

# Dominando Estruturas de Dados 1

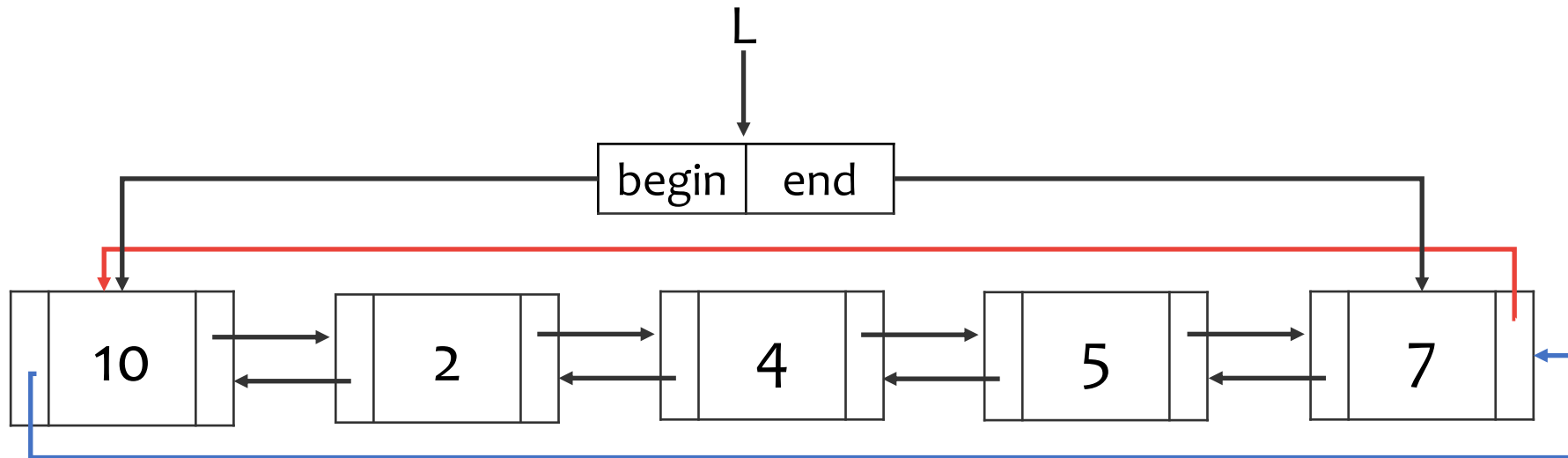
## Listas Circulares

Prof. dr. Samuel Martins (Samuka)  
@xavecoding @hisamuka



# Lista Circular Duplamente Encadeada

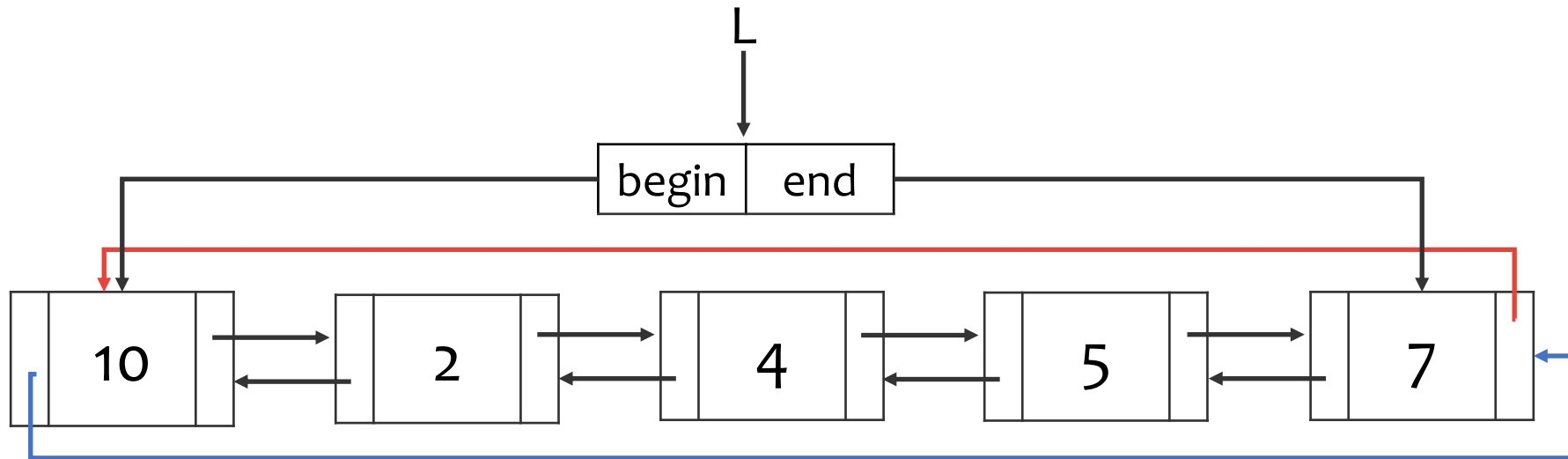
- Variação de **lista duplamente encadeada**;
- O **nó anterior ao nó inicial** é o **nó final**;
- O **próximo nó do final** é o **nó inicial**;



# Definição

```
typedef struct _circ_node {  
    int val;  
    struct _circ_node *prev;  
    struct _circ_node *next;  
} CircNode;
```

```
typedef struct _circ_list {  
    CircNode *begin;  
    CircNode *end;  
    size_t size;  
} CircList;
```



# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

- Inserção na cabeça (início) da lista;
- Impressão dos Elementos da Lista
- Inserção na cauda (fim) da lista;
- Remover elementos da lista;
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i

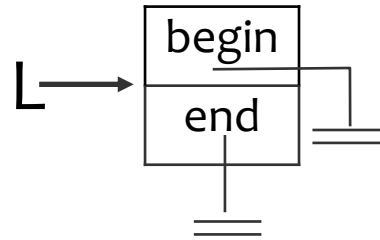
# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

- **Inserção na cabeça (início) da lista;**
- Impressão dos Elementos da Lista
- Inserção na cauda (fim) da lista;
- Remover elementos da lista;
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i

