

Laboratório 04 – Lista Duplamente Encadeada Circular (LSEC)

ATENÇÃO. O trabalho pode ser feito em **duplas**. Somente **UM dos integrantes da dupla deve submeter** o trabalho no *moodle*. Os nome do arquivo deve ser **Aluno01-Aluno02.zip**.

Problema para o laboratório. Implementar uma função para inserir um número indeterminado de elementos (números inteiros) em uma **lista duplamente encadeada circular**. A estrutura da lista deve ser:

```
typedef struct tipoNo ptLSE;  
struct tipoNo{  
    int numero;  
    ptLSE *prox;  
    ptLSE *ant;  
};
```

A função `insere()` recebe como parâmetro um ponteiro para o início da lista (`ptInicio`) e um número inteiro (`num`). O critério para inserção é o seguinte. Você deve procurar o número (`num`) na lista. Se o número for encontrado, você deve alocar memória e inserir o valor `num - 1` uma posição antes do número encontrado e `num + 1` uma posição depois do número encontrado. Se o número (`num`) não for encontrado, você deve inserir o número no início da lista. Por exemplo:

```
Lista Inicial: 1  7  3  
Função: ptIni, 7  
Lista final: 1  6  7  8  3  
Função: ptIni, 2  
Lista final: 2  1  6  7  8  3  
Função: ptIni, 8  
Lista final: 2  1  6  7  7  8  9  3
```

Passos para testar o laboratório (`main()`):

- Insere()** os nodos na lista (número indeterminado de elementos, condição de parada número igual a zero), conforme descrito acima.
- ExibeIni()** a lista todos os nodos da lista do início para o fim
- Função **ExibeInversoNum()** recebe um ponteiro para o início da lista e um número e exhibe a lista criada no item (a) do número informado até o seu sucessor. Isto é, você deve mostrar os elementos da lista percorrendo pelo ponteiro anterior. Se o numero não existir na lista, mostrar essa informação para o usuário. Por exemplo, se a lista for: {1, 5, 9, 14, 7} e o número informado for 9, a função deve exhibir 9, 5, 1, 7, 14.
- Destroi()** a lista, isto é, desaloca todos os elementos da lista.