Estrutura de Dados - Lista de Exercícios 3– Lista Encadeada – Prof. Julio

1. Considere a implementação de uma lista simplesmente encadeada conforme implementada a seguir:

```
struct no{
   int info;
   struct no *prox;
};
struct no *L, *L1, *L2;
```

   Implemente as funções a seguir:

   a) função **`comprimento (struct no *L)`**, que retorne um valor inteiro igual ao número de elementos da lista encadeada L;

   b) função **`existe (struct no *L, int n)`**, que retorne 1 se o valor de n for encontrado na lista L, do contrário retorne 0;

   c) função **`iguais (struct no *L1, struct no *L2)`** que retorne 1 se L1 = L2 e 0 se L1 <> L2. L1 e L2 são listas encadeadas distintas;

   d) função **`copia (struct no *L1, struct no *L2)`** para montar uma cópia L2 da lista encadeada L1. Obs.: ao final teremos duas listas com os mesmos elementos;

   e) função **`diferença (struct no **L, struct no *L1, struct no *L2)`** para construir a lista encadeada L igual à diferença L1-L2. Obs.: a lista L deve conter os elementos de L1 que não estão em L2; os elementos das listas L1 e L2 devem permanecer intactos, ou seja, para criar a lista L deve ser alocado espaço em memória para cada elemento;

   f) função **`comuns (struct no *L1, struct no *L2)`** que deve retornar um valor inteiro igual ao número de valores comuns às duas listas encadeadas L1 e L2;

2. Os dados relativos a um grupo de atletas foram organizados em uma lista linear encadeada. O campo de informação de cada nodo desta lista apresenta o nome e a altura de um atleta, conforme exibido na implementação a seguir:

```
struct no{
     char nome[30];
     float altura;
     struct no *prox;
};
struct no *atletas;
```

   Implemente as funções a seguir:

   a) uma função **`nroAltletas (struct no *Latletas)`** que retorna o número total de atletas desta lista;

   b) uma função **`exibeMaiores(struct no *Latletas,int altura)`** para imprimir o nome de todos os atletas que apresentam altura superior a altura passada por parâmetro;

   c) uma função **`dividirLista (struct no *atletas, struct no **Lmaiores, struct no **Lmenores, int altura)`** para dividir esta lista em duas outras, uma com os dados dos atletas com altura inferior a altura passada por parâmetro, e a outra com os atletas com altura igual ou superior a este valor. Nesta operação não deverão ser alocadas novas áreas de memória - deverão ser utilizadas as mesmas da lista original.