

# Filas com Listas Circulares

- **Dificuldade** principal das filas com listas lineares: Reorganização na retirada de elementos
- **Alternativa:** Deslocar o início da lista na remoção:



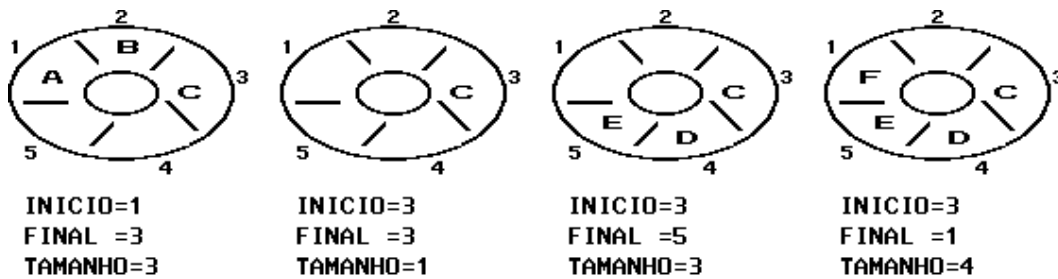
- **Inserção:** Feita no fim da fila (sem mudanças)
- **Remoção:** Feita somente mudando a posição do início da fila (sem necessidade de reorganização)  

```
f.inicio:=f.inicio+1;
```
- **Fila Vazia:**  

```
f.topo < f.inicio
```
- **Dificuldade:** a cada retirada de elemento, a lista diminui de tamanho.
- **Possível Solução:** utilizar as posições iniciais, desde que exista espaço disponível.

# Filas com Listas Circulares (cont.)

- A **solução** para este problema é enxergar o array como se fosse **circular**, ao invés de linear.



- Fila Vazia:** não se pode utilizar o critério  
 $f.topo < f.início$
- Opção:** utilizar uma variável que armazene o **tamanho** da fila:
  - Inserção:**  $tamanho := tamanho + 1;$
  - Remoção:**  $tamanho := tamanho - 1;$
- Fila Vazia:**  $tamanho = 0$
- Fila Cheia:**  $tamanho = tam\_max;$

# Filas com Listas Circulares (manipulação)

- **Criação da lista:**

```
const máximo = 100;
type fila = record
  itens : array[1..máximo] of integer;
  inicio, final, tamanho : 0..máximo;
end;
```

```
var f : fila;
```

- **Verificação de lista vazia:**

```
function vazia(f:fila) : boolean;
begin
  if f.tamanho = 0 then
    vazia := TRUE
  else
    vazia := FALSE;
end;
```

# Filas com Listas Circulares (manipulação)

- **Inserção na lista:**

```
procedure insere(var f:fila;
  x:integer; over:boolean);
begin
  if (f.tamanho = MÁXIMO) then
    over := TRUE    {OVERFLOW}
  else
    begin
      over := FALSE; {Existe espaço}
      if (f.final = máximo) then
        f.final := 1 {Estrutura circ.}
      else
        f.final := f.final + 1;
      f.itens[f.final] := x;
      f.tamanho := f.tamanho + 1;
    end;
  end;
```

# Filas com Listas Circulares (manipulação)

- **Inserção na lista:**

```
procedure remove(var f:fila;var
  x:integer;var und:boolean);
begin
  if f.tamanho = 0 then
    und := TRUE  {underflow}
  else
    begin
      und := FALSE;
      x := f.itens[f.inicio];
      f.tamanho := f.tamanho - 1;
      if f.inicio = maximo then
        f.inicio := 1
      else
        f.inicio := f.inicio + 1;
      end;
    end;
end;
```