

Exercícios

1. Considerando as definições a seguir, faça o que é pedido nos itens abaixo:

```
typedef struct {  
    int dia;  
    int mes;  
    int ano;  
} Data;
```

```
struct funcionario{  
    int id;  
    char nome[41];  
    double salario;  
    Data nascimento;  
    struct funcionario *proximo;  
};  
typedef struct funcionario Funcionario;
```

- Crie as estruturas indicadas, e crie o primeiro funcionário da lista;
- Adicione um segundo funcionário ao início da lista;
- Crie uma função capaz de imprimir todos os funcionários;

Exercícios

2. Considerando a estrutura proposta no exercício anterior, faça as seguintes adaptações em seu programa:
 - a. O programa deve ler (do teclado) um inteiro N que representará o número de registros que o usuário irá inserir. Após a leitura seu programa deve ler os dados dos N registros e os inserir na lista encadeada.
 - b. Imprima a lista para ver se todos os elementos estão presentes
 - c. Faça uma função que deleta um funcionário. A função deve receber como parâmetro a lista, e o id do funcionário a ser deletado, e deve retornar o primeiro elemento da lista

Exercícios

3. Implemente uma função que receba um vetor de valores inteiros com N elementos e construa uma lista encadeada armazenando os elementos do vetor (elemento a elemento). Assim, se for recebido por parâmetro o vetor $v[4] = \{1, 21, 4, 6\}$ a função deve retornar uma lista encadeada onde o primeiro elemento é '1', o segundo o '21', o terceiro o '4' e assim por diante. A função deve ter a seguinte assinatura: *Lista*Int *constroiLista (int n, int *v);

Exercícios

4. Transforme a estrutura da lista implementada nas questões 1 e 2 em uma lista duplamente encadeada. E implemente as seguintes funcionalidades:
 - a. Imprimir a lista do primeiro para o último elemento, e depois do último para o primeiro.
 - b. Crie uma função de busca que apresenta as informações de um funcionário. A busca deve ser feita utilizando o id.
 - c. Atualize a função de delete.