



Listas encadeadas são estruturas em que, para cada novo elemento inserido, é necessária a alocação de um espaço de memória para armazená-lo. Desta forma, o espaço total de memória gasto pela estrutura é proporcional ao número de elementos nela armazenados.

A estrutura consiste numa sequência encadeada de elementos, em geral chamados de *nós* da lista. A lista é representada por um ponteiro para o primeiro elemento (ou nó).

Declaração de um nó em linguagem C:

```
struct no {  
    int info;  
    struct no *prox;  
};
```

Declaração de um tipo lista:

```
typedef struct no lista;
```

Exercícios Práticos:

1) Usando a definição abaixo para uma lista encadeada:

```
struct no {  
    int info;  
    struct no *prox;  
};  
typedef struct no Lista;
```

Criar e iniciar uma lista com dez elementos numéricos inteiros, calcular e exibir:

- a) A quantidade de números pares
- b) A quantidade de números ímpares
- c) A média entre os números

O programa deverá ser realizado utilizando-se as funções e procedimento abaixo:

```
// Função que inicia a lista  
Lista* inicia(){  
    return NULL;  
}  
  
// Função para inserir um novo elemento na lista  
Lista* insere(Lista *l, int i){  
    Lista *novo = (Lista *) malloc (sizeof(Lista));  
    novo->info = i;  
    novo->prox = l;  
    return novo;  
}  
  
// Função para exibir na tela os valores armazenados na lista  
void imprime(Lista *l){  
    Lista *p;  
  
    for(p=l; p!=NULL; p=p->prox)  
        printf("  %d",p->info);  
}
```

2) Altere a função `insere` do exercício 1 para que os elementos sejam inseridos de forma ordenada (do menor para o maior).

Sugestão:

Para fazer a inserção ordenada, considere os dois casos seguintes:

- a) O elemento deve ser inserido no início da lista (a lista é vazia ou o novo elemento é menor que o primeiro da lista)
- b) O elemento deve ser encadeado no meio ou no final da lista. Nesse caso, utilize dois ponteiros, um para apontar para o nó corrente, e outro indicando o nó anterior.

3) Defina uma função `busca` para pesquisar um dado valor numa lista. Essa função deve retornar a posição (primeira) em que o valor foi encontrado.

4) Defina uma estratégia para a remoção de um dado elemento numa lista encadeada e implemente a função.

5) Teste o programa abaixo que implementa o gerenciamento de uma coleção de produtos, sendo cada produto identificado por: número, descrição, preço e estoque. O programa permite a inserção de um novo produto, e a impressão da lista completa de produtos. Complete o programa incluindo dois novos serviços: (i) alterar o preço de um produto e (ii) remover um produto da lista a partir de seu número.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Declaração da estrutura produto, com 5 campos
struct produto {
    int nro;
    char desc[15];
    float preco;
    int estoque;
    struct produto *prox;
};
typedef struct produto ListaProd;

// Função para iniciar a lista com o valor Nulo.
ListaProd* inicia(){
    return NULL;
}

// Função para a inserção ordenada (por número) dos produtos
ListaProd* insereOrdem(ListaProd *l, int i, char nome[], float prc, int
est){
    ListaProd *p=l, *ant=NULL;

    // Alocação e inicialização de valores para o nó "novo".
    ListaProd *novo = (ListaProd *) malloc (sizeof(ListaProd));
    novo->nro = i;
    strcpy(novo->desc, nome);
    novo->preco = prc;
    novo->estoque = est;

    // Produto deve ser inserido no início da lista (primeiro ou inferior)
    if (p == NULL || p->nro > i){
        novo->prox = p;
        return novo; }
}
```

```

    else {
        // Produto deve ser encadeado no meio (ou final) da lista
        while(p!= NULL && p->nro <= i){
            ant=p;
            p=p->prox;
        }
        novo->prox = ant->prox;
        ant->prox=novo;
        return 1;
    }
}

// Função para exibir os produtos na tela
void imprime(ListaProd *l){
    ListaProd *p;

    for(p=l; p!=NULL; p=p->prox){
        printf("\nNumero:  %d",p->nro);
        printf(" | Descricao: %s",p->desc);
        printf(" | Preco:  %.2f",p->preco);
        printf(" | Estoque:  %d",p->estoque);    }
}

// Função principal
int main()
{
    ListaProd *l;
    int num, nro, est, op=0;
    char desc[15];
    float m, prc;

    l = inicia();
    do { system("cls");
        printf(" \n\t----- OPCOES ----- ");
        printf(" \n\t1 - Inserir um novo produto ");
        printf(" \n\t2 - Listar produtos ");
        printf(" \n\t3 - Encerrar ");
        printf(" \n\tOpcao: ");
        scanf("%d",&op);
        switch(op){
            case 1: system("cls");
                printf(" \n\n--- Novo produto --- ");
                printf("\n Numero: ");
                scanf("%d",&nro);
                printf(" Descricao: ");
                scanf("%s",&desc);
                printf(" Preco: ");
                scanf("%f",&prc);
                printf(" Estoque: ");
                scanf("%d",&est);
                l = insereOrdem(l, nro, desc, prc, est);
                break;
            case 2: system("cls");
                printf("\n\n--- Lista de produtos --- \n");
                imprime(l);
                system("pause>>null");
                break;
            otherwise: printf(" Opcao invalida ");
        }
    } while (op != 3);
    system("PAUSE>>null");
    return 0;
}

```