

Fila

SCC0202 – Algoritmos e Estruturas de
Dados I

Relembrando

■ Declaração em C

```
#define TamFila 100
```

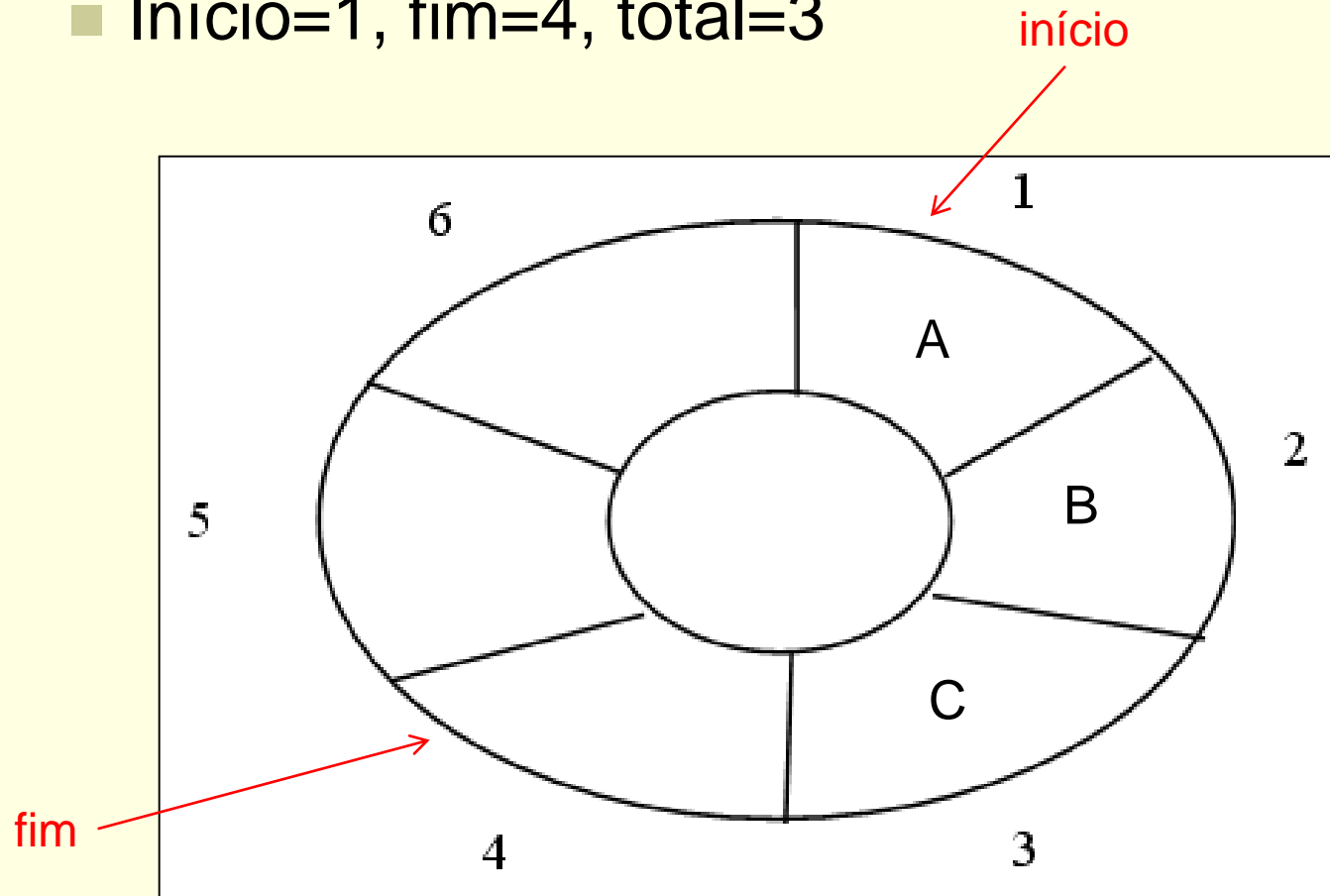
```
typedef int elem;
```

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

```
Fila F;
```

Relembrando

- Fila com vetor circular
 - Início=1, fim=4, total=3



Retomando

- Implementação da função de retirada de um elemento da fila

- Até então:

```
void Cria(Fila *F) {  
    F->inicio=0;  
    F->fim=0;  
    F->total=0;  
    return;  
}
```

```
void Esvazia(Fila *F) {  
    F->inicio=0;  
    F->fim=0;  
    F->total=0;  
    return;  
}
```

```
int EstaVazia(Fila *F) {  
    if (F->total==0)  
        return 1;  
    else return 0;  
}
```

```
int EstaCheia(Fila *F) {  
    if (F->total==TamFila)  
        return 1;  
    else return 0;  
}
```

```
void Entra(Fila *F, elem *X, int *erro) {  
    if (!EstaCheia(F)) {  
        *erro=0;  
        F->total++;  
        F->itens[F->fim]=*X;  
        if (F->fim==TamFila-1)  
            F->fim=0;  
        else F->fim++;  
    }  
    else *erro=1;  
    return;  
}
```

Uma função mais avançada

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

Exercício

- Faça uma rotina que receba duas filas previamente ordenadas e, a partir da união delas, produza uma **terceira fila ordenada**

Exercício

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

Exercício

- Faça uma rotina que **inverte** uma fila F
 - Use uma pilha auxiliar

Exercício... para fazer em casa

- Considere a situação de uma rotina recursiva que contém uma fila de impressão local de documentos. Quando uma nova execução da rotina é acionada, os dados da execução anterior – incluindo a fila – devem ser empilhados na memória, para que sejam retomados posteriormente.
- Declare a **estrutura de dados “pilha de filas”** que seria utilizada em situações como essa

Exercício... para fazer em casa

- Considere a situação de uma rotina recursiva que contém uma fila de impressão local de documentos. Quando uma nova execução da rotina é acionada, os dados da execução anterior – incluindo a fila – devem ser empilhados na memória, para que sejam retomados posteriormente.
- Declare a estrutura de dados “pilha de filas” que seria utilizada em situações como essa
- Implemente as funcionalidades básicas de **empilhar uma fila** e **desempilhar uma fila**

Exercício... para pensar

- E como seria uma “**fila de pilhas**”?

Exercício... para pensar

- E como seria uma “fila de pilhas”?
- E uma “**fila de filas**”?

Exercício

- Implemente o sistema para a biblioteca usando o TAD fila
 - 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
 - Implemente as demais funcionalidades (cadastra livro, retira livro, etc.) que julgar necessárias

Um começo...

- Implemente o sistema para a biblioteca usando o TAD fila
 - Cada livro deve ser representado por um registro
 - Nome do livro, disponibilidade, fila de espera
 - Como é a declaração dessa estrutura de dados?

Um começo...

■ Possível declaração

```
#define NroLivros 1000
#define TamFila 100

typedef struct {
    char nome_pessoa[50];
    int telefone[20];
} elem;

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

```
typedef struct {
    char nome_livro[100];
    int disponivel;
    Fila fila_espera;
} Livro

Livro Biblioteca[NroLivros];
```