Lista de Exercícios de Estruturas de Dados I

Q1) Considerando as seguintes declarações de lista encadeada:

```
struct lista{
        int info;
        struct lista *prox;
};
```

a) escreva uma função em C que, dada uma lista **l** qualquer, inverta os elementos de **l**. O protótipo da função de inversão é o seguinte: **void inverte (struct lista* l);**

b) escreva uma função em C que, dada uma lista **l** qualquer, inverta os elementos de **l** em uma outra lista de saída. Portanto, a lista de entrada não pode ser alterada. O protótipo da função de inversão é o seguinte: **struct lista * inverte (struct lista* l).**

Q2) Considerando as seguintes declarações de uma lista encadeada:

```
typedef struct lista{
        int mat;
        float cr;
        struct lista *prox;
} TLista;
```

Escreva uma função em C que, dadas duas listas l1 e l2 encadeadas, verifique se l1 e l2 são iguais. As listas l1 e l2 devem permanecer inalteradas. Esta função retorna 1 se as listas são iguais e 0, caso contrário. O protótipo desta função é o seguinte: **int igual (TLista *l1, TLista *l2).**

Q3) Considere seguinte declaração de uma lista simplesmente encadeada:

```
typedef struct lista{
        int info;
        struct lista *prox;
}TLista;
```

Escreva uma função em C que, dada uma lista l qualquer, desloque uma vez os elementos de l, de acordo com n. Se n é ímpar, o elemento que está na última posição passa a ser o primeiro quando a lista é deslocada. Senão, o elemento que está na primeira posição passa a ser o último. O protótipo desta função é o seguinte: **TLista* desloca (TLista* l, int n)**.

Q4) Considere a existência de um tipo que representa um aluno numa universidade hipotética:

```
typedef struct aluno {
        int mat;
        float cr;
        struct aluno *prox;
}TAluno;
```

Escreva uma função que copie uma lista. A lista original deve permanecer inalterada. O protótipo da função é o seguinte: **TAluno *copia (TAluno *l).**