# Inserção na cabeça (início) da lista

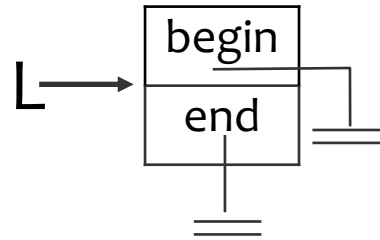
**Caso 1:** Lista está vazia



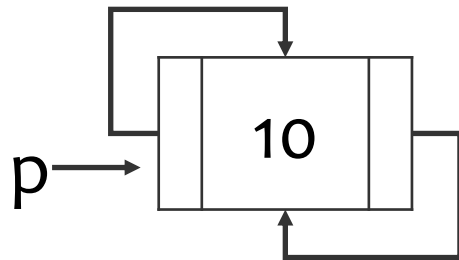
inserir 10

# Inserção na cabeça (início) da lista

**Caso 1:** Lista está vazia



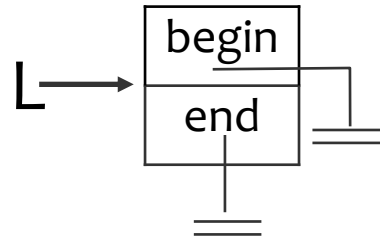
inserir 10



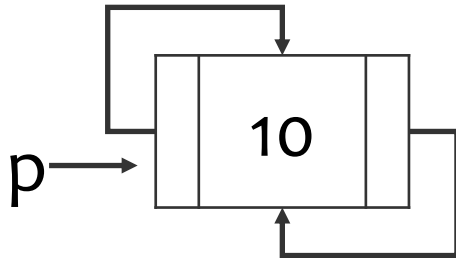
cria nó **p**

# Inserção na cabeça (início) da lista

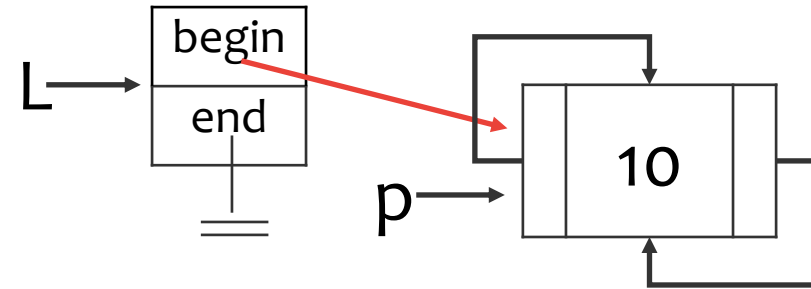
**Caso 1:** Lista está vazia



inserir 10



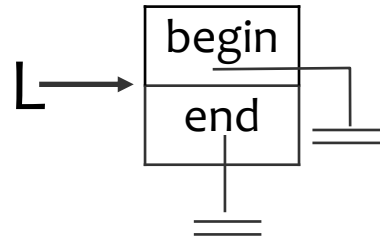
cria nó  $p$



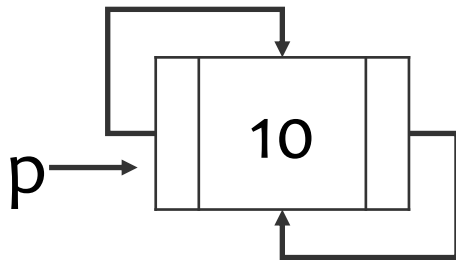


# Inserção na cabeça (início) da lista

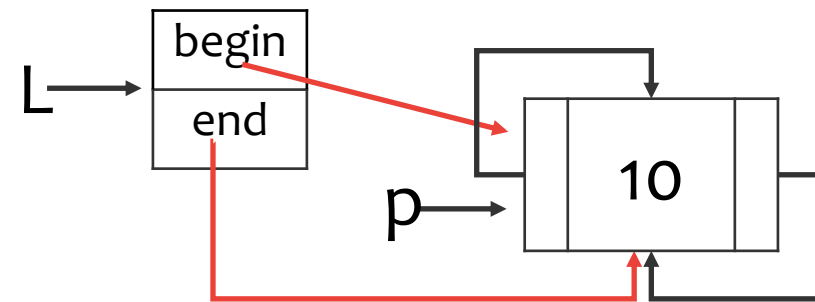
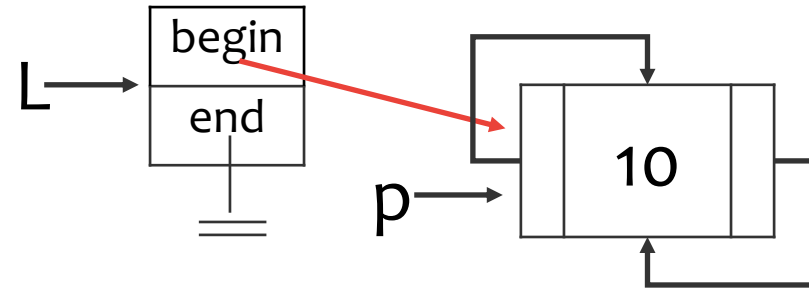
**Caso 1:** Lista está vazia



inserir 10

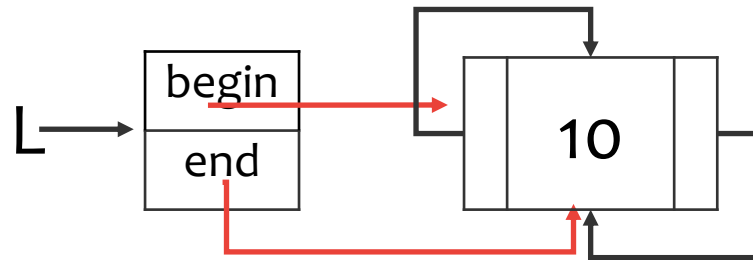


cria nó  $p$



# Inserção na cabeça (início) da lista

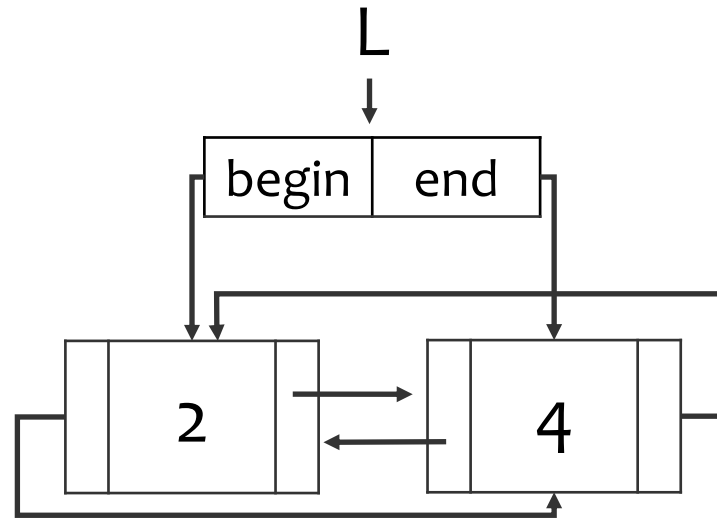
**Caso 1:** Lista está vazia



configuração final da lista  
após a inserção

# Inserção na cabeça (início) da lista

**Caso 2:** Lista possui elementos

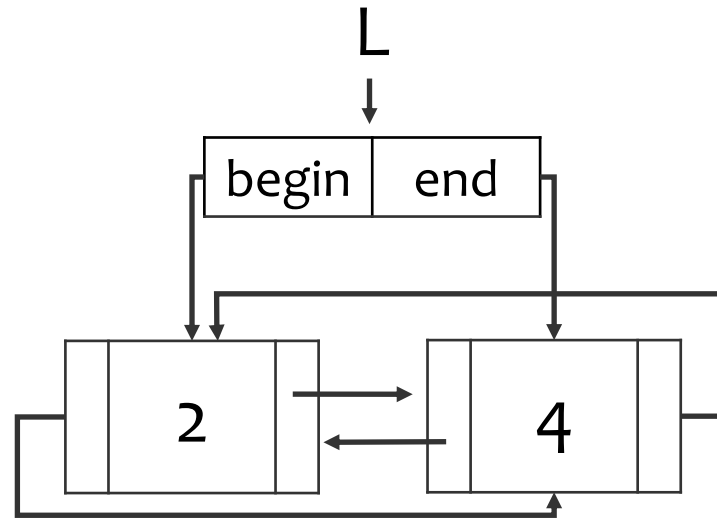
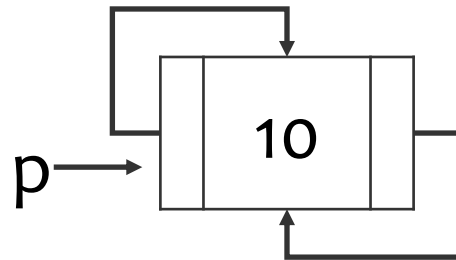


