

# Estruturas de Dados Clássicas – Filas – Parte 1

**Prof. Bárbara Quintela**

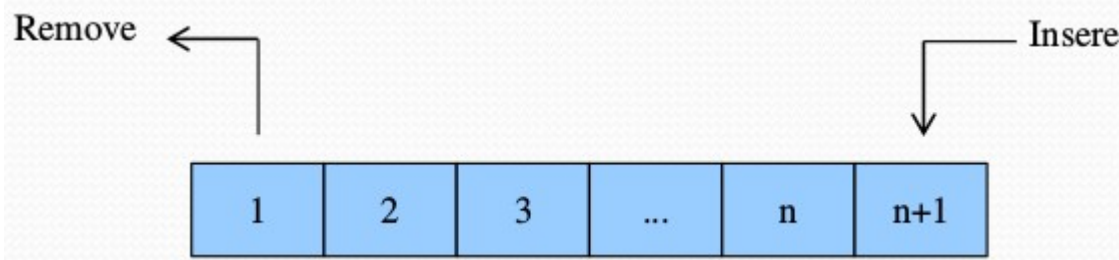
[barbaraquintela@pucminas.cesjf.br](mailto:barbaraquintela@pucminas.cesjf.br)





# Definição de Fila - Queue

- São estruturas de dados do tipo FIFO (first-in first-out) - o primeiro elemento a ser inserido, será o primeiro a ser retirado.
- A inserção dos elementos é dada por uma das extremidades da lista e a remoção por outra.
- Exemplos de filas são:
  - Fila de caixa de banco,
  - Fila de vagões de trem,
  - etc.





# Aplicações de Fila

- Fila de arquivos para impressão;
- Atendimento de processos requisitados a um sistema operacional;
- Buffer para gravação de dados em mídia;
- Processos de comunicação em redes de computadores.



# Operações com Fila

- **Criação da fila** - informar a capacidade no caso de fila por contiguidade (usando vetor);
- **Inicialização da fila** - determina o status inicial da fila, a fim de prepará-la para a inserção de dados.
- **Verificar se a fila está cheia** (no caso de fila por contiguidade - vetor)
- **Enfileiramento** (enqueue) - consiste em inserir um valor no fim da fila. É preciso verificar previamente se a fila está cheia.
- **Verificar se a fila está vazia**
- **Desenfileiramento** (dequeue) - consiste em retirar um valor do início da fila. É preciso verificar previamente se a fila está vazia.
- **Mostrar a fila**



# Criar a Fila

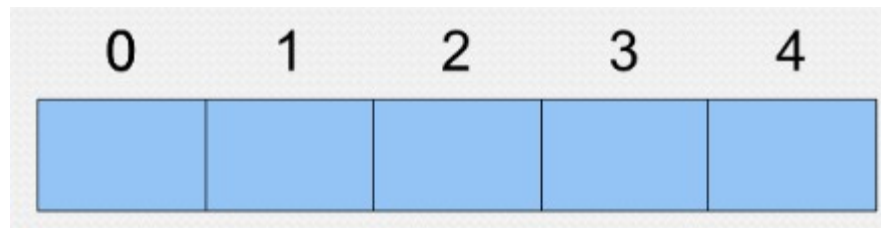
- Supondo fila com capacidade para 5 valores reais (5 nós)

**# define TAM 5** ou **int const tam = 5;**

**float** fila[TAM]; //fila propriamente dita

**int** inicio, //armazena indice do inicio da fila

fim; //armazena indice do elemento no fim da fila





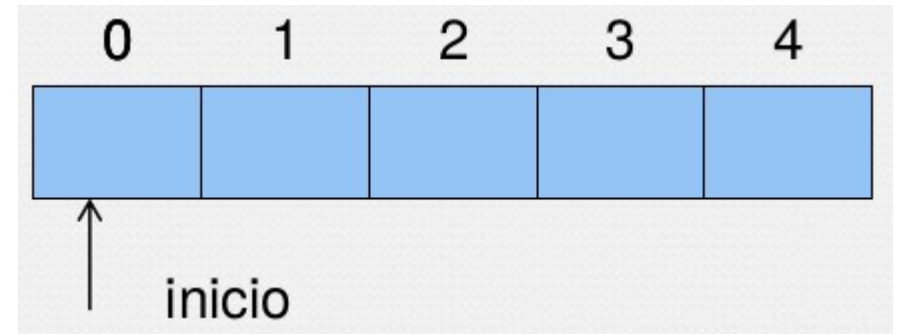
# Inicializar a Fila

- Determina status inicial da fila, para preparar para inserção de dados

Na função principal (main):

