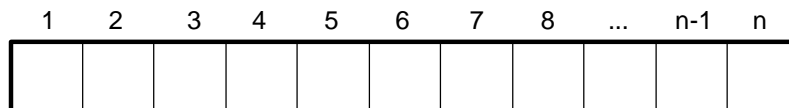


Listas Lineares

- São estruturas que permitem representar um conjunto de dados de forma a preservar a relação de ordem linear entre eles.
- São compostas de nós que podem conter tanto tipos de dados primitivos quanto construídos.
- Operações:
 - acessar determinado nó da lista;
 - inserir um nó em determinada posição da lista;
 - remover determinado nó da lista;
 - concatenar duas listas;
 - determinar o número de nós de uma lista;
 - localizar determinado dado dentro da lista;

Listas Lineares (cont.)



- são implementadas através de variáveis estáticas;
- o acesso aos seus componentes é feito de maneira direta (através de um índice, por exemplo);
- as estruturas de representação que melhor permitem a implementação de listas lineares seqüenciais (ou contíguas) são os *arrays* e arquivos.
- **Vantagem:** possibilidade de implementação em quase todas as linguagens;
- **Desvantagens:** Dimensão fixa e necessidade de reorganização constante.

Listas Lineares (Exemplo)

1	Aarão
2	Carlos
3	Célia
4	Dalva
5	Elídio
6	Fábio
7	Francisco
8	Gustavo
9	Hélio
10	Iria
11	Joaquim
12	José
13	Kátia
....	Zaqueu
n-1	Zulmira
n	

- A lista ao lado contém um arranjo de nomes em ordem alfabética.
- A inclusão de um novo item (por ex. *Benedito*), levaria a uma reorganização da lista.
- Com a inclusão de mais um item, a lista estará “cheia”.

Listas Lineares (manipulação)

(Criação)

```
const tam_maximo = 100;
type lista = record
    item : array[1..tam_maximo] of integer;
    topo : 0..tam_maximo;
end;

var l : lista;
```

(Vazia - verifica se lista está vazia)

```
function Vazia(l : lista) : boolean;
begin
    if l.topo = 0 then
        empty := TRUE
    else
        empty := FALSE;
    end; {função Vazia}
```

Listas Lineares (manipulação)

(Insere - Insere novos itens na lista em uma determinada posição)

```
Function Insere(var l:lista; item:integer;
  i:integer):boolean;
var
  j:integer;
begin
  if (l.topo>tam_maximo) then
    Insere=FALSE; {lista cheia}
  else
    begin
      Insere=TRUE;
      if (i>l.topo) then
        begin
          l.topo=l.topo+1;
          l.item[l.topo]=item;
        end;
      else
        begin
          l.topo=l.topo+1;
          for j:=l.topo downto (i+1) do
            l.item[j]=l.item[j-1];
          l.item[i]=item;
        end;
      end;
    end;
end; {função Insere}
```

Listas Lineares (manipulação)

(Recupera - Lê o valor de uma posição determinada da lista)

```
Function Recupera(l:lista; i:integer):inteiro;  
begin  
  if Vazia(l) then  
    writeln("Lista Vazia");  
  else  
    if (i>l.topo) then  
      writeln("Item não faz parte da lista");  
    else  
      Recupera = l.item[i];  
    end;  
  end; {função Insere}
```

Listas Lineares (manipulação)

(Apaga - Apaga itens na lista em uma determinada posição)

```
Function Apaga(var l:lista; i:integer):
    boolean;
var
    j:integer;
begin
    if (i>tam_maximo) then
        begin
            Apaga=FALSE; {fora do escopo da lista}
            writeln("Este índice não existe na lista")
        end;
    else
        if (i>l.topo) then
            begin
                Apaga=FALSE
                writeln("Este índice não consta na
                    lista");
            end;
        else
            begin
                l.topo=l.topo-1;
                for j:=i to l.topo do
                    l.item[j]=l.item[j+1];
                end;
            end;
    end; {função Apaga}
```