

Fila

26 e 31/8/2008

Representação/Implementação:

Seqüencial estática

Encadeada dinâmica

Exercícios/Aplicações

Fila

- O que é?
- Para que serve?



Modelos intuitivos de filas

- Linhas para comprar bilhetes de cinema e de caixa de supermercado.

Situação: Filas de Supermercado

Entrar na fila

Sair da Fila

Verificar se há fila vazia

Buscar a fila de tamanho menor

Verificar se não há mais atendimento para aquela fila (pode ser entendido como fila cheia)

A fila, como a pilha, é conceitualmente uma estrutura **dinâmica** que está continuamente mudando, pois itens são adicionados/retirados.

Problema: automação de uma biblioteca

- Todos os livros devem ser **cadastrados**
 - O sistema deve informar se um livro está **disponível ou não** nas estantes
 - Caso o livro não esteja disponível, o usuário pode aguardar em uma **fila de espera**
 - Quando o livro for devolvido, o primeiro da fila de espera pode retirá-lo
-
- Sua tarefa: desenvolver esse sistema
 - **Só empreste livros para pessoas cadastradas!!!!**

Biblioteca

- 1^o passo: abstração
 - Identificar os elementos do mundo real que são relevantes para a solução do problema

Biblioteca

■ Quais são eles?

fila de espera para o livro	livros do acervo	disponível?
 <p>último ---> <--- 1º</p>	 <p>trigonometria</p>	não
 <p>último ---> <--- 1º</p>	 <p>química inorgânica</p>	não
fila vazia!	 <p>estruturas de dados</p>	sim

Biblioteca

- Elementos relevantes
 - Um cadastro de livros
 - Indicação da disponibilidade dos livros
 - Uma fila de espera para cada livro, com indicação da ordem das pessoas
 - Primeiro e último da fila
 - Cadastro de pessoas: nome, endereço e telefone

Biblioteca

- 2º passo: quais são as operações possíveis?
 - Entrar na fila
 - Quem entra, entra onde?
 - Sair da fila
 - Quem sai, sai de onde?
 - Outras?
- Implementaremos após o desenvolvimento do **TAD Fila**

Fila (*queue*)

- O que é?
 - É uma estrutura para armazenar um conjunto de elementos, que funciona da seguinte forma:
 - Novos elementos sempre **entram** no **fim** da fila
 - O único elemento que se pode **retirar** da fila em um dado momento é seu **primeiro** elemento
- Para que serve?
 - Modelar situações em que é preciso **armazenar um conjunto ordenado** de elementos,
 - no qual o primeiro elemento a entrar no conjunto será também o primeiro elemento a sair do conjunto, e assim por diante.
- FIFO ou LILO
 - *First In, First Out OU Last in, Last Out*

Aplicações de fila

- Biblioteca
 - Lista de espera para livros
- Impressão
 - Documentos a serem impressos
- Aeroporto
 - Lista de espera para vôos
- Saúde
 - Lista de espera de um órgão humano
- Outras?

TAD FILA – operações usuais

- **Cria(F):** cria uma fila F vazia
- **Entra(F,X):** X entra no fim da fila F
- **Sai(F,X):** o primeiro elemento da fila F é retirado da fila e atribuído a X
- **Y=Vazia(F):** verdade se a fila estiver vazia; caso contrário, falso
- **Y=Cheia(F):** verdade se a fila estiver cheia; caso contrário, falso

Outras rotinas?

- ❑ $Y = \text{Tamanho}(F)$: Retorna o tamanho da fila F
- ❑ $\text{Começo_Fila}(F, X)$: Mostra o começo da fila F sem remover o item.
- ❑ $\text{Tornar_Vazia}(F)$: Reinicializa uma fila existente F como uma fila vazia.
- ❑ $\text{Destroi}(F)$: Remove a fila criada da memória

Representação/Implementação da fila

- Representação **seqüencial**
 - Os elementos da fila ficam, necessariamente, em seqüência (um ao lado do outro) na memória
- Implementação **estática**
 - Todo o espaço de memória a ser utilizado pela fila é reservado (alocado) em tempo de compilação
 - Todo o espaço reservado permanece reservado durante todo o tempo de execução do programa, independentemente de estar sendo efetivamente usado ou não

Uso de Vetores para Seqüencial/Estática??

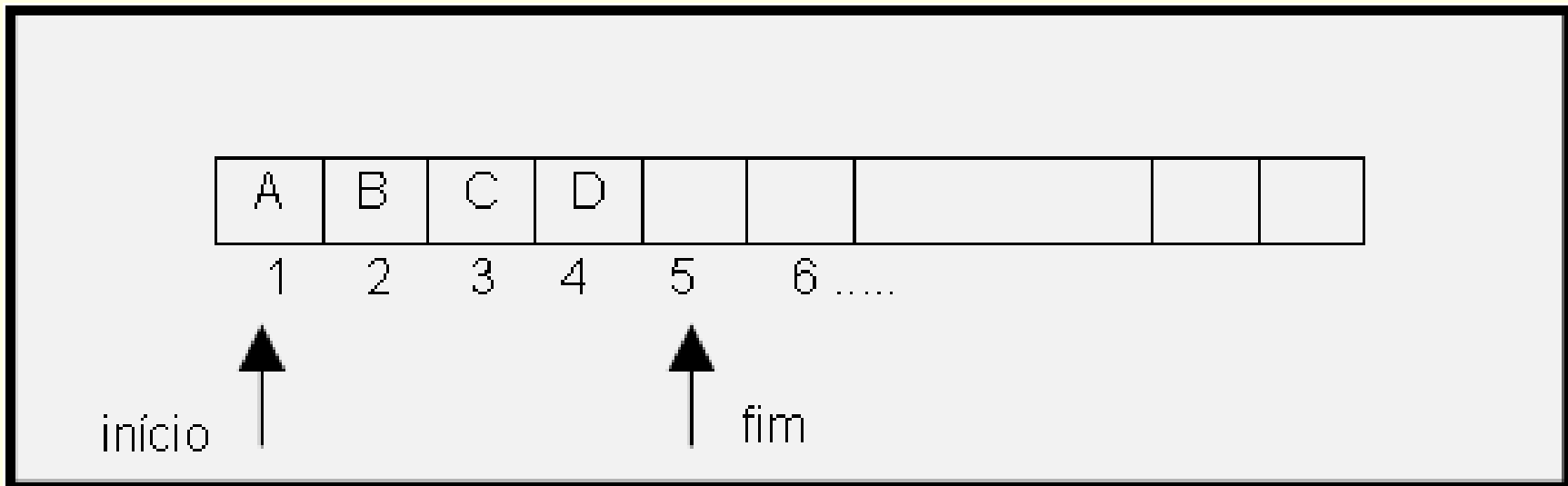
- Uma fila é uma lista de itens. Um array em C também é. Uma fila pode ser declarada simplesmente como um array???
- A resposta é NÃO!
 - Desde que um array é um objeto de tamanho fixo e a **fila** (como também a **pilha**) é **dinâmica**.
- Há um meio de se utilizar de um array na implementação de uma fila?