inserir 10

# Inserção na cabeça (início) da lista

**Caso 2:** Lista possui elementos

cria nó **p**



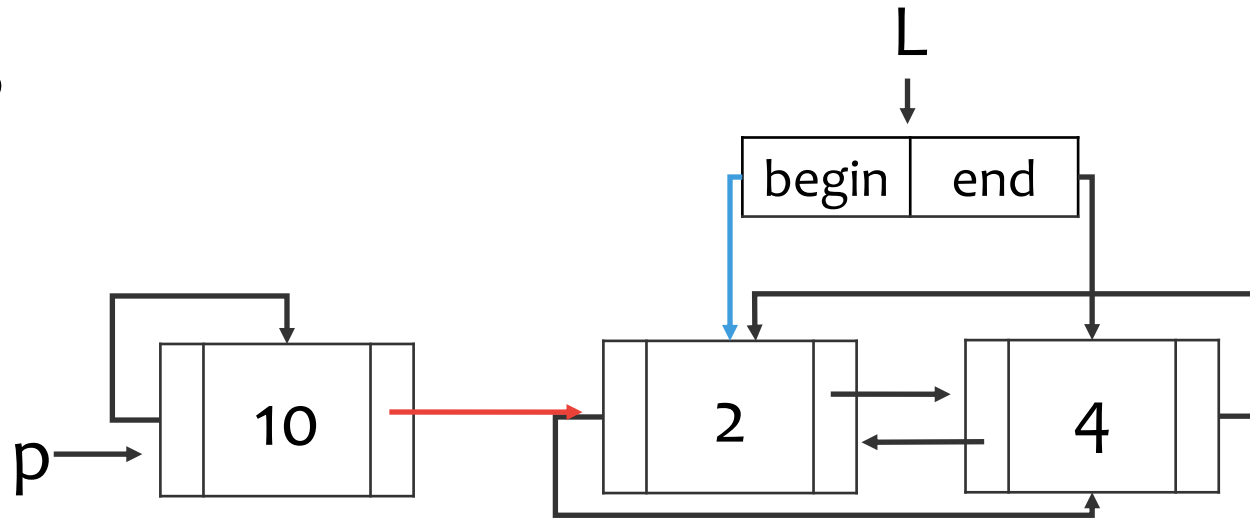
inserir **10**

# Inserção na cabeça (início) da lista

**Caso 2:** Lista possui elementos

cria nó **p**

inserir **10**

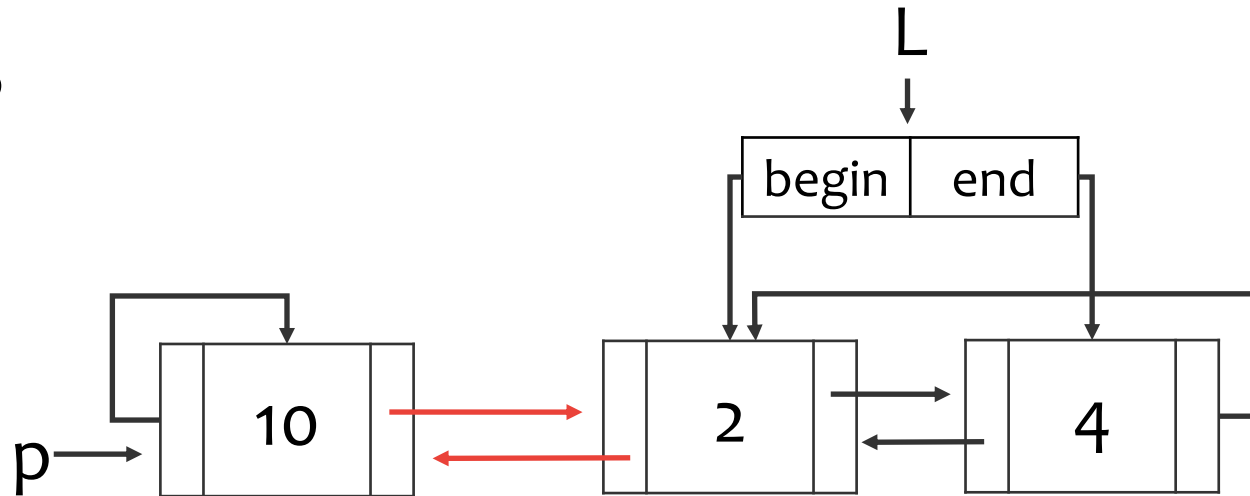


# Inserção na cabeça (início) da lista

**Caso 2:** Lista possui elementos

cria nó **p**

inserir **10**

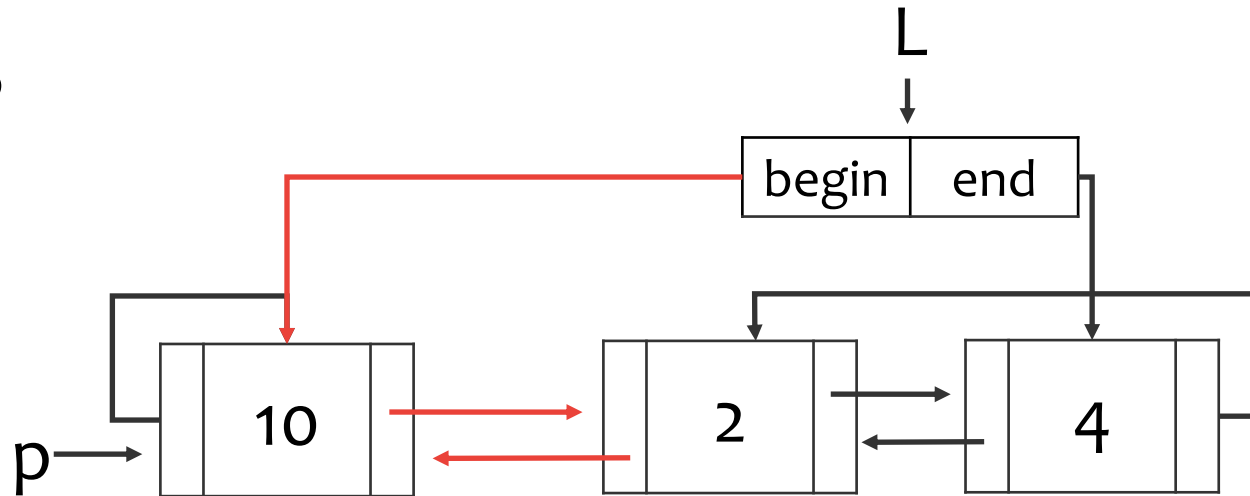


# Inserção na cabeça (início) da lista

**Caso 2:** Lista possui elementos

cria nó **p**

inserir **10**

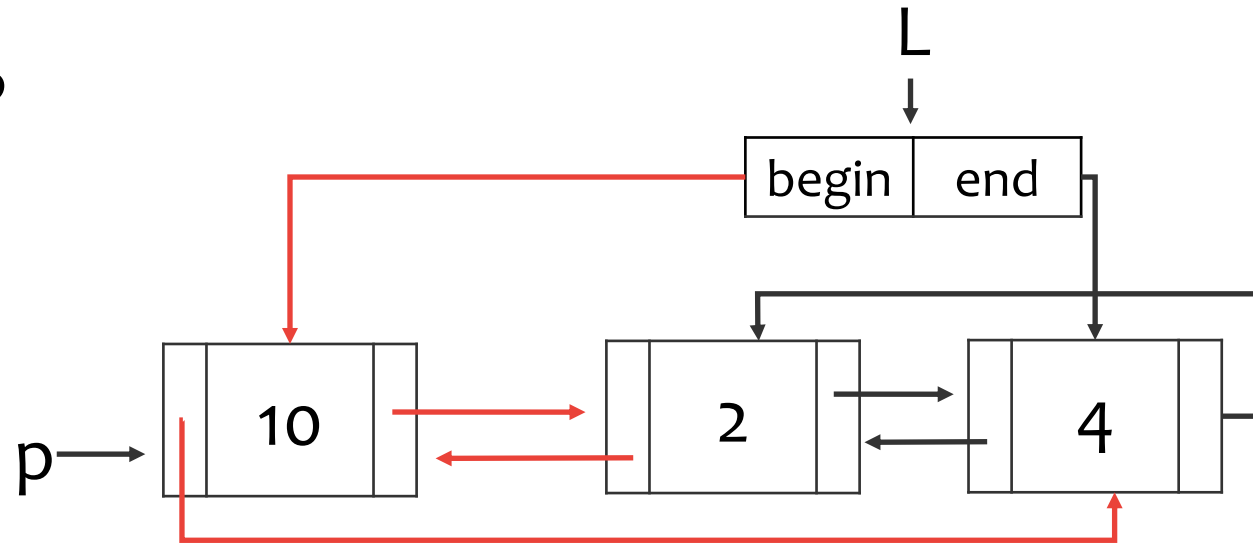


# Inserção na cabeça (início) da lista

**Caso 2:** Lista possui elementos

cria nó **p**

inserir **10**



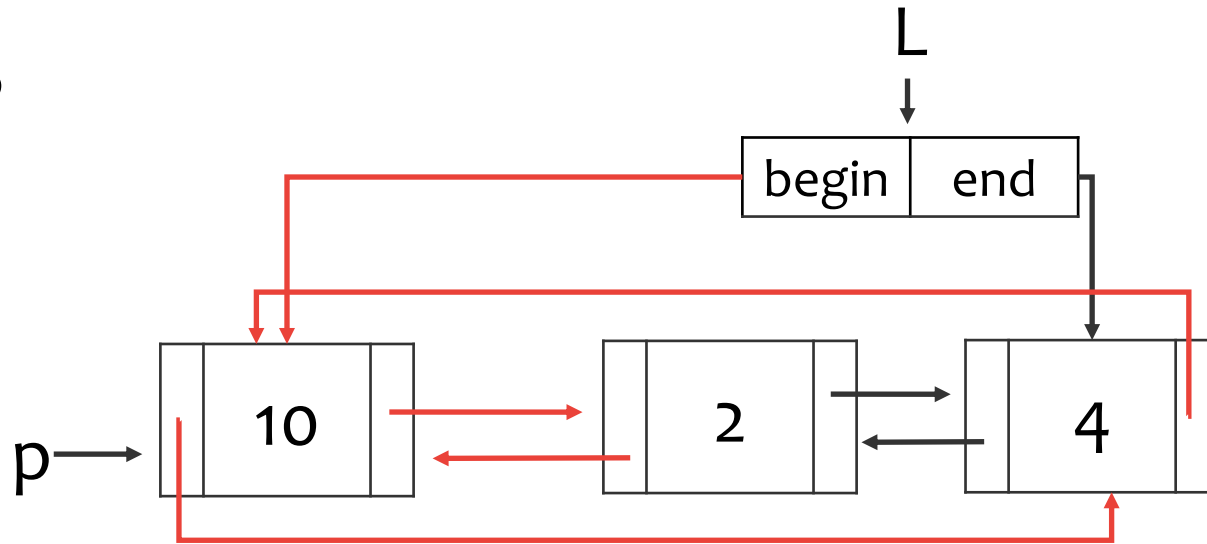


# Inserção na cabeça (início) da lista

**Caso 2:** Lista possui elementos

cria nó **p**

inserir **10**



configuração final da  
lista após a inserção

# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

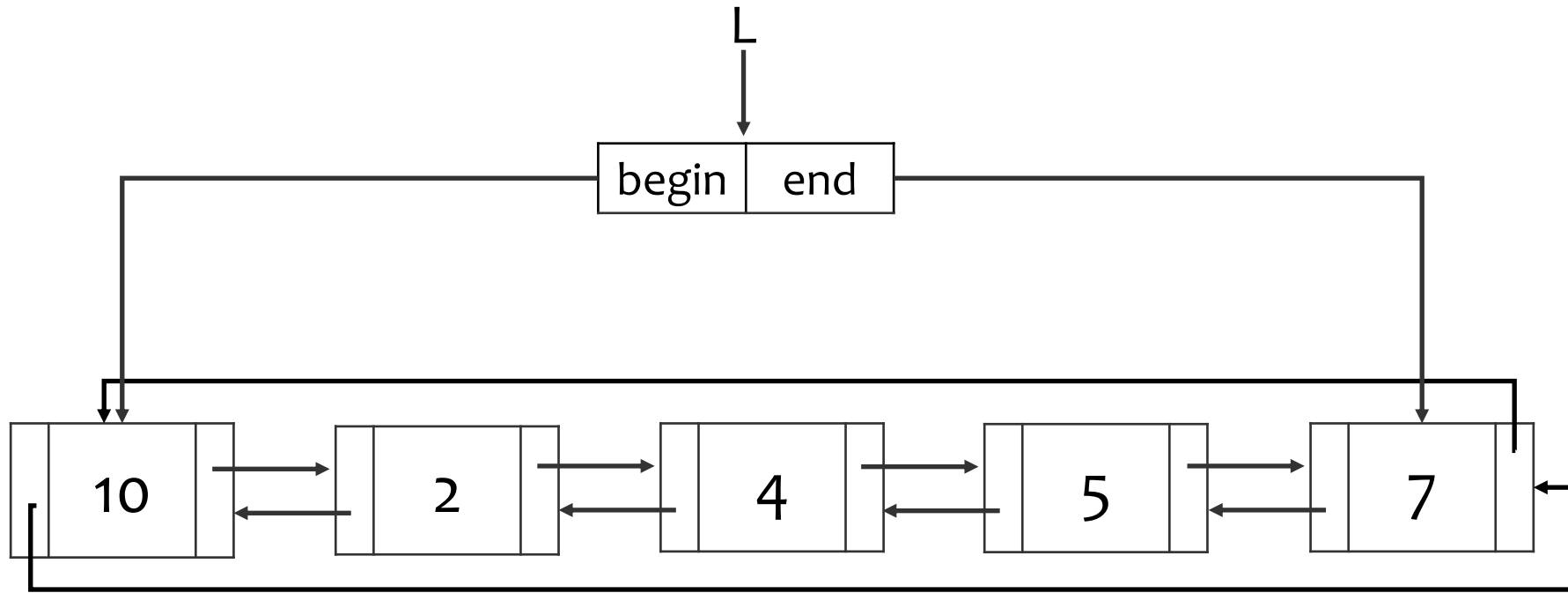
- **Inserção na cabeça (início) da lista;**
- Impressão dos Elementos da Lista
- Inserção na cauda (fim) da lista;
- Remover elementos da lista;
