informado.

```
int verifica_maior(arvore *t, int v)
{
    if(!t)
        return 0;
    if(t->info > v)
        return 1 + (verifica_maior(t->sae, v) + verifica_maior(t->sad, v));
    else
        return (verifica_maior(t->sae, v) + verifica_maior(t->sad, v));
}
```

1.3 Função que retorne a quantidade de nós que possuem apenas um filho.

```
int somente_um_filho(arvore *t)
{
    if(!t)
        return 0;
    else if(((t->sae != NULL) && (t->sad == NULL)) ||
            ((t->sae == NULL) && (t->sad != NULL)))
        return 1;
    return(somente_um_filho(t->sae) + somente_um_filho(t->sad));
}
```

OBS: Inclua no menu as opções para acionar as funções criadas.

```
case 5:
    system("cls");
    if(testa_vazia(t)) /* Verifica se a árvore está vazia */
        cout << "\n\nArvore vazia!!\n";
    else
    {
        cout << "Quantidade de nos folhas: " << qtd_folhas(t);
    }
}
```

```

        cout << "\n\nSomente um filho: " << somente_um_filho(t);
        cout << "\n\nInforme o valor a ser comparado: ";
        cin >> num;
        fflush(stdin);
        cout << "\n\nQuantidade de nos com valores maiores que " << num << ": " <<
verifica_maior(t, num);
    }
    getchar();
    break;

```

- 2 – Implemente um programa que receberá a informação de quantos nós a árvore irá possuir e qual o valor do maior nó e o sistema vai gerar valores aleatórios para preencher toda a árvore, após deverá ser possível incluir novos valores, mostrar a árvore.

```

void inserir_aleatorio(arvore **t, int qtd, int maior)
{
    int v;
    for(int i = 0; i < qtd; i++)
    {
        do{
            v = rand() % maior;
        }while(v < 1);
        inserir(&(*t), v);
    }
}

```

OBS: está função gera valores duplicados, caso não for aceito duplicados, sugere-se armazenar os valores em um vetor, podendo fazer a verificação de duplicidade, depois percorre os elementos do vetor chamando a função inserir.