Uso de array para representar uma fila

- SIM se nós dimensionarmos o array com um tamanho que dê para acomodar o tamanho máximo da fila,
 - e além disso precisamos dos ponteiros **FIM** e **INÍCIO**.

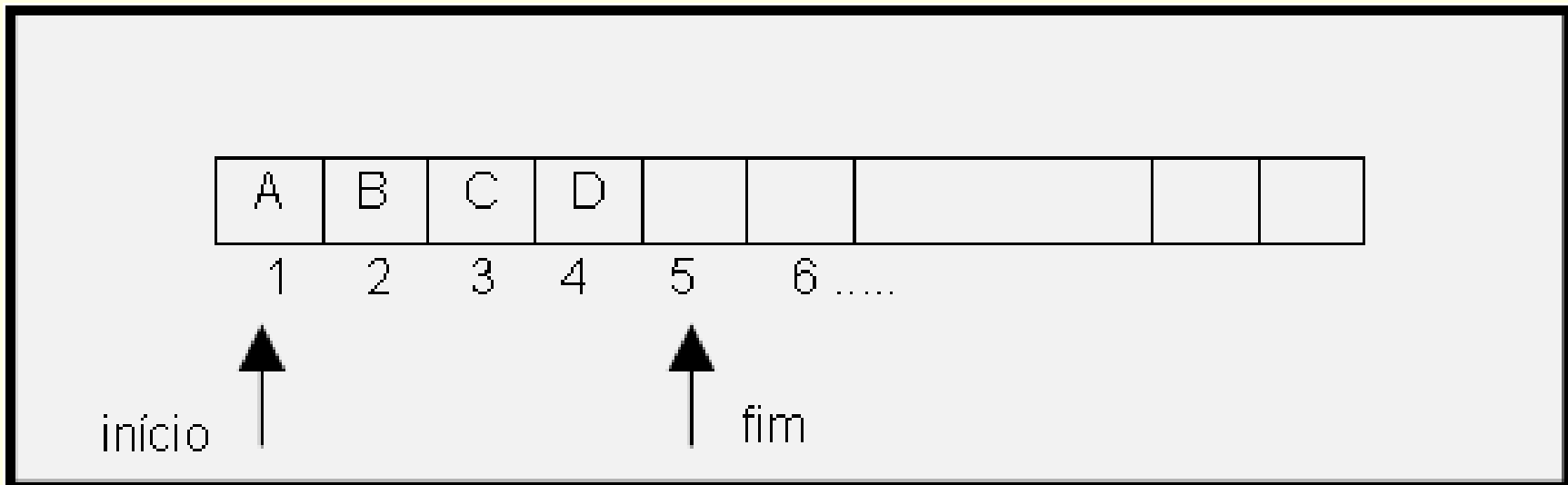
Implementação da fila

- **Início** aponta para/indica o primeiro da fila, ou seja, o primeiro elemento a sair
- **Fim** aponta para/indica o fim da fila, ou seja, onde o próximo elemento entrará



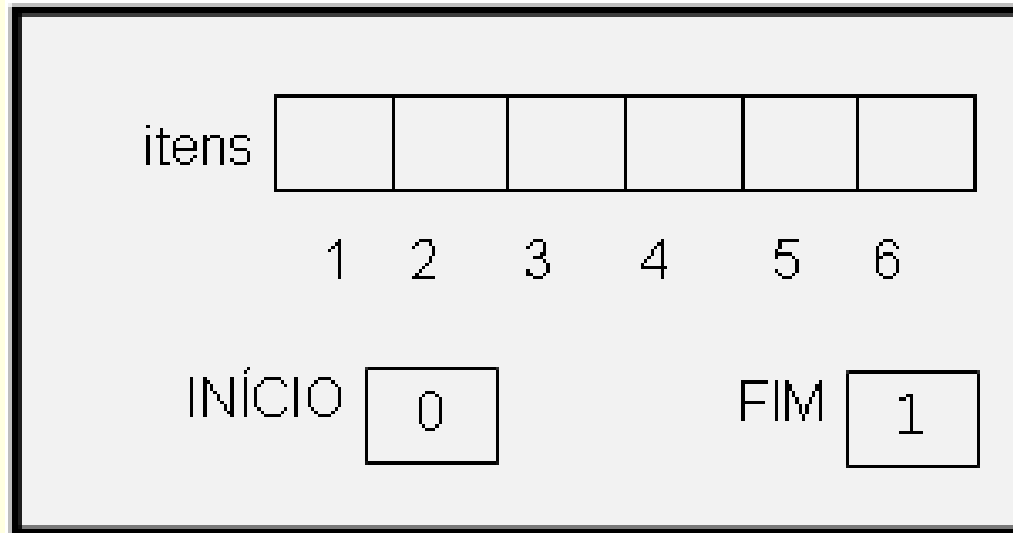
Implementação da fila

- Qual a condição inicial, quando a fila é criada?
 - Início=0, fim=1
- Qual a condição para fila vazia?
 - Início=0, fim=1 ? **Tentem retirar os 4 elementos abaixo**
- Qual a condição para fila cheia?
 - Fim=tamanho da fila +1 ? **Será um mito?**



