

ESTRUTURA DE DADOS – P3

1) Considere a seguinte estrutura para uma lista duplamente encadeada

```
struct NoLista{  
    int chave;  
    NoLista *ant, *prox;  
};
```

```
struct Lista {  
    NoLista *prim, *ult;  
};
```

- a) Escreva uma função recursiva que conte quantos elementos tem na lista. O protótipo deve ser o seguinte:
int contaElementos(NoLista* n);
- b) Escreva uma função que quebre uma lista em duas listas. O primeiro elemento da segunda lista será o elemento o primeiro elemento da segunda metade da lista. O protótipo da função deve ser o seguinte:
Lista* divideAoMeio(Lista* l1)

A função recebe como parâmetro a lista que deverá ser quebrada e devolve o ponteiro da nova lista criada para guardar os elementos da segunda metade da lista.

2) Considere uma árvore binária onde cada nó da árvore armazena uma lista simplesmente encadeada:

```
struct NoLista{  
    int chave;  
    NoLista *prox;  
}
```

```
struct NoArvore{  
    NoLista *lista;  
    NoArvore *esq, *dir;  
}
```

- a) Escreva um algoritmo que imprima os elementos das listas que estão nas folhas.
 - b) Escreva um algoritmo que imprima o número de elementos total, em todas as listas, de todos os nós.
- 3) Partindo de uma árvore AVL inicialmente vazia:
- a) Insira as seguintes chaves: 10, 20, 30, 25, 28, 31, 32
 - b) Quantas rotações foram feitas? De que tipo?
 - c) A altura da árvore é mínima? Justifique.
 - d) Visite a árvore em pré-ordem.