

# ESTRUTURAS DE DADOS

---

## Aula 2 – Listas Estáticas Sequenciais

- Conceito
- Inserção
- Remoção
- Alteração

# Lembrando Ponteiros

- São tipos especiais de variáveis que armazenam o endereço da memória de outra variável
- A declaração de um ponteiro é parecido com a declaração de uma variável qualquer, porém ela contém um asterisco “ \* ” **entre** seu **tipo** e seu **identificador**

# Ponteiros

- São exemplos de ponteiros:

```
int *i;  
unsigned long int *l;  
char* s;  
float *f;  
void *v; //Ponteiro sem tipo  
int **i; //Ponteiro de Ponteiro
```

# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    *j = 4;  
    printf("%i", *j);  
    free(j);  
}
```

gerenciador de memória

| HEAP  |      |      |      |
|-------|------|------|------|
|       |      |      |      |
| 0xF1  | 0xF2 | 0xF3 | 0xF4 |
|       |      |      |      |
| 0xF5  | 0xF6 | 0xF7 | .... |
| STACK |      |      |      |
|       |      |      |      |
| 0x01  | 0x02 | 0x03 | 0x04 |
|       |      |      |      |
| 0x05  | 0x06 | 0x07 | .... |

# Ponteiros

```
void main() {  
»» int *j;  
   j=malloc(4);  
   *j = 4;  
   printf("%i", *j);  
   free(j);  
}
```

gerenciador de memória

| HEAP  |      |      |      |
|-------|------|------|------|
|       |      |      |      |
| 0xF1  | 0xF2 | 0xF3 | 0xF4 |
|       |      |      |      |
| 0xF5  | 0xF6 | 0xF7 | .... |
| STACK |      |      |      |
|       |      |      |      |
| 0x01  | 0x02 | 0x03 | 0x04 |
|       |      |      |      |
| 0x05  | 0x06 | 0x07 | .... |

# Ponteiros

```
void main() {  
»» int *j;  
   j=malloc(4);  
   *j = 4;  
   printf("%i", *j);  
   free(j);  
}
```

**Onde são armazenados os  
endereços que os  
ponteiros apontam?**

gerenciador de memória

| HEAP  |      |      |      |
|-------|------|------|------|
|       |      |      |      |
| 0xF1  | 0xF2 | 0xF3 | 0xF4 |
|       |      |      |      |
| 0xF5  | 0xF6 | 0xF7 | .... |
| STACK |      |      |      |
|       |      |      |      |
| 0x01  | 0x02 | 0x03 | 0x04 |
|       |      |      |      |
| 0x05  | 0x06 | 0x07 | .... |

# Ponteiros

```
void main() {  
»» int *j;  
   j=malloc(4);  
   *j = 4;  
   printf("%i", *j);  
   free(j);  
}
```

**NA STACK!**

gerenciador de memória

| HEAP    |      |      |      |
|---------|------|------|------|
|         |      |      |      |
| 0xF1    | 0xF2 | 0xF3 | 0xF4 |
|         |      |      |      |
| 0xF5    | 0xF6 | 0xF7 | .... |
| STACK   |      |      |      |
| J: lixo |      |      |      |
| 0x01    | 0x02 | 0x03 | 0x04 |
|         |      |      |      |
| 0x05    | 0x06 | 0x07 | 0x08 |

# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    *j = 4;  
    printf("%i", *j);  
    free(j);  
}
```

gerenciador de memória

| HEAP    |      |      |      |
|---------|------|------|------|
|         |      |      |      |
| 0xF1    | 0xF2 | 0xF3 | 0xF4 |
|         |      |      |      |
| 0xF5    | 0xF6 | 0xF7 | .... |
| STACK   |      |      |      |
| J: lixo |      |      |      |
| 0x01    | 0x02 | 0x03 | 0x04 |
|         |      |      |      |
| 0x05    | 0x06 | 0x07 | 0x08 |



# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    *j = 4;  
    printf("%i", *j);  
    free(j);  
}
```

**“Solicito 4 Bytes da memória”**

gerenciador de memória

| HEAP    |      |      |      |
|---------|------|------|------|
|         |      |      |      |
| 0xF1    | 0xF2 | 0xF3 | 0xF4 |
|         |      |      |      |
| 0xF5    | 0xF6 | 0xF7 | .... |
| STACK   |      |      |      |
| J: lixo |      |      |      |
| 0x01    | 0x02 | 0x03 | 0x04 |
|         |      |      |      |
| 0x05    | 0x06 | 0x07 | 0x08 |

# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    *j = 4;  
    printf("%i", *j);  
    free(j);  
}
```