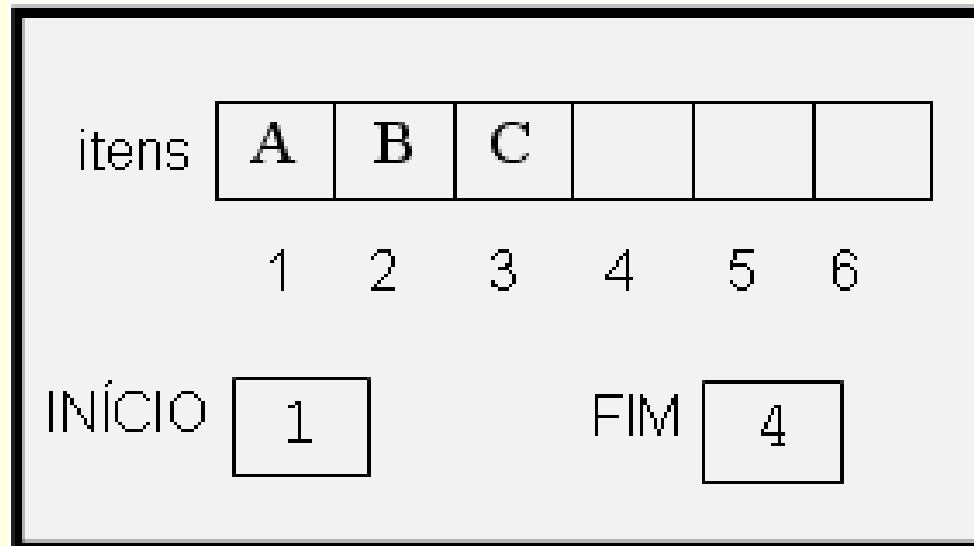
Exemplo de uso da fila

- Criação da fila



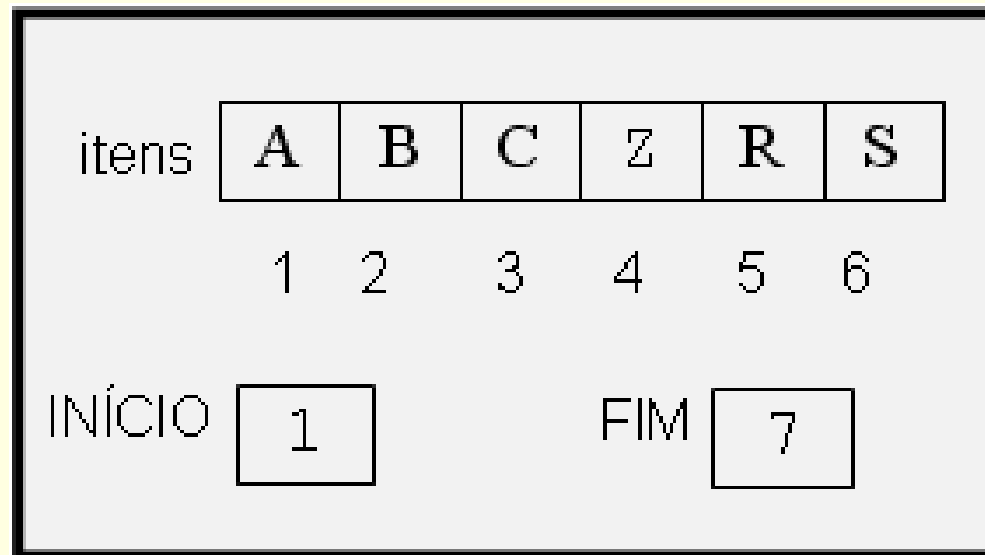
Exemplo de uso da fila

- `entra(F,A)`, `entra(F,B)`, `entra(F,C)`



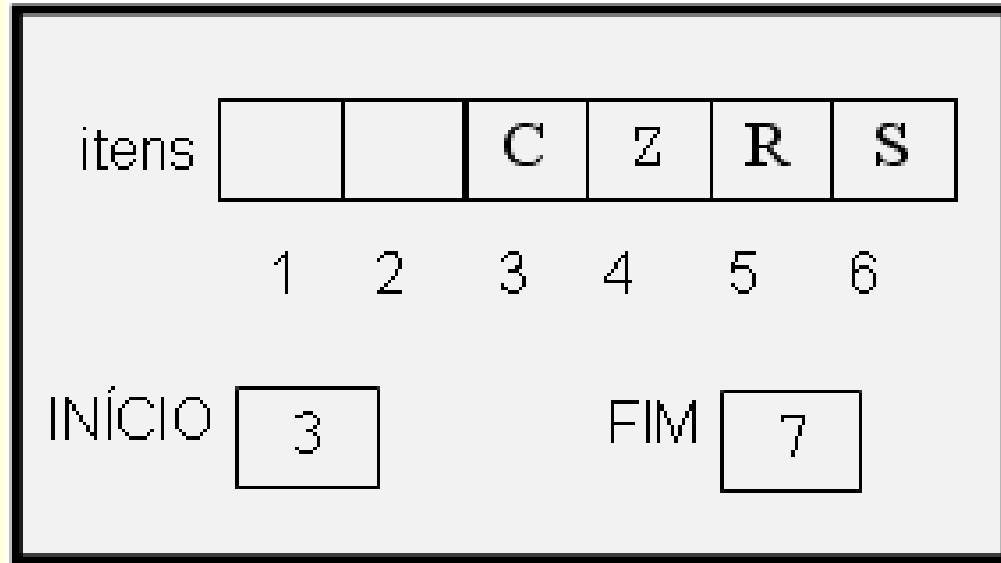
Exemplo de uso da fila

- entra(F,Z), entra(F,R), entra(F,S)
 - Cheia=TRUE



Exemplo de uso da fila

- sai(F,X), sai(F,X)
 - Cheia=FALSE



Como inserir mais elementos?

Qual o problema com esta fila?

