

1º Trabalho da disciplina de Estrutura de Dados I (noturno)
Prof. Dr. Glauco Vitor Pedrosa

1. Usando uma estrutura de **Lista Estática ORDENADA**, implemente as funções solicitadas abaixo usando a linguagem C e considere que o campo *dados* da nossa estrutura para Lista seja o seguinte:

```
#define Max 50

struct produto{
    int codProd;           //código do produto
    char nomeProd[10];     //nome do produto
    float valor;           //valor do produto
    int qtdeEstoque;       //quantidade disponível em estoque
}

typedef struct produto Produto;

struct Lista{
    Produto dados[Max];    //arranjo com os elementos da lista
    int FL;                // índice da primeira posição livre da lista
};

typedef struct Lista Lista;
```

- a) Implemente uma função para inserir produtos ordenados pelo código em uma lista. Essa função recebe como parâmetro um tipo de dado Produto e retorna 1 se o produto foi inserido com sucesso na lista ou 0 se houve algum problema na inserção do produto na lista, por exemplo, a lista está cheia, lista não inicializada e/ou código do produto já está cadastrado. O protótipo dessa função é definido por:

```
int inserir_produto (Lista* L, Produto p);
```

- b) Função para remover os n primeiros elementos da lista. A função recebe como entrada o endereço da lista, e o valor de n (um número inteiro não negativo). Caso n seja maior que o tamanho da lista, todos os elementos da lista são removidos e a lista resultante deve ser vazia, Caso a operação seja executada com sucesso retorne (1), caso contrário, retorne (0). O protótipo da função é definido por:

```
int remover_produto (Lista *L, int n);
```

- c) Função para trocar dois elementos de posição. A função recebe como entrada o endereço da lista, e o índice de duas posições da lista. Caso a lista exista, possua pelo menos dois elementos, e as posições sejam válidas, o campo nomeProd das posições indicadas são trocados entre si e a função retorna sucesso (1). Caso contrário, a função retorna insucesso (0). O protótipo da função é definido por:

```
int trocaProdutos (Lista *L, int pos1, int pos2);
```

- d) Implemente uma função para buscar a posição e as informações do produto com menor numero de peças em estoque. A função recebe como entrada o endereço da lista e retorna a posição e o produto que tem menor quantidade de peças em estoque. Caso a lista não exista ou não possua elementos, a função retorna insucesso (0). Caso seja possível encontrar o maior elemento a função retorna sucesso. O protótipo da função é definido por:

```
int busca_produto (Lista *L, int *pos, Produto *MenorEstoque);
```

- e) Implemente uma função para efetuar a compra de um produto. A função recebe como parâmetro o código do produto a ser comprado e a quantidade a ser comprada. Caso o produto exista e a quantidade em estoque seja suficiente para finalizar a compra, a função retorna um inteiro indicando sucesso (1) e decrementa a quantidade em estoque desse produto de acordo com o que foi comprado. Caso haja alguma falha, lista inexistente, lista vazia, produto inexistente ou quantidade insuficiente em estoque, a função retorna insucesso (0). O protótipo é definido por:

```
int efetuarCompra(Lista *L, int codProd, int qtde);
```

- f) Implemente uma função para imprimir a lista de produtos:

```
void imprimeLista(Lista *L);
```