**ALOCADO!!**  
**Posição 0xF1**

gerenciador de memória

| HEAP                            |      |      |      |
|---------------------------------|------|------|------|
| .....LIXO.....LIXO.....LIXO.... |      |      |      |
| 0xF1                            | 0xF2 | 0xF3 | 0xF4 |
|                                 |      |      |      |
| 0xF5                            | 0xF6 | 0xF7 | .... |
| STACK                           |      |      |      |
| J: lixo                         |      |      |      |
| 0x01                            | 0x02 | 0x03 | 0x04 |
|                                 |      |      |      |
| 0x05                            | 0x06 | 0x07 | 0x08 |

# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    *j = 4;  
    printf("%i", *j);  
    free(j);  
}
```

**ALOCADO!!**  
**Posição 0xF1**

gerenciador de memória

| HEAP                            |      |      |      |
|---------------------------------|------|------|------|
| .....LIXO.....LIXO.....LIXO.... |      |      |      |
| 0xF1                            | 0xF2 | 0xF3 | 0xF4 |
|                                 |      |      |      |
| 0xF5                            | 0xF6 | 0xF7 | .... |
| STACK                           |      |      |      |
| J: 0xF1                         |      |      |      |
| 0x01                            | 0x02 | 0x03 | 0x04 |
|                                 |      |      |      |
| 0x05                            | 0x06 | 0x07 | 0x08 |

# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    >>> *j = 4;  
    printf("%i", *j);  
    free(j);  
}
```

gerenciador de memória

| HEAP                            |      |      |      |
|---------------------------------|------|------|------|
| .....LIXO.....LIXO.....LIXO.... |      |      |      |
| 0xF1                            | 0xF2 | 0xF3 | 0xF4 |
|                                 |      |      |      |
| 0xF5                            | 0xF6 | 0xF7 | .... |
| STACK                           |      |      |      |
| J: 0xF1                         |      |      |      |
| 0x01                            | 0x02 | 0x03 | 0x04 |
|                                 |      |      |      |
| 0x05                            | 0x06 | 0x07 | 0x08 |

# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    >>> *j = 4;  
    printf("%i", *j);  
    free(j);  
}
```

**“Navegue” até 0xF1  
e escreva “4” neste endereço**

gerenciador de memória

| HEAP                            |      |      |      |
|---------------------------------|------|------|------|
| .....LIXO.....LIXO.....LIXO.... |      |      |      |
| 0xF1                            | 0xF2 | 0xF3 | 0xF4 |
|                                 |      |      |      |
| 0xF5                            | 0xF6 | 0xF7 | .... |
| STACK                           |      |      |      |
| J: 0xF1                         |      |      |      |
|                                 | 0x02 | 0x03 | 0x04 |
|                                 |      |      |      |
|                                 | 0x06 | 0x07 | 0x08 |

# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    >>> *j = 4;  
    printf("%i", *j);  
    free(j);  
}
```

gerenciador de memória

| HEAP    |      |      |      |
|---------|------|------|------|
| 4       |      |      |      |
| 0xF1    | 0xF2 | 0xF3 | 0xF4 |
|         |      |      |      |
| 0xF5    | 0xF6 | 0xF7 | .... |
| STACK   |      |      |      |
| J: 0xF1 |      |      |      |
| 0x01    | 0x02 | 0x03 | 0x04 |
|         |      |      |      |
| 0x05    | 0x06 | 0x07 | 0x08 |

# Ponteiros

```
void main() {  
    int *j;  
    j=malloc(4);  
    *j = 4;  
    >>> printf("%i", *j);  
    free(j);  
}
```

**Imprima o conteúdo do endereço 0xF1**

gerenciador de memória

| HEAP    |      |      |      |
|---------|------|------|------|
| 4       |      |      |      |
| 0xF1    | 0xF2 | 0xF3 | 0xF4 |
|         |      |      |      |
| 0xF5    | 0xF6 | 0xF7 | .... |
| STACK   |      |      |      |
| J: 0xF1 |      |      |      |
| 0x01    | 0x02 | 0x03 | 0x04 |
|         |      |      |      |
| 0x05    | 0x06 | 0x07 | 0x08 |