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i

# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

- Inserção na cabeça (início) da lista;
- **Impressão dos Elementos da Lista**
- Inserção na cauda (fim) da lista;
- Remover elementos da lista;
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i

# Impressão dos Elementos da Lista



# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

- Inserção na cabeça (início) da lista;
- **Impressão dos Elementos da Lista**
- Inserção na cauda (fim) da lista;
- Remover elementos da lista;
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i

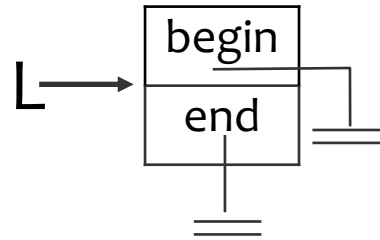
# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

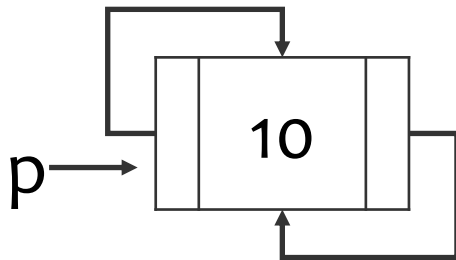
- Inserção na cabeça (início) da lista;
- Impressão dos Elementos da Lista
- **Inserção na cauda (fim) da lista;**
- Remover elementos da lista;
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i

# Inserção na cauda (fim) da lista

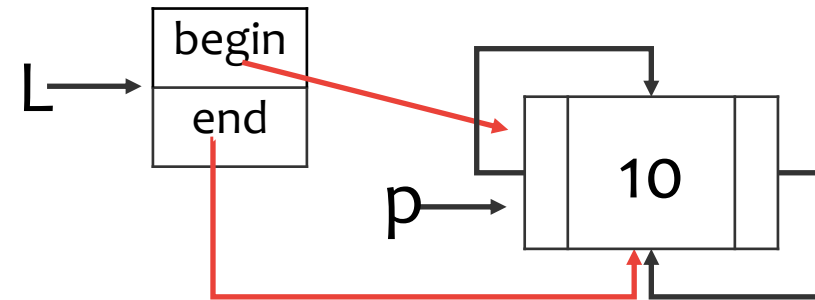
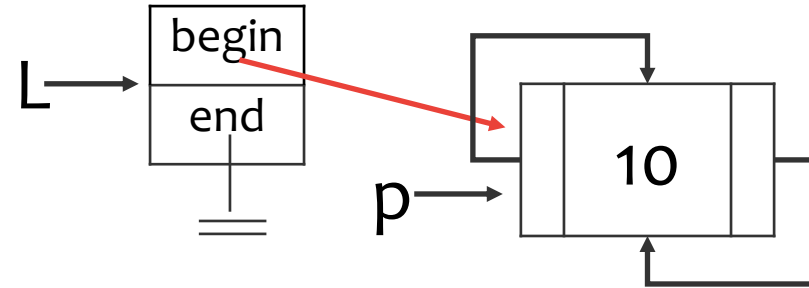
**Caso 1:** Lista está vazia: *idêntico à inserção no começo da lista*



