

TADs sobre Listas Lineares  
Prof. Hebert Coelho

## **Conteúdo**

<b>1</b>	<b>TAD: Lista Circular Encadeada com nó cabeça</b>	<b>2</b>
<b>2</b>	<b>TAD: Lista Circular Encadeada com nó cabeça ordenada</b>	<b>3</b>
<b>3</b>	<b>TAD: Lista Duplamente Encadeada não circular sem nó cabeça ordenada</b>	<b>4</b>

# 1 TAD: Lista Circular Encadeada com nó cabeça



(+)

Desenvolva um TAD que represente uma lista Circular Encadeada com nó cabeça. Os dados serão uma struct a escolha mas com no mínimo 3 campos de dados mais um campo ponteiro para o próximo elemento.

```
struct dados {  
    tipo1 campo1  
    tipo2 campo2  
    tipo3 campo3
```

...

```
    tipoN campoN  
struct dados *next;  
}
```

Inclua no mínimo as seguintes funções:

Criação da lista;

Inserção de um elemento X;

Remoção de um elemento X;

Apagar a lista toda;

Imprimir a lista;

Buscar um elemento X;

Para este exercício, comprima todos os arquivos (.h e .c) e submeta o arquivo .zip resultante. A correção será feita manualmente.

## 2 TAD: Lista Circular Encadeada com nó cabeça ordenada



(+)

Desenvolva um TAD que represente uma lista Circular Encadeada com nó cabeça e ordenada. A lista deve ser ordenada pelo campo denominado chave. Os dados serão uma struct a escolha mas com no mínimo 3 campos de dados mais um campo ponteiro para o próximo elemento.

```
struct dados {  
    int chave;  
    tipo2 campo2;  
    tipo3 campo3;  
    ...  
    tipoN campoN;  
    struct dados *next;  
}
```

Inclua no mínimo as seguintes funções:

Criação da lista;

Inserção de um elemento, a chave não pode ser repetida;

Remoção de um elemento pela chave;

Apagar a lista toda;

Imprimir a lista;

Buscar um elemento X;

Alterar um elemento;

combinar 2 listas;

Determinar o número de elementos

Para este exercício, comprima todos os arquivos (.h e .c) e submeta o arquivo .zip resultante. A correção será feita manualmente.

### 3 TAD: Lista Duplamente Encadeada não circular sem nó cabeça ordenada



(+)

Desenvolva um TAD que represente uma lista Duplamente Encadeada não circular sem nó cabeça e ordenada. A lista deve ser ordenada pelo campo denominado chave. Os dados serão uma struct a escolha mas com no mínimo 3 campos de dados mais um campo ponteiro para o próximo elemento.

```
struct dados {  
int chave;  
tipo2 campo2;  
tipo3 campo3;  
...  
tipoN campoN;  
struct dados *next;  
}
```

Inclua no mínimo as seguintes funções:

Criação da lista;

Inserção de um elemento, a chave não pode ser repetida;

Remoção de um elemento pela chave;

Remoção de um elemento por outro campo que não seja a chave a escolha do usuário;

Apagar a lista toda;

Imprimir a lista;

Buscar um elemento X;

Alterar um elemento;

Combinar 2 listas;

Quebrar uma lista em n listas, as chaves de inicio e fim para cada nova lista deve ser passado como parâmetro;

Determinar o número de elementos;

Para este exercício, comprima todos os arquivos (.h e .c) e submeta o arquivo .zip resultante. A correção será feita manualmente.