`inicio = 0;`

`fim = -1;`

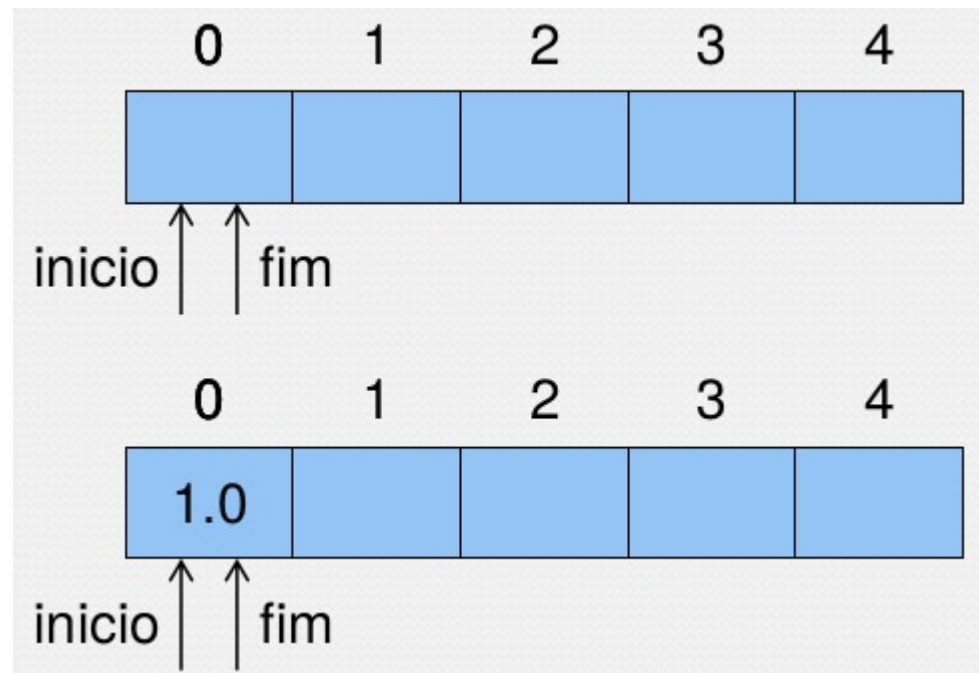


- Por que -1?
  - Quando primeiro valor for enfileirado, início e fim serão iguais a zero (índice)
  - -1 no fim indica fila sem dados



# Enfileirar (enqueue)

- Consiste em inserir valor no fim da fila
    - Desde que não esteja cheia
  - É preciso ajustar o fim e então armazenar o dado no vetor
  - Parâmetros:
    - Vetor de elementos
    - Valor a ser inserido
    - Fim da fila
  - Retorno: não há
- Ex: enfileirar 1.0

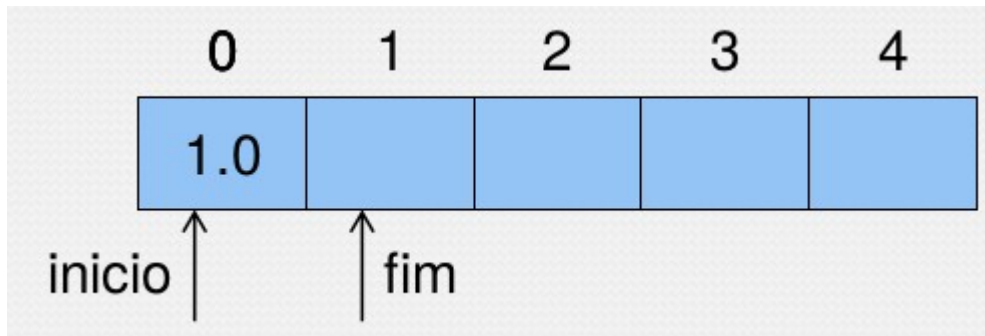




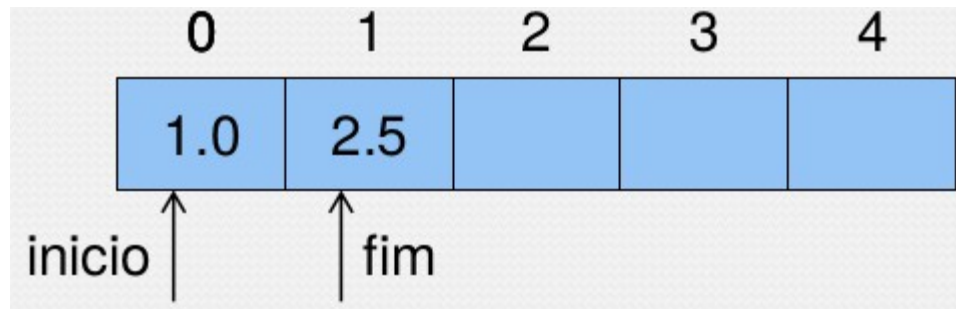
# Enfileirar (enqueue)