Fila

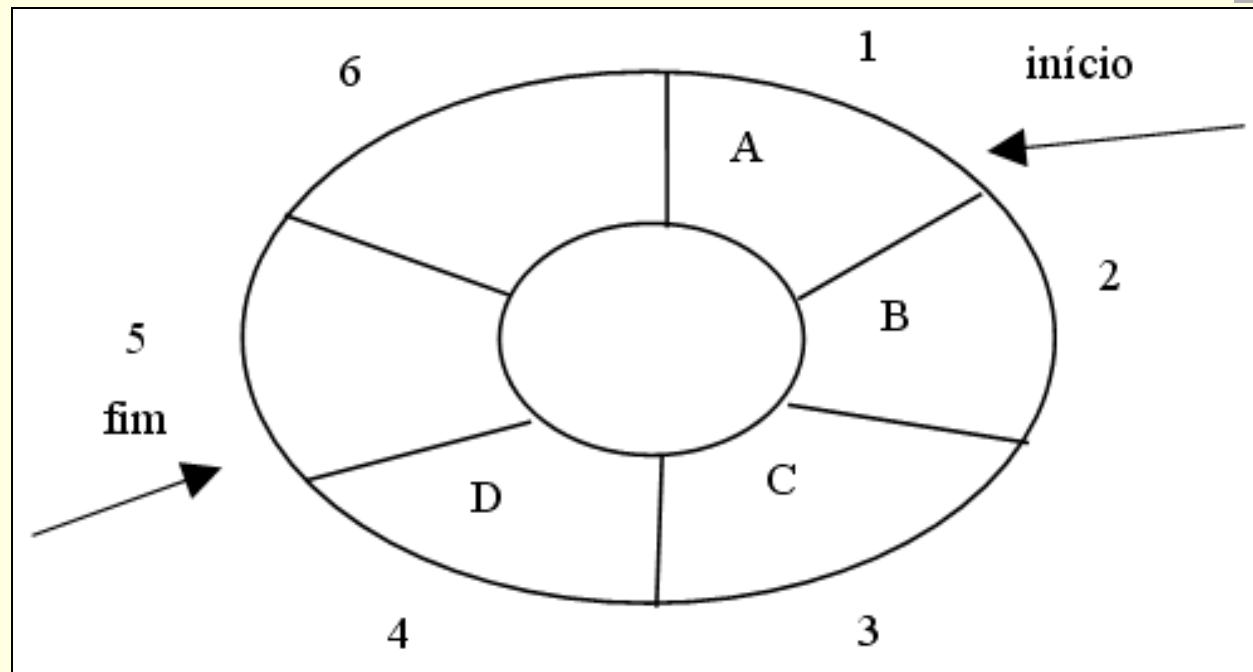
- Como reutilizar os espaços do início da fila?

Fila

- Como reutilizar os espaços do início da fila?
 - Outra forma de implementação
 - Melhor aproveitamento da representação utilizada

Fila em vetor circular!

Fila em vetor circular: se o último estiver ocupado podemos inserir no início do vetor, se esta posição estiver livre



- Qual a condição para fila vazia?
- Qual a condição para fila cheia?
- Qual a condição inicial (quando a fila é criada)?

Uma Solução

- Uso de uma variável para guardar o número de elementos na fila
- Início é ponteiro que indica o elemento inicial (menos quando está vazia) e
- Fim é ponteiro adiantado

Implementação da fila circular estática

```
#define TamFila 100
typedef char elem;
typedef struct fila Fila;
```

Fila.h

```
#include "fila.h"
#include <stdlib.h> /* malloc, free, exit */
#include <stdio.h> /* printf */
```

```
struct fila{
    int inicio, fim, total;
    elem itens[TamFila];
};
```

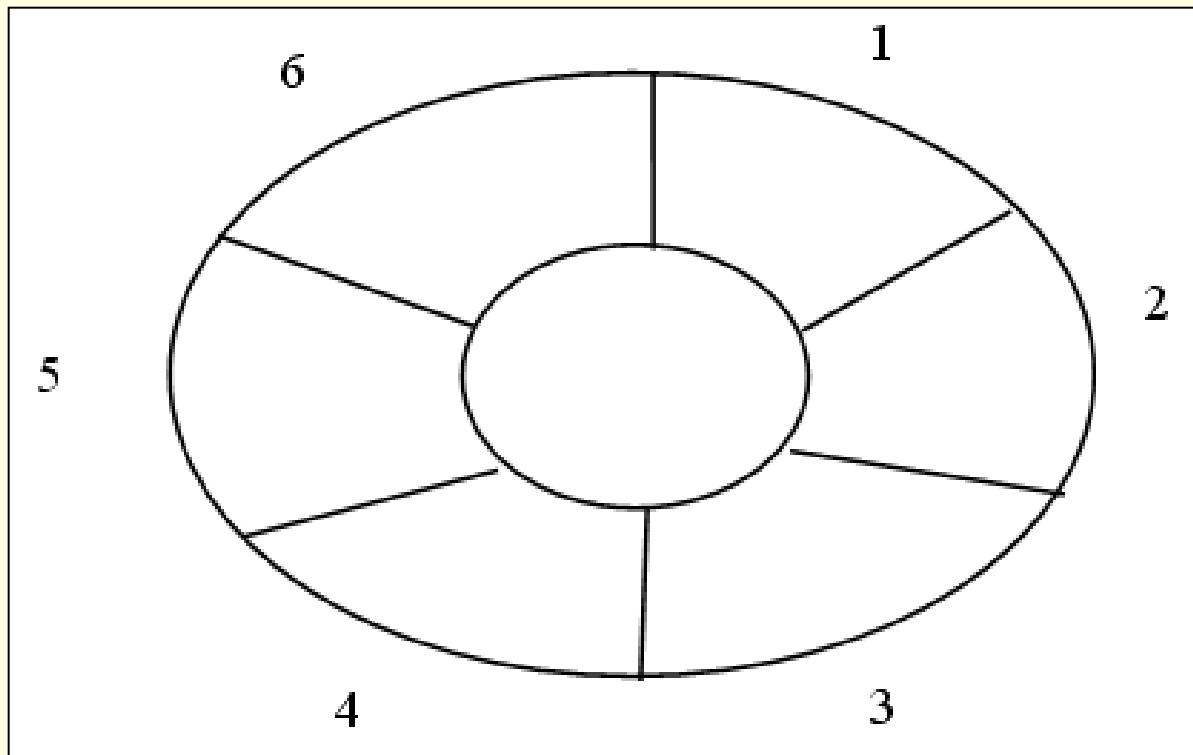
Fila.c

Fila em vetor circular

- Qual a condição para fila vazia?
 - Total=0
- Qual a condição para fila cheia?
 - Total=tamanho da fila
- Qual a condição inicial (quando a fila é criada)?
 - Total=0, início=1, fim=1

