

# Laboratório de Programação II

## Listas Contíguas

Universidade Federal de Juiz de Fora  
Departamento de Ciência da Computação

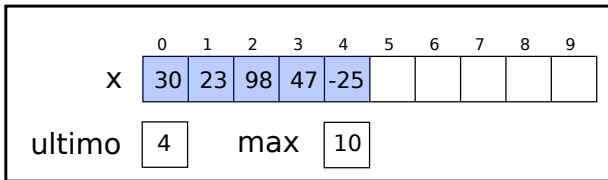
# Aula de Hoje

- ▶ Listas
  - ▶ **Contíguas**
  - ▶ Encadeadas

# Lista Contígua

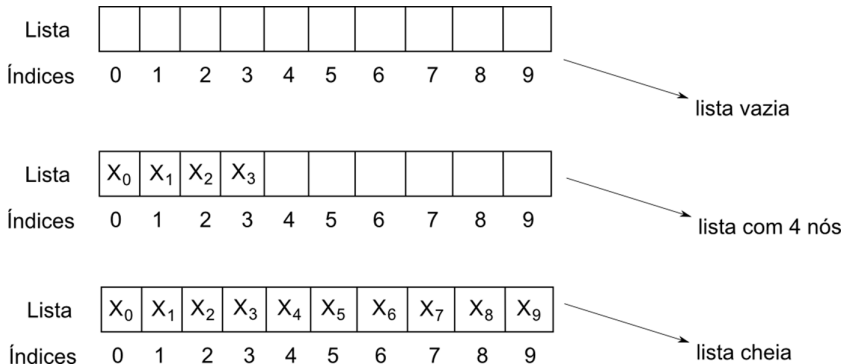
- ▶ Exemplo de uma lista de inteiros com representação por contiguidade dos nós, isto é, os nós são armazenados em um vetor `x` de inteiros com capacidade máxima definida por `max`.
- ▶ Para saber o tamanho atual da lista (e a posição do último nó), um inteiro chamado aqui de `ultimo` é usado.

## ListaCont



# Lista Contígua

## ► Exemplos



# Lista Contígua

- ▶ As listas podem armazenar elementos de **qualquer tipo de dados**, isto é, podem ser do tipo `int`, `float`, `double`, `bool` ou até mesmo algum tipo definido pelo programador.
- ▶ Vamos mostrar um **exemplo** de uma lista para armazenar elementos do tipo `float` em C++.

# Lista Contígua

```
class ListaCont
{
private:
    int max;
    int ultimo;
    float *x;

public:
    ListaCont(int tam);
    ~ListaCont();
    float get(int k);
    void set(int k, float val);
    void insereK(int k, float val);
    void inserePrim(float val);
    void insereUlt(float val);
    void eliminaK(int k);
    void eliminaPrim();
    void eliminaUlt();
};
```

# Exercícios

0. Faça uma leitura da implementação deste TAD, a qual se encontra no arquivo `ListaCont.cpp`.
1. Implemente as seguintes operações:
  - ▶ Imprimir a lista;
  - ▶ Verificar se existe um elemento na lista com valor maior do que um valor fornecido. Retornar o índice do primeiro elemento que seja maior que o dado de entrada ou -1 caso contrário.
  - ▶ Retornar o número de nós de uma lista;
  - ▶ “Limpar” a lista, isto é, remover todos os elementos.

Utilize os seguintes protótipos:

```
void ListaCont::imprime();  
int  ListaCont::buscaMaior(float val);  
int  ListaCont::numNos();
```

## Exercícios

2. Faça uma operação que dados um vetor de números reais e o seu tamanho, verifica e insere (se possível) esses elementos no final da lista. Utilize o seguinte protótipo:

```
void ListaCont::insereValores(int tam,  
                             float vet[]);
```

Teste a implementação dessa operação no programa principal.

3. Faça um programa que crie 3 listas contíguas L1, L2 e L3. Inicialize L1 com 50 valores quaisquer até que a lista esteja cheia. Em seguida, retire a primeira metade de L1 e guarde em L2. Faça o mesmo com a segunda metade, mas guarde em L3. Imprima L2 e L3.

**Observação:** Utilize apenas as operações de inserção e remoção do TAD `ListaCont`.



## Exercícios

4. Modifique o TAD `ListaCont` para **armazenar pontos** (TAD `Ponto`) ao invés de números reais. Em seguida, faça um programa que crie vários pontos e **imprima uma tabela de distâncias** entre todos os pontos.