# LISTAS LINEARES

---

Prof. Guilherme Alberto Wachs Lopes



# Listas Lineares

- Sequencia de 0 ou mais elementos do mesmo tipo **que possuem uma relação de ordem**:
  - $a_i$  precede  $a_{i+1}$
  - $a_0$  é o primeiro elemento da lista
  - $a_{n-1}$  é o último elemento da lista
- Operações básicas:
  - Criar e iniciar uma lista;
  - Inserir e retirar um  $i$ -ésimo elemento.
  - Buscar um  $i$ -ésimo elemento.

# Listas Lineares

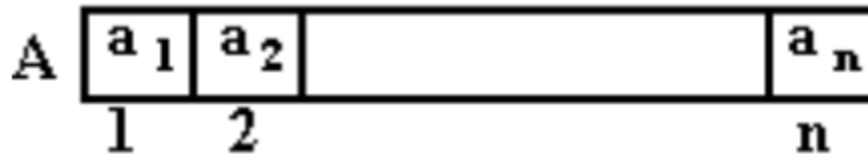
- Uma lista pode ser:

- **Simplesmente Ligada:** A partir de um elemento da lista não se alcança o elemento anterior.
- **Duplamente Ligada:** A partir de um elemento da lista se alcança o elemento anterior.
- **Circular:** A partir do primeiro elemento da lista se alcança o último. E a partir do último elemento da lista se alcança o primeiro.
- **Não circular:** A partir do primeiro não se alcança o último. E do último elemento não se alcança o primeiro.

# Tipos de Listas Lineares

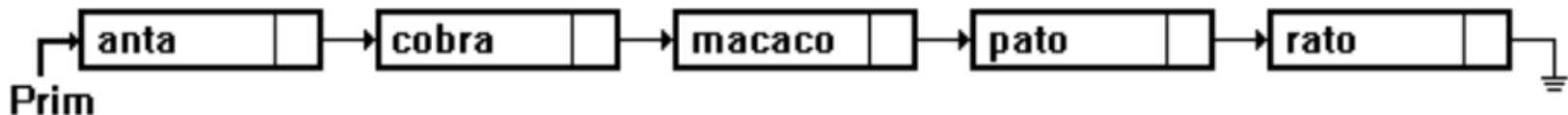
## ■ Estáticas

- Os elementos são armazenados em um vetor.



## ■ Dinâmicas

- Os elementos são alocados dinamicamente conforme a necessidade. Cada elemento armazena os dados E um PONTEIRO para o próximo elemento da lista.



# Motivação para listas dinâmicas

- A Estrutura vetor é estática e nem sempre eu sei qual é o número máximos de elementos que eu posso ter
- Desperdício de memória pois normalmente utilizamos valores muito altos para o tamanho do vetor

# Tipos de Listas Lineares

- Estrutura de dados do Tipo Listas:
  - Lista Estática Sequencial (LES)
  - Lista Estática Encadeada (LEE)
  - Lista Dinâmica Encadeada (LDE)
  - Lista Dinâmica Duplamente Encadeada (LDDE)

# Definições

## ■ Seja

- $L$  um lista com  $n$  elementos
- $i$  um índice da lista tal que  $0 \leq i \leq n - 1$

# Listas Estáticas Sequenciais (LES)

## ■ Características das listas LES

- Os elementos da lista estão ordenados através de um campo chave;
- São armazenados fisicamente em posições consecutivas
- A inserção de um elemento na posição  $L[i]$  causa o deslocamento a direita do elemento de  $L[i]$  ao último elemento
- A eliminação do elemento  $L[i]$  requer o deslocamento à esquerda do elemento  $L[i + 1]$  ao último;

# Listas Estáticas Sequenciais (LES)

## ■ Características das listas LES

- Os elementos da lista estão ordenados através de um campo chave;
- São armazenados fisicamente em posições consecutivas
- A inserção de um elemento na posição  $L[i]$  causa o deslocamento a direita do elemento de  $L[i]$  ao último elemento
- A eliminação do elemento  $L[i]$  requer o deslocamento à esquerda do elemento  $L[i + 1]$  ao último;