inserir 10



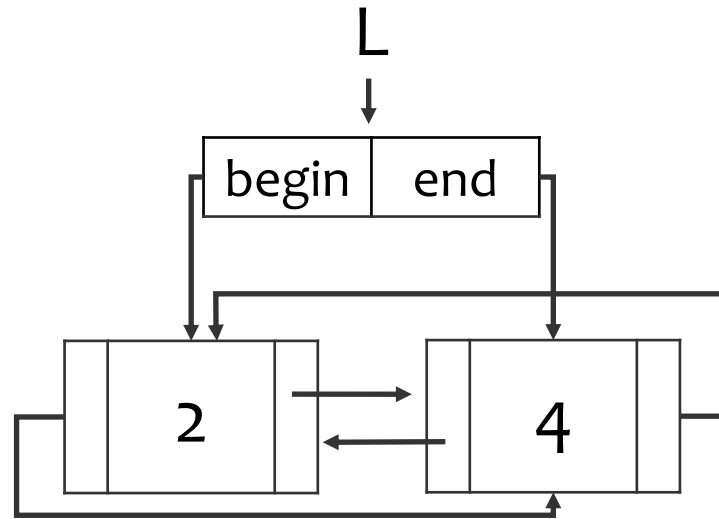
cria nó **p**



# Inserção na cauda (fim) da lista

**Caso 2:** Lista possui elementos

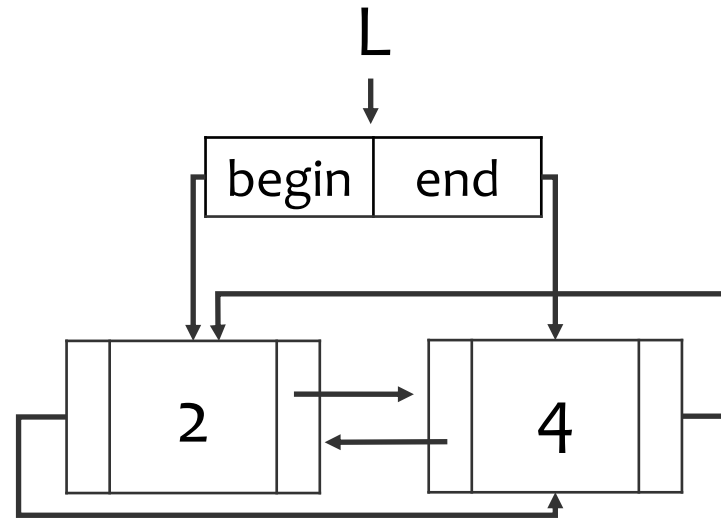
inserir 7





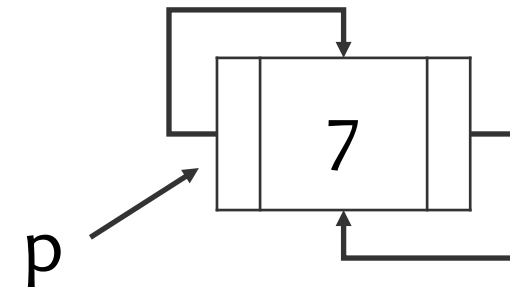
# Inserção na cauda (fim) da lista

**Caso 2:** Lista possui elementos



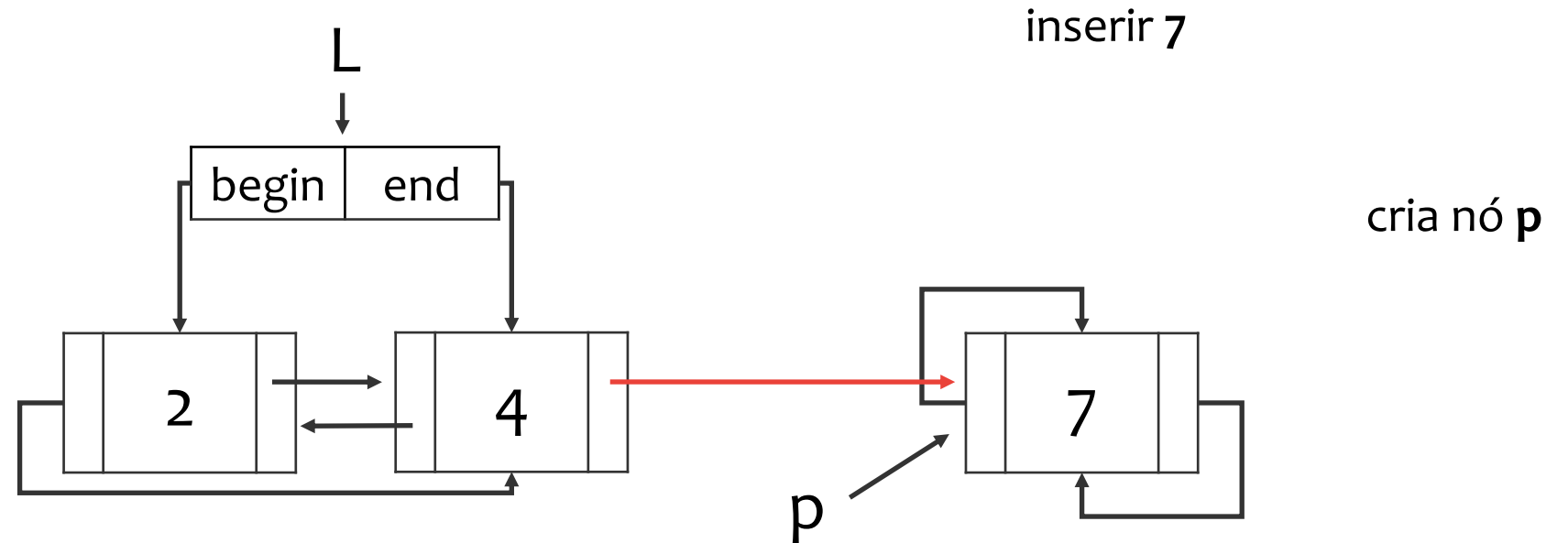
inserir 7

cria nó p



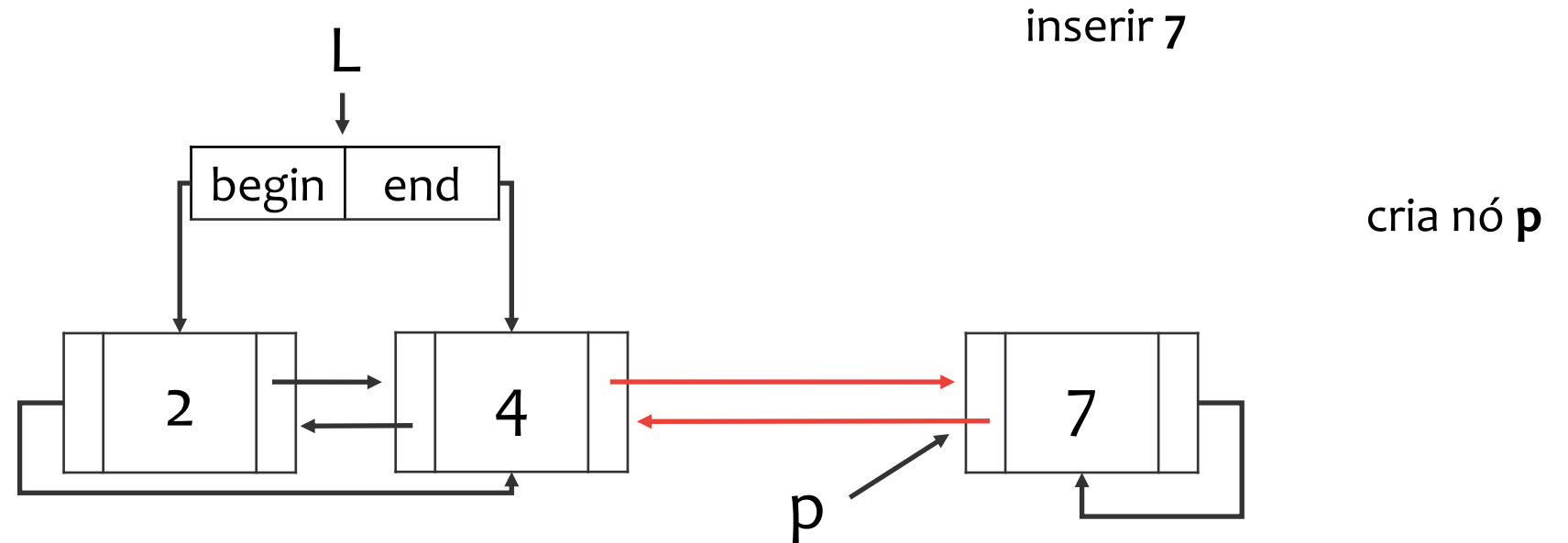
# Inserção na cauda (fim) da lista

**Caso 2:** Lista possui elementos



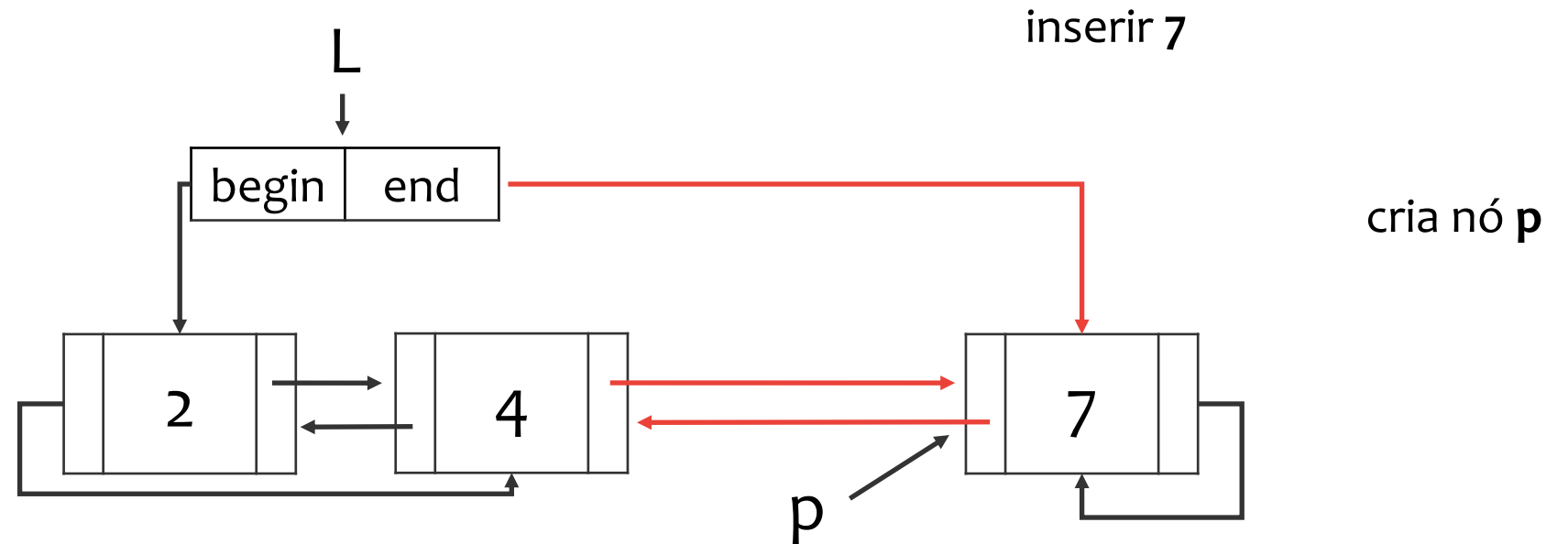
