

---

# BCC202 – Estrutura de Dados I

## 2s2015

### Prática 04

1. Implemente uma TAD Lista utilizando *Array*. Sua TAD deve contemplar:

→ As mesmas operações vistas em sala de aula:

```
void TLista_FazVazia ( TLista * pLista );  
int TLista_EhVazia ( TLista * pLista );  
int TLista_Insera ( TLista * pLista , TItem x ); /*insere um item no final da lista */  
int TLista_Retira ( TLista * pLista , int p , TItem * pX ); /*retira um item da posição p da lista
```

\*/

→ Acrescentar as seguintes operações ao conjunto anteriormente mencionado:

- concatenação de duas listas
- pesquisa por um elemento na lista

2. Implemente uma TAD Lista *com cabeça* utilizando *Ponteiros*. Sua TAD deve contemplar:

→ As mesmas operações vistas em sala de aula:

```
void TLista_Inicia ( TLista * pLista );  
int TLista_EhVazia ( TLista * pLista );  
int TLista_Insera ( TLista * pLista , TItem x ); /* insere um item no final da lista */  
int TLista_RetiraPrimeiro ( TLista * pLista , TItem * pX ); /*retira o primeiro elemento da  
lista */  
void TLista_Imprime ( TLista * pLista );
```

→ Do ponto de vista conceitual, foram vistas três formas de inserção e remoção de elementos de listas encadeadas. No entanto, foi apresentado em aula apenas uma forma para inserção e uma para remoção. Escolha uma forma de inserção e outra de remoção, não apresentada em sala de aula, e faça a sua implementação na TAD Lista.

Seu algoritmo deve permitir que o usuário forneça as entradas correspondentes para testar as operações de sua TAD.

3. Implemente um TAD Lista *sem cabeça* utilizando **Ponteiros**. Sua TAD deve contemplar:

→ As mesmas operações vistas em sala de aula:

```
void TLista_Inicia ( TLista * pLista ) ;  
int TLista_EhVazia ( TLista * pLista ) ;  
int TLista_Inserir ( TLista * pLista , TItem x ) ; /* insere um item no final da lista */  
int TLista_RetiraPrimeiro ( TLista * pLista , TItem * pX ) ; /*retira um item do final da  
lista */
```

## Procedimento de Entrega

Você deve entregar os exercícios no Moodle até o dia **20/12/15** até às **23:55**. **Não serão aceitas soluções entregues após o horário estipulado.**

Para cada questão:

- 1- Crie uma pasta para a questão: PrimeiroNome-UltimoNome-Questao (exemplo: amanda-nascimento-1).
- 2- Crie os arquivos de código fonte necessários para a solução da questão (arquivos .c e .h)
- 3- Compile na linha de comando usando gcc \*.c -o prog.exe -lm.
- 4- Apague os arquivos gerados na compilação (mantenha apenas os arquivos .c e .h).

Compacte todas as pastas criadas anteriormente num único arquivo (“*.zip*”), que deve ser entregue via Moodle. ***O arquivo compactado deve também conter um arquivo “.txt” com seu nome e número de matrícula.***