# Listas Estáticas Sequenciais (LES)

## ■ Operações Básicas

- Inserção
- Remoção
- Busca

# LISTAS ESTÁTICAS SEQUENCIAIS

---

Prof. Guilherme Alberto Wachs Lopes

# Listas Lineares Estáticas

## ■ Vetores



## ■ Criação:

- Declarar um vetor de tamanho  $n$
- Declarar uma variável que informa a quantidade de itens na lista ( $n$ )

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inicialização

```
void main() {  
  
    char vetor[10];  
  
    int n = 0;  
  
}
```

| STACK |      |      |      |
|-------|------|------|------|
|       |      |      |      |
| 0x01  | 0x02 | 0x03 | 0x04 |
|       |      |      |      |
| 0x05  | 0x06 | 0x07 | 0x08 |
|       |      |      |      |
| 0x09  | 0x0A | 0x0B | 0x0C |

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inicialização

```
void main() {  
  
    char vetor[10];  
  
    int n = 0;
```

**Alocar 10 Bytes  
na Ram**

| STACK |      |      |      |
|-------|------|------|------|
| Lixo  | Lixo | Lixo | Lixo |
| 0x01  | 0x02 | 0x03 | 0x04 |
| Lixo  | Lixo | Lixo | Lixo |
| 0x05  | 0x06 | 0x07 | 0x08 |
| Lixo  | Lixo |      |      |
| 0x09  | 0x0A | 0x0B | 0x0C |

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 0$$

$L =$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |   |   |   |

Como inserir o número 25?

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 0$$

$L =$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |   |   |   |

Como  $n = 0$ , a lista está vazia. Então, vamos inserir na posição  $n$

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1$$

$L =$

| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 25 |   |   |   |   |   |   |   |   |   |

Como  $n = 0$ , a lista está vazia. Então, vamos inserir na posição  $n$



# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1$$

$L =$

| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 25 |   |   |   |   |   |   |   |   |   |

Como inserir o número 32?

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1$$

$L =$

| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 25 |   |   |   |   |   |   |   |   |   |

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento
2. Inserir o elemento
3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1$$

$L =$

| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 25 |   |   |   |   |   |   |   |   |   |

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1$$

$L =$

| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 25 |   |   |   |   |   |   |   |   |   |



É Aqui?

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1$$

$L =$

| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 25 |   |   |   |   |   |   |   |   |   |



É Aqui?  
NÃO!!

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1$$

$L =$

| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 25 |   |   |   |   |   |   |   |   |   |



É Aqui?

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1$$

$L =$

| 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|
| 25 |   |   |   |   |   |   |   |   |   |



É Aqui?  
SIM!!

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inserção

$$n = 1$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |

A lista não está vazia, então:  
2. Inserir o elemento



# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 1 + 1$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |

A lista não está vazia, então:

3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |

A lista não está vazia, então:

3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |

Como inserir o número 27?

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento
2. Inserir o elemento
3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |



É Aqui?

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |



É Aqui?  
NÃO!!

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |



É Aqui?

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento



# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 32 |   |   |   |   |   |   |   |   |



É Aqui?  
SIM!!

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inserção

$$n = 2$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

A lista não está vazia, então:  
2. Inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

A lista não está vazia, então:

3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 2 + 1$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

A lista não está vazia, então:

3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

A lista não está vazia, então:

3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

Como inserir o número 1?

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inserção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |



É Aqui?

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento



# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |



É Aqui?  
SIM!!!

A lista não está vazia, então:

1. Procurar a posição para inserir o elemento

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inserção

$$n = 3$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

A lista não está vazia, então:

2. Inserir o elemento

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Inserção

$$n = 3 + 1$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

A lista não está vazia, então:

3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inserção

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

A lista não está vazia, então:

3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inserção

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

A lista não está vazia, então:

3.  $n = n + 1$

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Inserção

$n =$

$L =$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |   |   |   |

Mais Exemplos em SALA

# Listas Estáticas Sequenciais (LES)

- Inserção: (Lembrete...)
  - Caso 1: Inserir primeiro elemento em uma lista vazia:
    - Checar a quantidade, se for Zero adicionar no índice 0 da lista.
  - Caso 2: Inserir primeiro elemento de uma lista não vazia:
    - Checar a quantidade, se for maior que Zero, então desloque todos os itens da lista um item para a direita e adicione o item novo no índice 0 da lista.
  - Caso 3: Inserir o último elemento de uma lista:
    - Quando chegar ao final da lista (atual igual quantidade). Adicione o item novo no índice quantidade da lista (última posição).
  - Caso 4: Inserir no meio da lista:
    - Quando encontra a posição de inserção do novo elemento. Desloque todos os elementos do final da lista até a posição de inserção um índice para direita e adicione o novo item na posição correta.