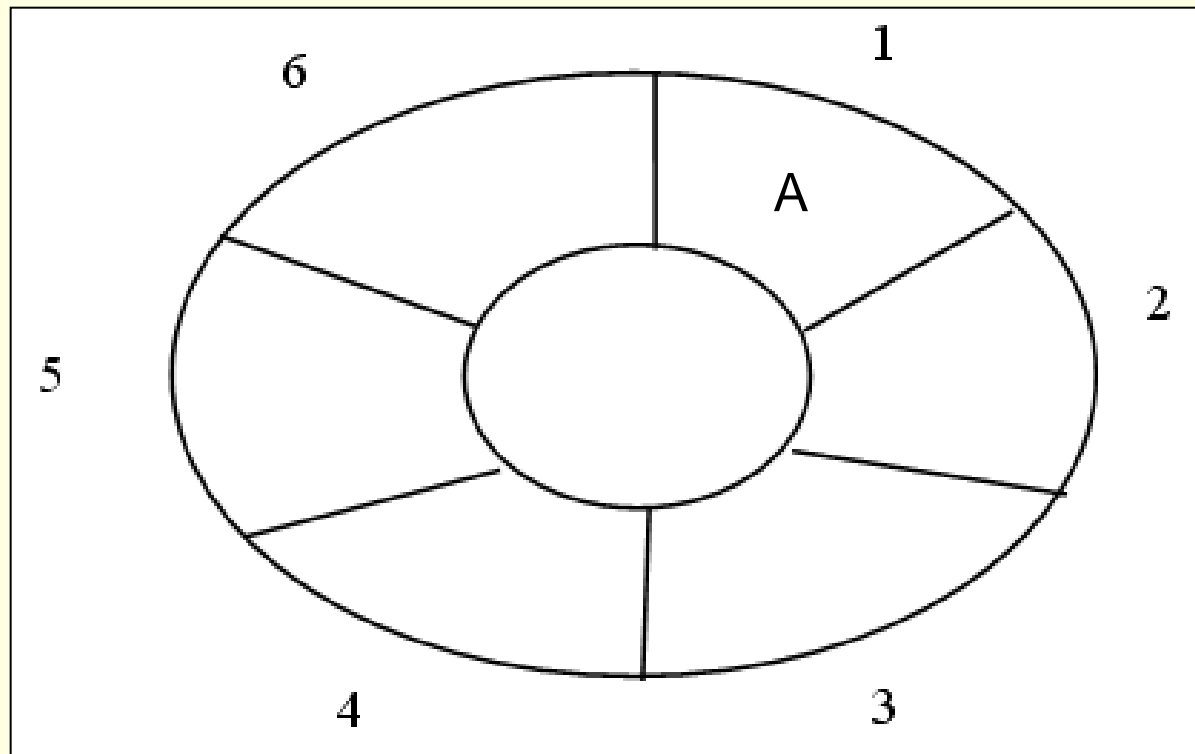
Exemplo

- Fila criada (AJUSTEM PARA C \rightarrow 0)
 - Início=1, fim=1, total=0



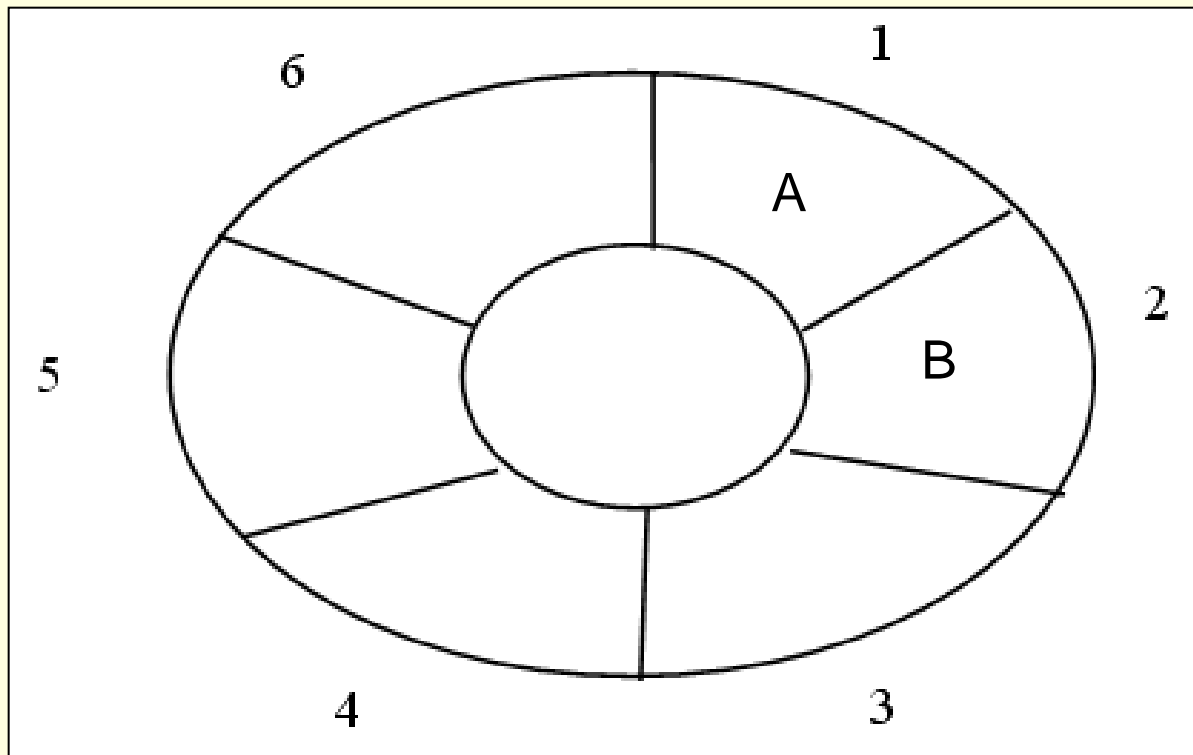
Exemplo

- Entra A
 - Início=1, fim=2, total=1



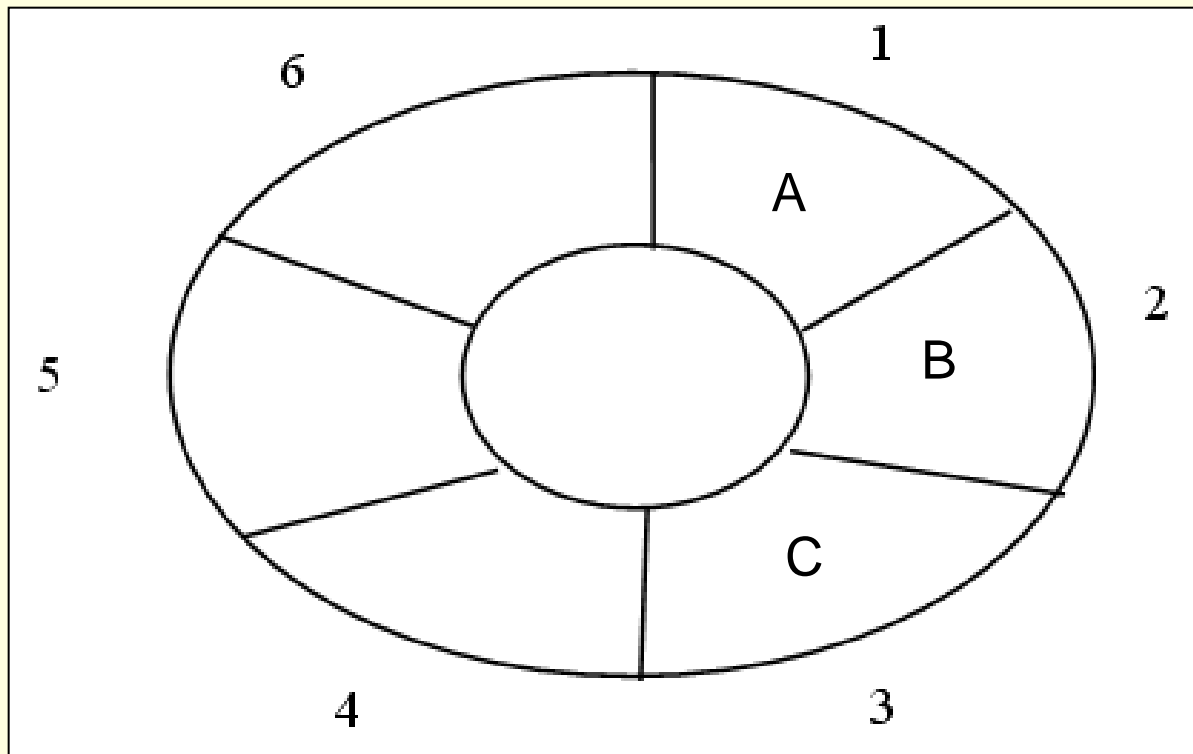
Exemplo

- Entra B
 - Início=1, fim=3, total=2



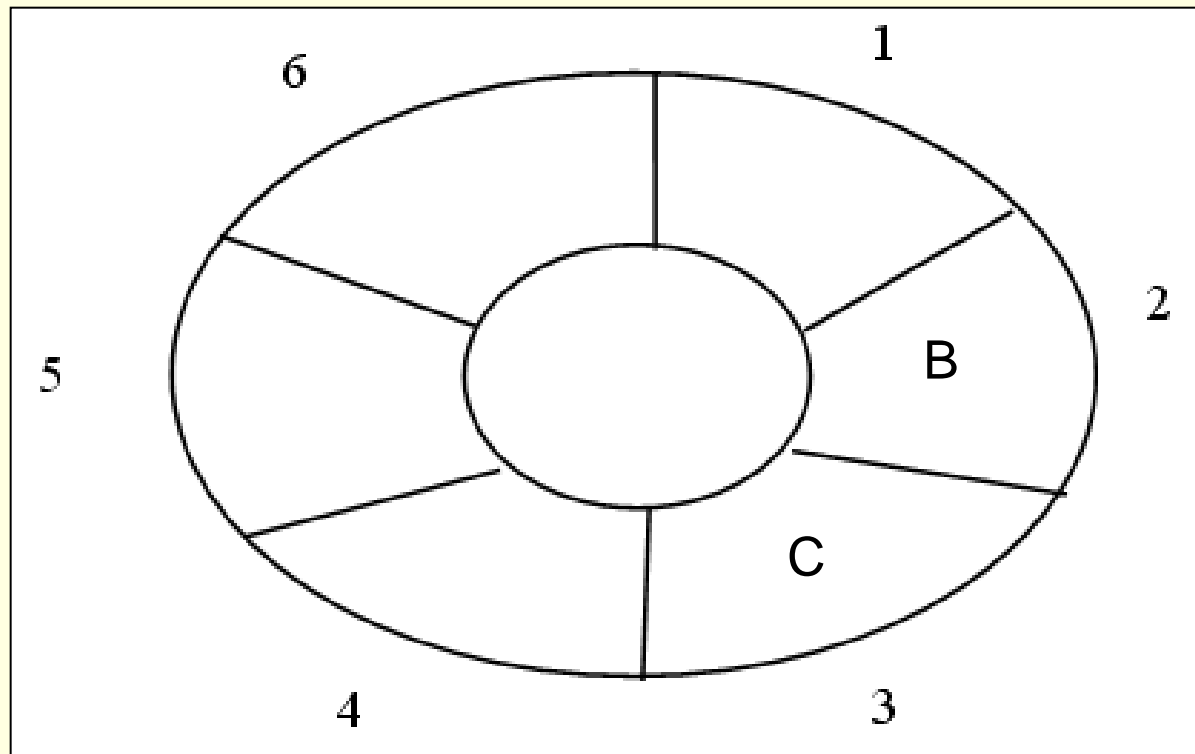
Exemplo

- Entra C
 - Início=1, fim=4, total=3



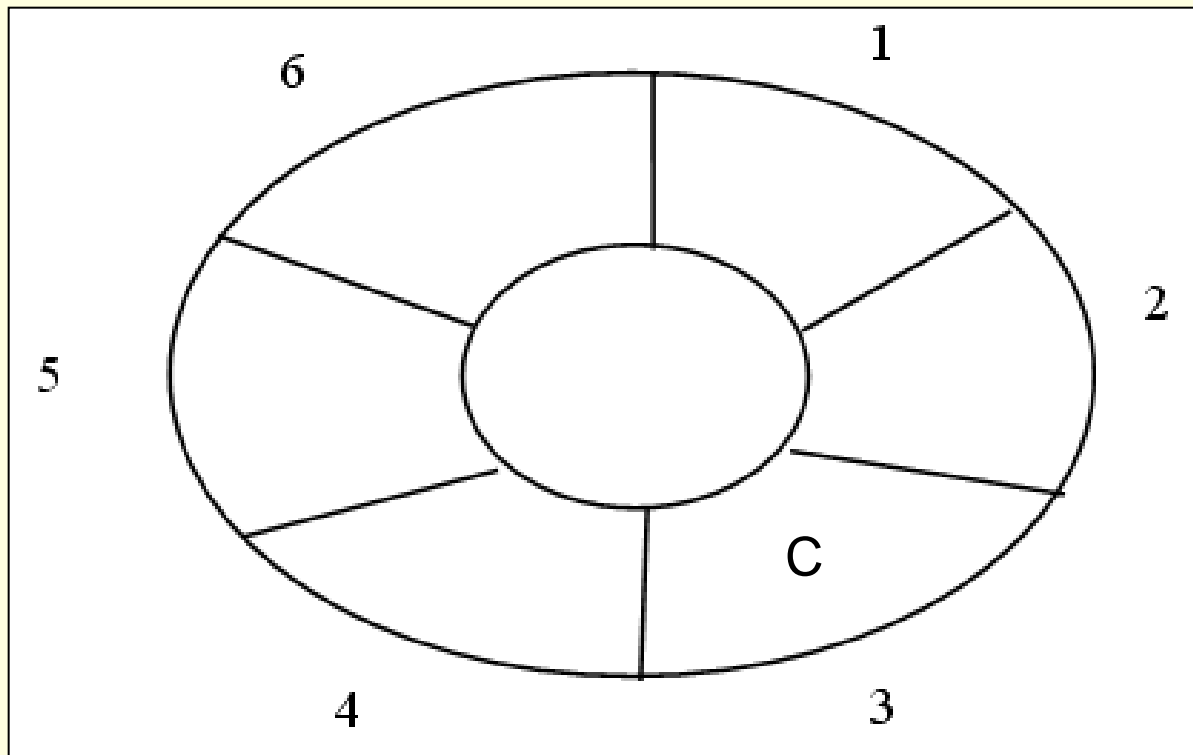
Exemplo