# Inserção na cauda (fim) da lista

**Caso 2:** Lista possui elementos



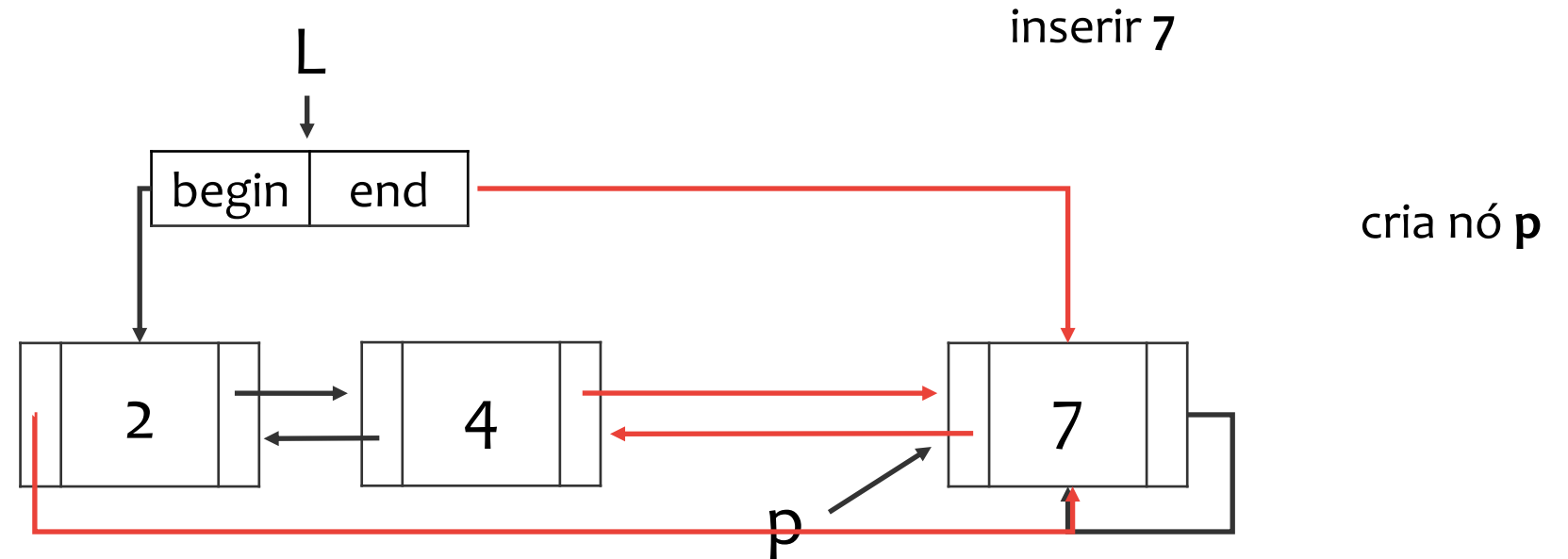
# Inserção na cauda (fim) da lista

**Caso 2:** Lista possui elementos



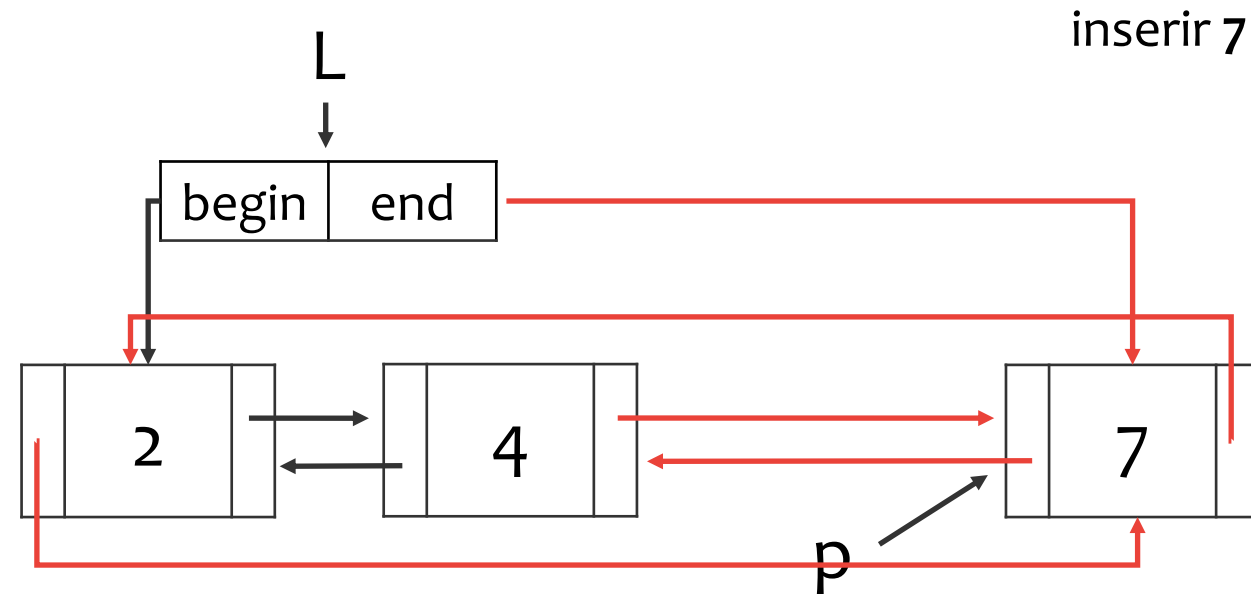
# Inserção na cauda (fim) da lista

**Caso 2:** Lista possui elementos



# Inserção na cauda (fim) da lista

**Caso 2:** Lista possui elementos



cria nó **p**

# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

- Inserção na cabeça (início) da lista;
- Impressão dos Elementos da Lista
- **Inserção na cauda (fim) da lista;**
- Remover elementos da lista;
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i

# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

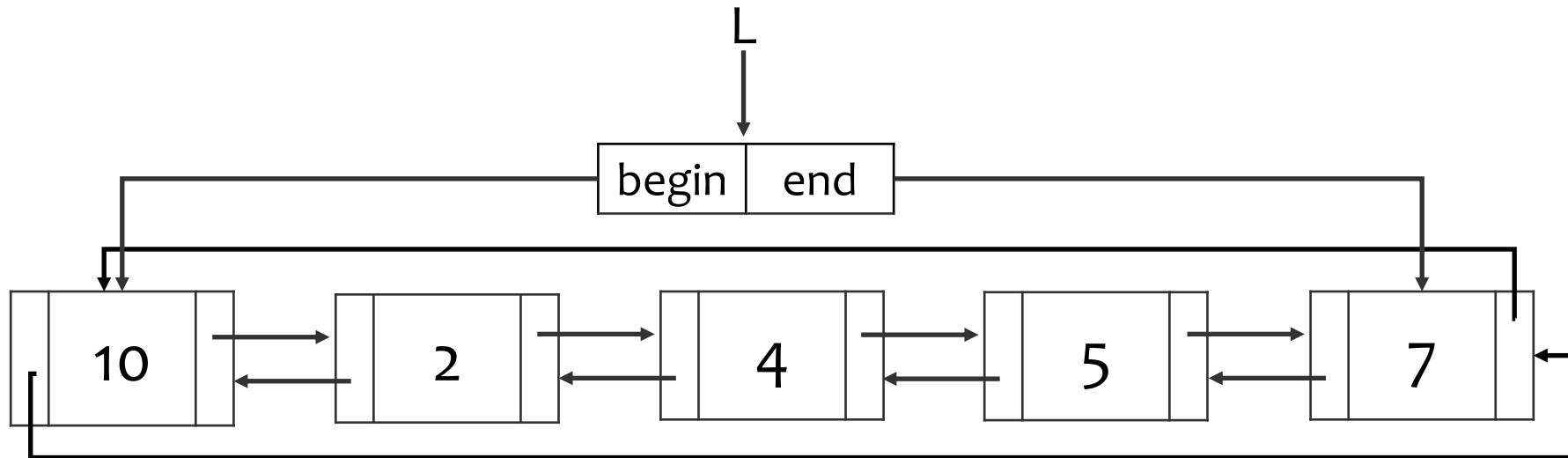
- Inserção na cabeça (início) da lista;
- Impressão dos Elementos da Lista
- Inserção na cauda (fim) da lista;
- **Remover elementos da lista;**
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i



# Remoção de Elementos da Lista

**Caso 1:** Elemento está na **cabeça (início)** da lista

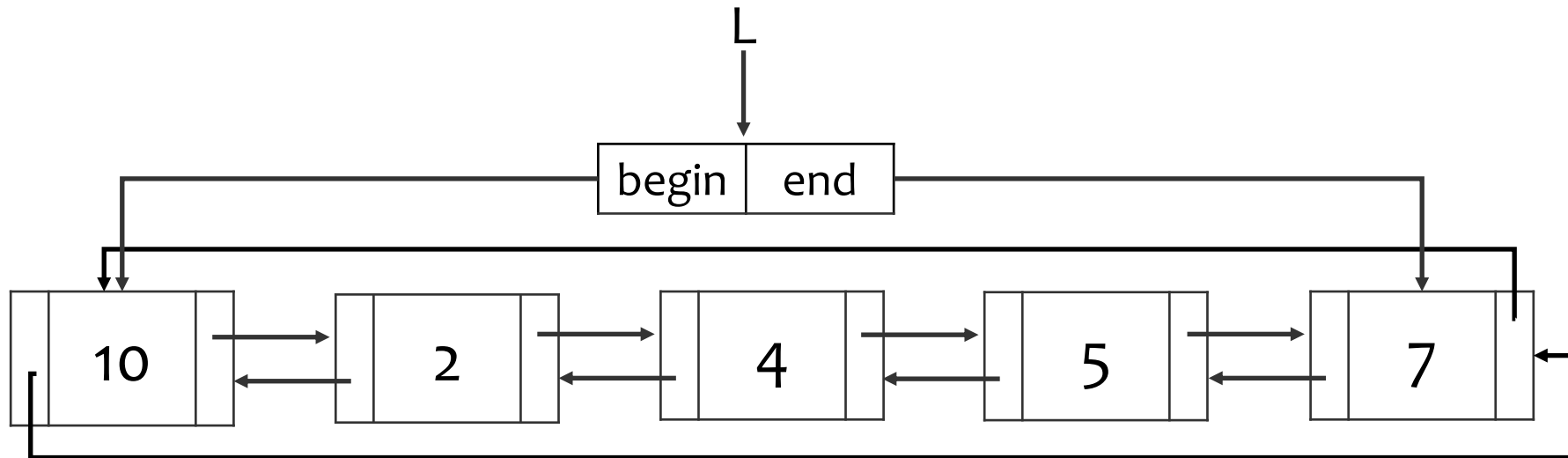
- Mesmos passos da Lista Duplamente Encadeada, apenas atualizando o **encadeamento circular**



# Remoção de Elementos da Lista

**Caso 1:** Elemento está na **cabeça (início)** da lista

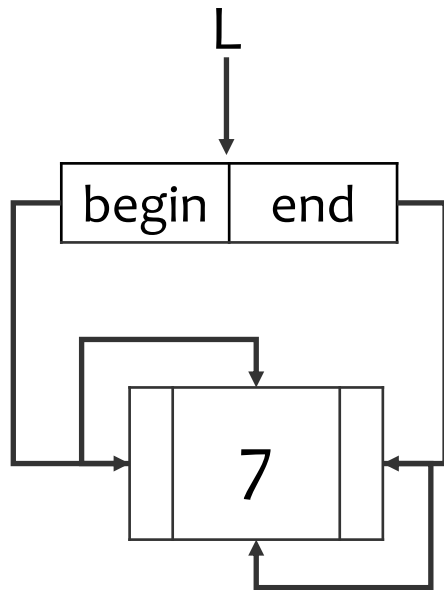
- Mesmos passos da Lista Duplamente Encadeada, apenas atualizando o **encadeamento circular**



# Remoção de Elementos da Lista

**Caso 1:** Elemento está na **cabeça (início)** da lista

- Mesmos passos da Lista Duplamente Encadeada, apenas atualizando o **encadeamento circular**
- A lista só tem um elemento