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Busca

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

Como fazer uma consulta na Lista?



# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Busca

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

Como fazer uma consulta na Lista?

- Busca Linear

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Busca

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

Como fazer uma consulta na Lista?

- Busca Linear
- Busca Binária

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Busca

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

Como fazer a remoção de um elemento?

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Busca

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

Remover o item 1

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Busca

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

Remover o item 1

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Remoção

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |

A lista não está vazia, então:

1. Procurar o elemento a ser removido
2. Caso elemento exista remova-o da lista
  - a.  $n = n - 1$
3. Caso contrário, erro.

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Remoção

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |



Busca retorna  
posição 0

A lista não está vazia, então:

1. Procurar o elemento a ser removido

# Listas Estáticas Sequenciais (LES)


## ■ Funcionamento

- Remoção

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
| 1 | 25 | 27 | 32 |   |   |   |   |   |   |



A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista

a.  $n = n - 1$



# Listas Estáticas Sequenciais (LES)


## ■ Funcionamento

- Remoção

$$n = 4$$

$L =$

| 0 | 1  | 2  | 3  | 4 | 5 | 6 | 7 | 8 | 9 |
|---|----|----|----|---|---|---|---|---|---|
|   | 25 | 27 | 32 |   |   |   |   |   |   |



A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista

a.  $n = n - 1$


# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Remoção

$$n = 4$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |



A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista
  - a.  $n = n - 1$

# Listas Estáticas Sequenciais (LES)


## ■ Funcionamento

- Remoção

$$n = 4 - 1$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |



A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista

a.  $n = n - 1$

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Remoção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista
  - a.  $n = n - 1$

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Remoção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

Remover o item 32

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Remoção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |

A lista não está vazia, então:

1. Procurar o elemento a ser removido
2. Caso elemento exista remova-o da lista
  - a.  $n = n - 1$
3. Caso contrário, erro.

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Remoção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |



Busca retorna  
posição 2

A lista não está vazia, então:

1. Procurar o elemento a ser removido

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Remoção

$$n = 3$$

$L =$

| 0  | 1  | 2  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|----|---|---|---|---|---|---|---|
| 25 | 27 | 32 |   |   |   |   |   |   |   |



A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista

a.  $n = n - 1$



# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Remoção

$$n = 3$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 27 |   |   |   |   |   |   |   |   |



A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista

a.  $n = n - 1$

# Listas Estáticas Sequenciais (LES)

## ■ Funcionamento

- Remoção

$$n = 3 - 1$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 27 |   |   |   |   |   |   |   |   |



A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista

a.  $n = n - 1$

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Remoção

$$n = 2$$

$L =$

| 0  | 1  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|----|---|---|---|---|---|---|---|---|
| 25 | 27 |   |   |   |   |   |   |   |   |

A lista não está vazia, então:

2. Caso elemento exista, remova-o da lista
  - a.  $n = n - 1$

# Listas Estáticas Sequenciais (LES)

- Funcionamento
  - Remoção

$n =$

$L =$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |   |   |   |

Mais Exemplos em SALA

# Lista Estática Sequencial

- Pense no algoritmo de Inserção
- Escreva um programa em C para Inserção

## Lista Estática Sequencial

- Pense no algoritmo de Remoção
- Escreva um programa em C para Remoção

# Lista Estática Sequencial

- Implemente um programa em C que utiliza a estrutura apresentada abaixo para implementar uma lista. O programa deve mostrar ao usuário duas opções:
  - Se usuário escolher 1, a lista deve ser impressa;
  - se escolher 2, ele deve entrar com o valor do conteúdo do novo elemento da lista.

| lista          |
|----------------|
| meuTipo v[N];  |
| int elementos; |

| meuTipo            |
|--------------------|
| char nome[35];     |
| char telefone[15]; |