

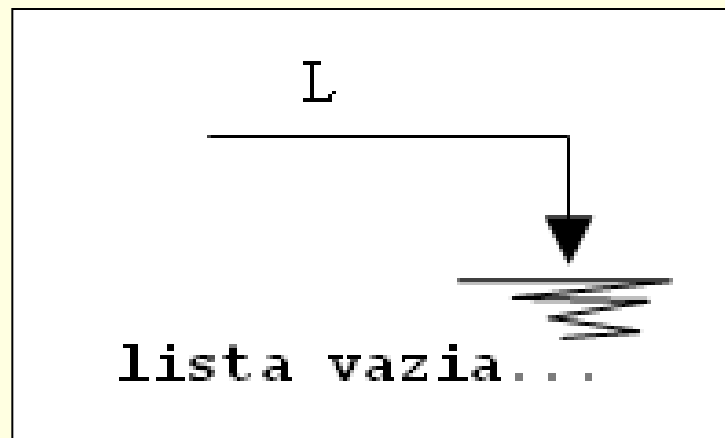
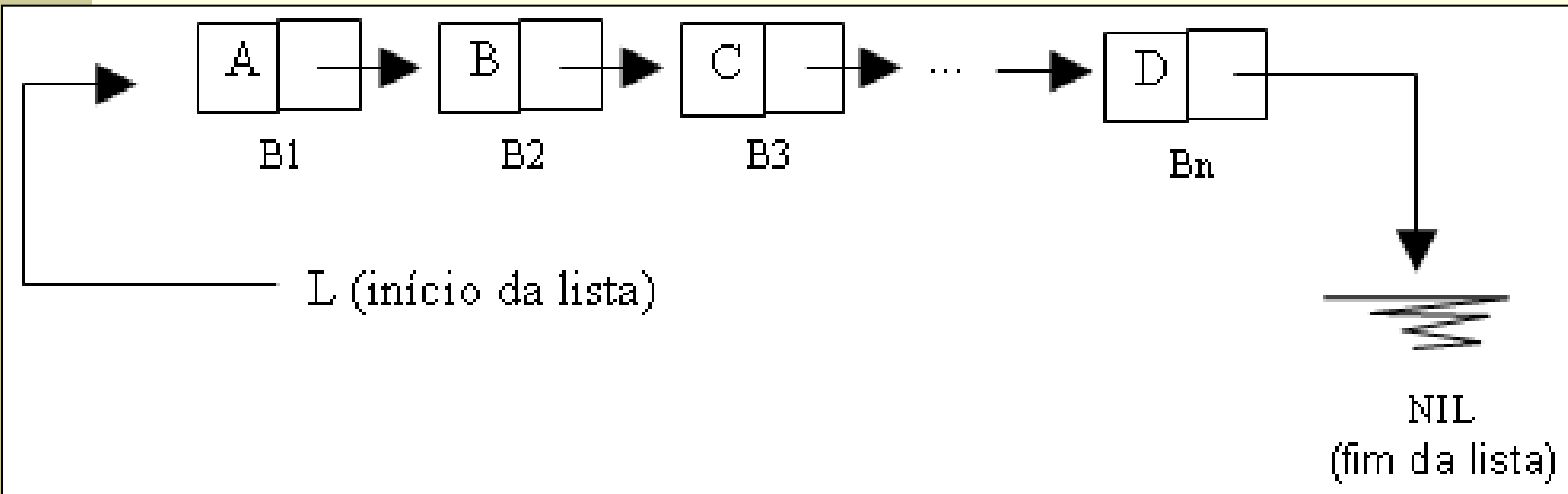
# Lista: conceito, representação e algoritmos

SCC0502 – Algoritmos e Estruturas de  
Dados I

# Listas encadeadas

- **Definição:** uma lista encadeada  $L$ , com  $n$  blocos de memória  $B_1, B_2, \dots, B_n$  é definida pelas seguintes características:
  - Cada bloco de memória  $B_i$ , ou cada “nó” da lista, tem pelo menos dois campos:
    - **Informação** a ser armazenada
    - Indicação do **próximo elemento** da lista
  - Os blocos de memória não estão necessariamente em sequência física na memória
  - O acesso aos elementos da lista ocorre por meio de um indicador do início da lista (o primeiro elemento); o acesso aos demais elementos ocorre por meio da indicação de quem é o próximo na sequência
  - O último nó da lista indica um endereço inválido, chamado NIL ou NULL

# Representação



# Lista

---

- Lista encadeada e dinâmica
  - Uma das representações mais interessantes e flexíveis que há
  - Aplicável para diversos problemas

# Declaração da lista

---

```
typedef char elem;
```

```
typedef struct bloco {  
    elem info;  
    struct bloco *prox;  
} no;
```

```
typedef struct {  
    no *inicio, *fim;  
} Lista;
```

# Operações genéricas sobre lista

---

- Considerando a representação de lista anterior, implemente o TAD lista com as seguintes operações
  - cria-lista
  - inserir-na-lista
  - finaliza-lista
  - tamanho
    - recursiva e não recursiva
  - esta-na-lista
    - recursiva e não recursiva
  - imprimir

# Operações genéricas sobre lista

---

- Considerando a representação de lista anterior, implemente o TAD lista com as seguintes operações
  - **eliminar-da-lista**
    - **recursiva e não recursiva**