- Ex: enfileirar 2.5

Atualiza o fim



Insere o 2.5







# Enfileirar (enqueue)

- Ex: enfileirar 3.2  
enfileirar 3.0





# Código da Função Enfileirar

```
void enqueue (float f[], float valor, int *fim) {  
    if ((*fim) == TAM - 1) { // testa fila cheia – é uma operação  
        printf("ERRO: Fila cheia.\n");  
        return; // sai da função  
    }  
    (*fim)++; // ajusta fim  
    f[(*fim)] = valor; // armazena o valor fim da fila  
}
```

- **Chamada** : enqueue (fila, valor, &fim);



# Desenfileirar (dequeue)

- Consiste em retirar um valor do início da fila e ajustar o início.
- Só é possível se a fila **não** estiver vazia.
- Parâmetros :
  - vetor de elementos;
  - inicio da fila;
  - fim da fila;
  - valor para armazenar o dado desenfileirado.
- Retorno : nada



- [illegible]



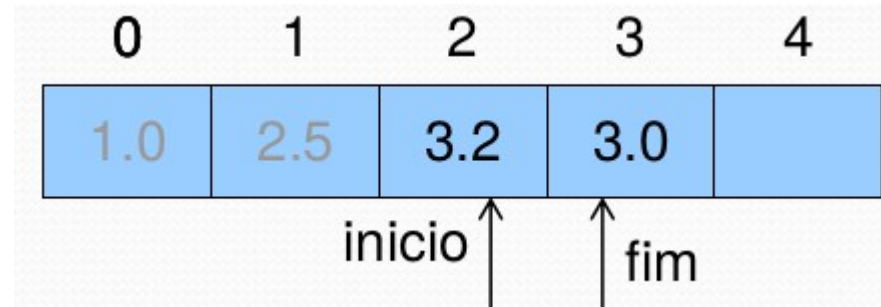
# Código da Função Desenfileirar

```
int dequeue (float f[], int *inicio, int *fim, float *valor) {  
    if ((*inicio) > (*fim)) { // testa fila vazia - operação  
        printf("ERRO: Fila fazia.\n");  
        return 0; // sinaliza fracasso  
    }  
    (*valor) = f[(*inicio)]; // retira do início  
    (*inicio)++; //ajusta o início  
    return 1; // sinaliza sucesso  
}
```

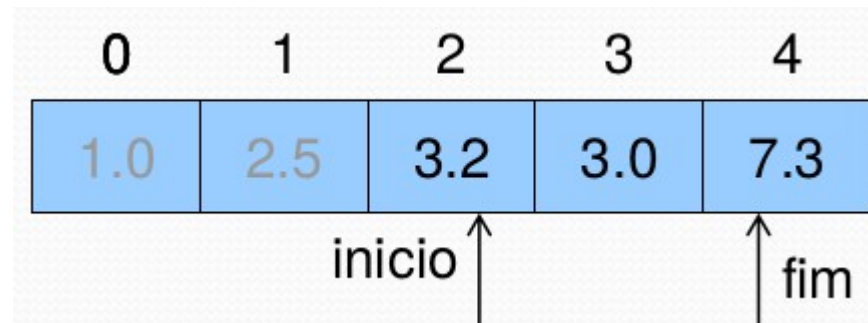
- **Chamada (trecho):** dequeue (fila,&inicio,&fim,&valor);



# Desenfileirar/Enfileirar



Enfileirar 7.3



Enfileirar 5.4

O vetor acabou mas ha espaço no início da fila!!!