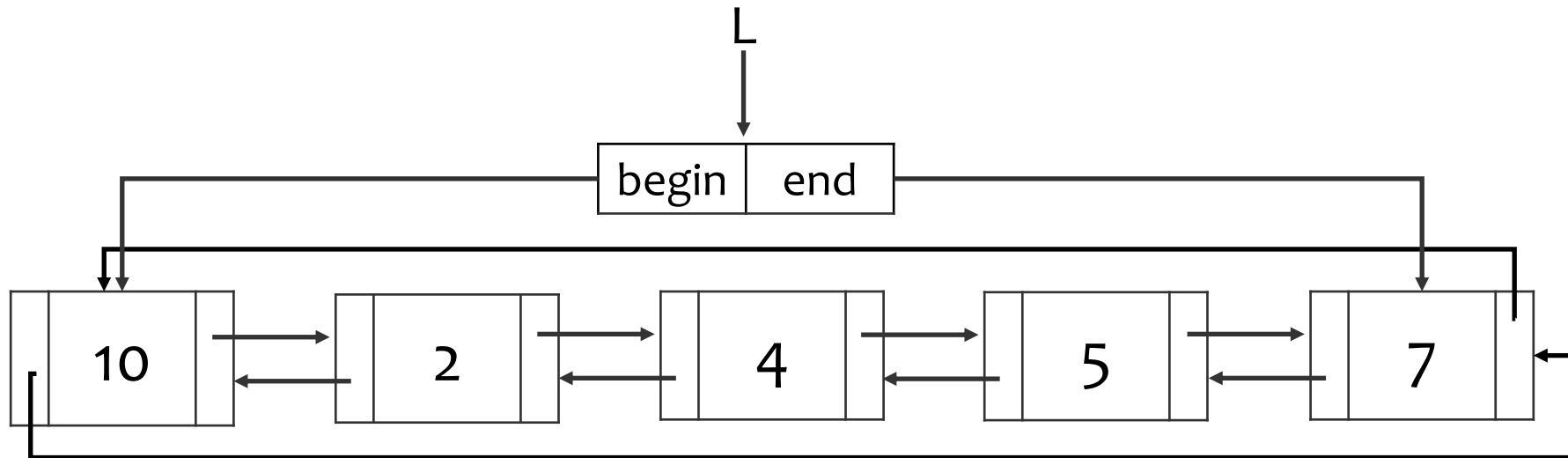


# Remoção de Elementos da Lista

**Caso 2:** Elemento está no **meio** da lista

- Idêntico à Lista Duplamente Encadeada

remover 4

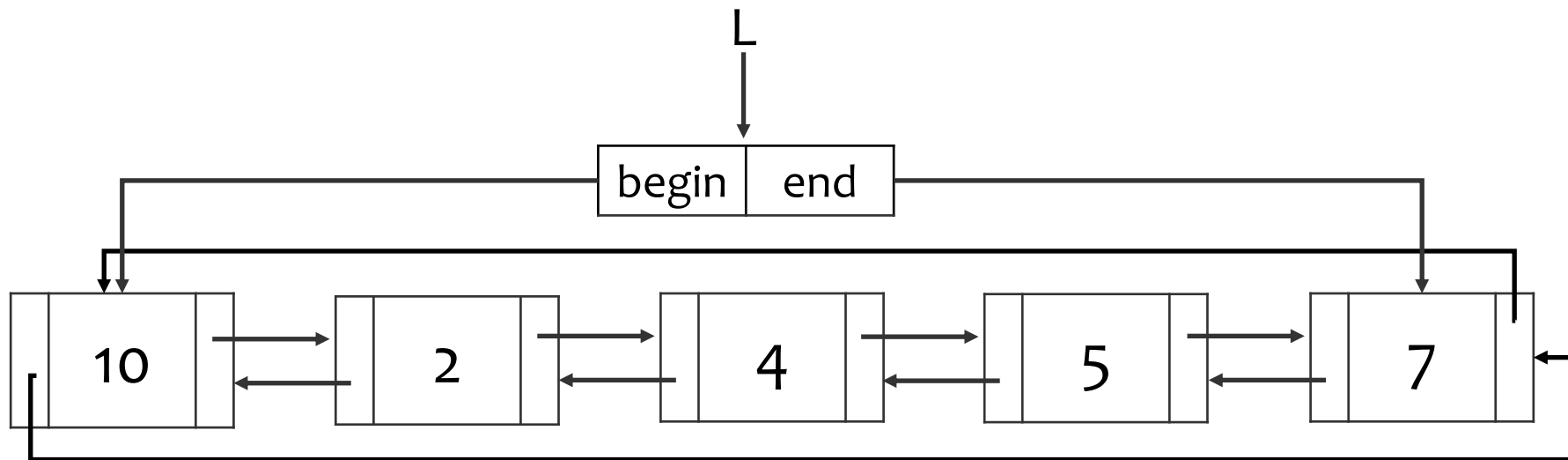


# Remoção de Elementos da Lista

**Caso 3:** Elemento está na **cauda (final)** da lista

- Mesmos passos da Lista Duplamente Encadeada, apenas atualizando o **encadeamento circular**

remover 7



# Lista Circular Duplamente Encadeada

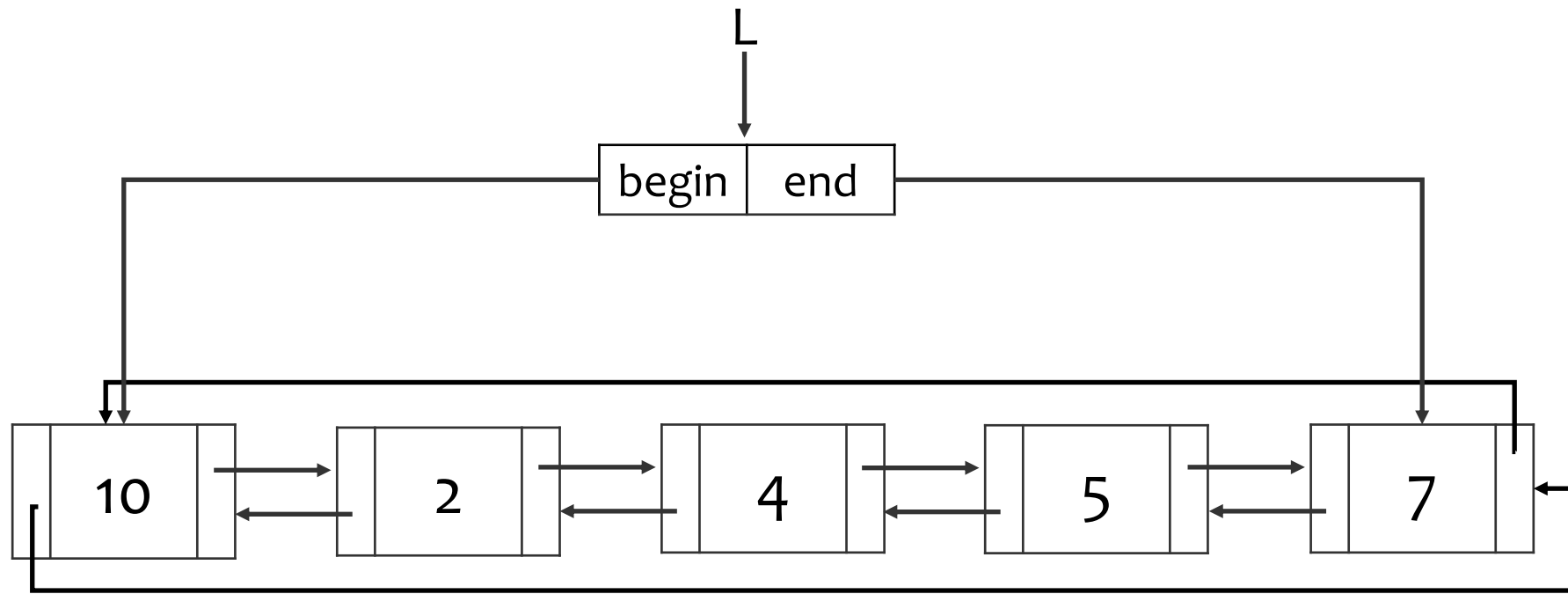
Diversos tipos de operações:

- Inserção na cabeça (início) da lista;
- Impressão dos Elementos da Lista
- Inserção na cauda (fim) da lista;
- **Remover elementos da lista;**
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i

# Lista Circular Duplamente Encadeada

Diversos tipos de operações:

- Inserção na cabeça (início) da lista;
- Impressão dos Elementos da Lista
- Inserção na cauda (fim) da lista;
- Remover elementos da lista;
- Contar o número de elementos da Lista;
- Verificar se a lista está vazia e retornar verdadeiro/falso
- Retornar o primeiro elemento;
- Retornar o último elemento;
- Retornar um elemento na posição i





- Inserir um elemento em uma dada posição
- Inverter uma lista
- Copiar/clonar uma lista;
- Apagar todos os elementos da lista
- Concatenar de duas listas;
- Ordenação de uma lista;
- Buscar um dado elemento na lista e retornar seu ponteiro;
- Inserção ordenada;

# Dominando Estruturas de Dados 1

## Listas Circulares

Prof. dr. Samuel Martins (Samuka)  
@xavecoding @hisamuka