- Sai primeiro
 - Início=2, fim=4, total=2



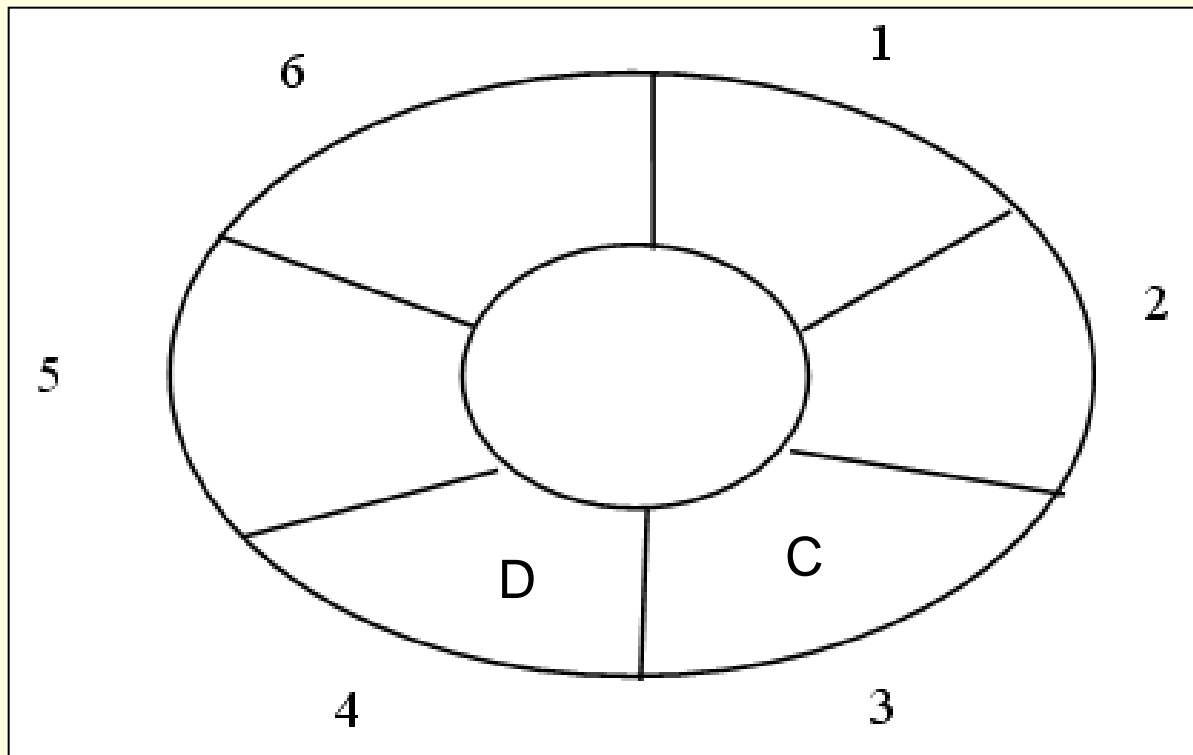
Exemplo

- Sai primeiro
 - Início=3, fim=4, total=1



Exemplo

- Entra D
 - Início=3, fim=5, total=2



Passo a passo para *Entra e Sai*

- Entra elemento no fim da fila
 - vetor[fim]:=elemento
 - avança fim (do TamFila pula para 1, senão incrementa)
 - atualiza total
- Sai primeiro elemento
 - elemento:=vetor[início]
 - avança início (do TamFila pula para 1, senão incrementa)
 - atualiza total
- **AJUSTEM para C**

Operações sobre a fila

■ Implementar as 9 operações do TAD FILA:

Cria(F): cria uma fila F vazia

Tornar_Vazia(F): Reinicializa uma fila existente F como uma fila vazia.

Destroi(F): Remove a fila criada da memória

Entra(F,X): X entra no fim da fila F

Sai(F,X): o primeiro elemento da fila F é retirado da fila e atribuído a X

Começo_Fila(F, X): Mostra o começo da fila F sem remover o item.

Y=Vazia(F): verdade se a fila estiver vazia; caso contrário, falso

Y=Cheia(F): verdade se a fila estiver cheia; caso contrário, falso

Y = Tamanho (F): Retorna o tamanho da fila F

■ Atenção: considerações sobre TAD

- Arquivos .c e .h, parâmetros, mensagens de erro

Exercício

- Faça uma rotina para verificar se os elementos de uma fila estão ordenados de forma crescente

Exercício

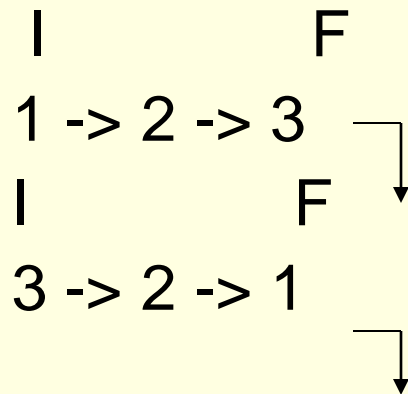
- Faça uma rotina que inverta uma fila F1, criando-se uma nova fila F2

Exercício

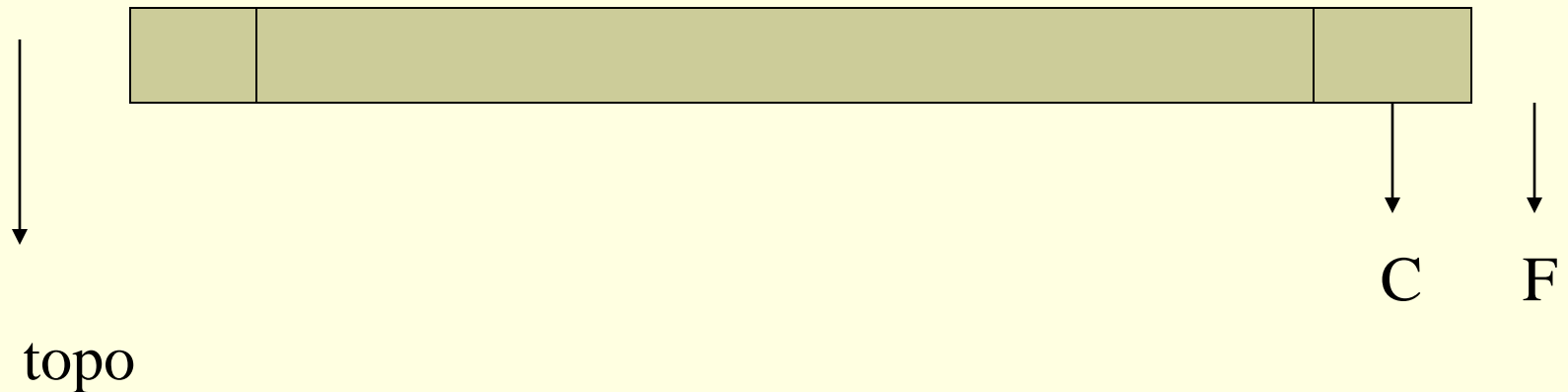
- Desafio: como criar uma fila “mais genérica” que possa guardar tipos diferentes (inteiros e reais, por exemplo)?
 - TAD ainda melhor!

Exercícios:

- 1) Existe algum problema em se trocar os ponteiros da cabeça da fila?
- 2) Implemente um procedimento reverso que reposiciona os elementos na fila de forma que o início se torne fim e vice-versa. Use uma pilha.



3) Obtenha uma representação mapeando uma pilha P e uma fila F em um único array $V[1..n]$. Escreva algoritmos para inserir e eliminar elementos destes 2 objetos de dados. O que você pode dizer sobre a conveniência de sua representação?



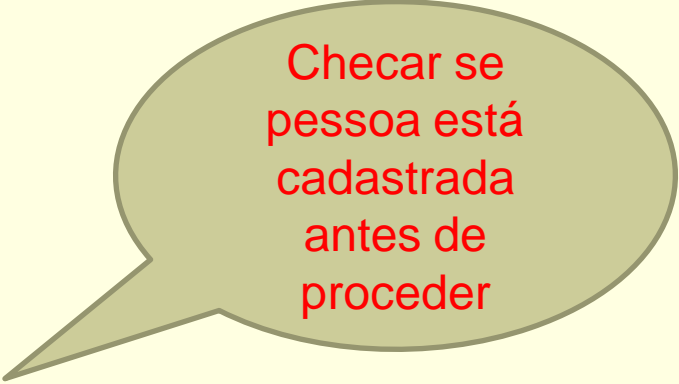
Exercício

- Implemente o sistema para a biblioteca
 - Cada livro deve ser representado por um registro
 - Nome do livro, disponibilidade, fila de espera
 - Ao requisitar um livro, a pessoa entra na fila de espera se o livro não estiver disponível
 - Quando um livro fica disponível, o primeiro da fila de espera do livro deve receber o livro

- Considere prontas as operações sobre a fila

Exercício

```
program biblioteca;
const nroLivros: 1000;
type reg=          record
                   nome: string;
                   disponível: boolean;
                   fila: Fila;
                   end;
var livros: array[1..nroLivros] of reg;
begin
  se livro requisitado
    procure livro no vetor de livros
    se livro disponível
      dar livro para pessoa
      tornar livro indisponível
    senão entra(fila,pessoa)
```



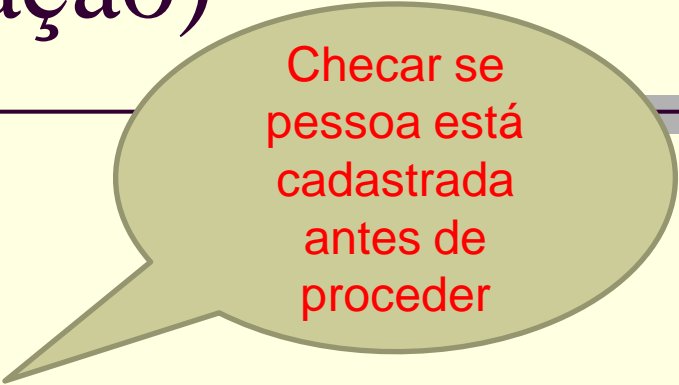
Checar se
pessoa está
cadastrada
antes de
proceder

(continua)

Exercício (continuação)

```
senão se livro devolvido
  procure livro no vetor de livros
  se há lista de espera
    sai(fila,pessoa)
    dar livro para pessoa
  senão tornar livro disponível
```

end.



Checar se
pessoa está
cadastrada
antes de
proceder

Agradecimentos

- Material montado com exemplos do Prof. Thiago A. S. Pardo