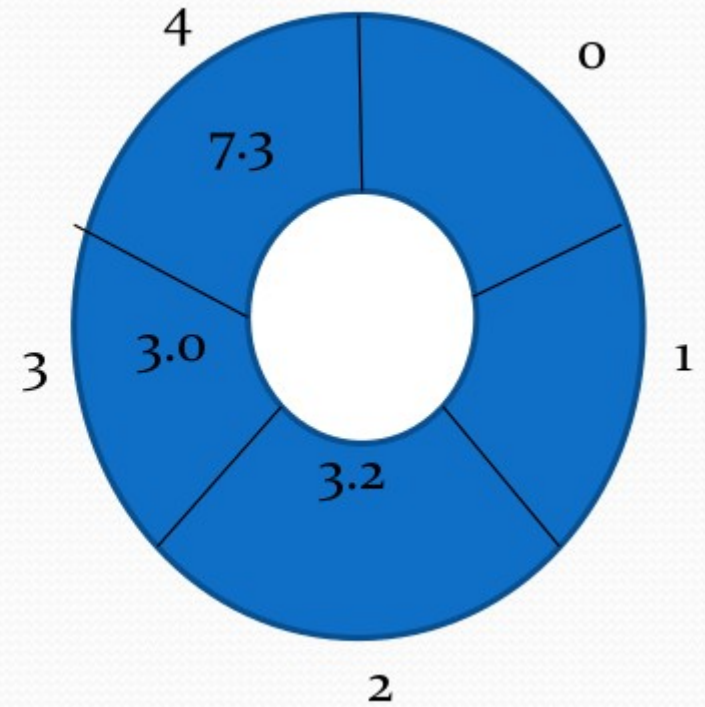
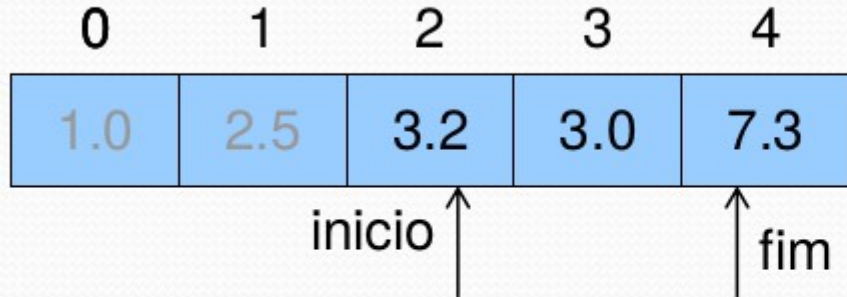


# Solução

- Deslocar elementos para início da fila
- Uso de uma FILA CIRCULAR
  - Solução mais empregada
  - Menor custo computacional
  - Mais fácil implementação



# Fila circular

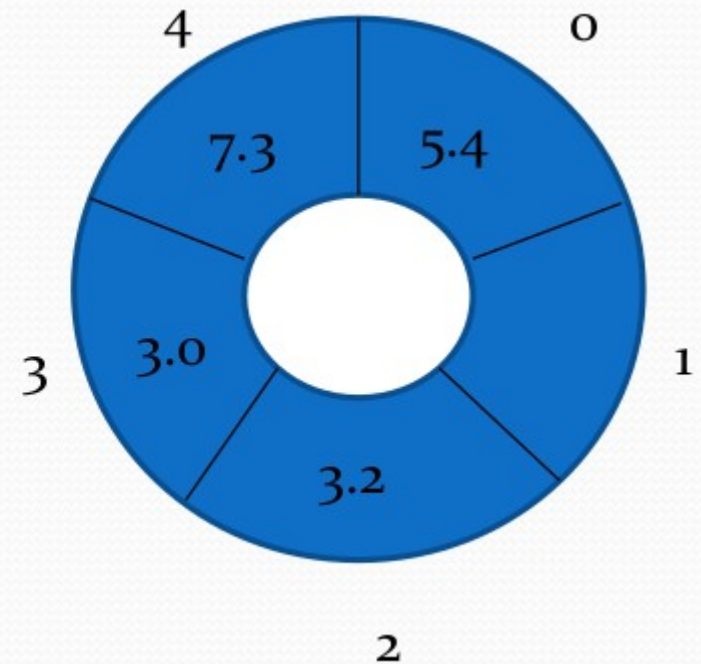
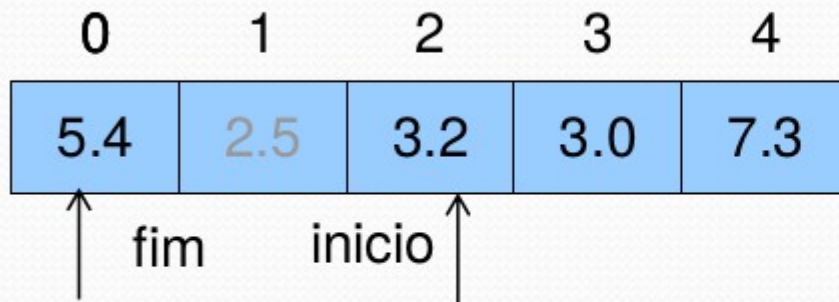






# Fila circular

- Enfileirar o 5.4





# Criar Fila circular

- Para facilitar incluímos uma informação – quantidade de elementos existente na fila circular (nltens)
- Supondo uma fila com capacidade para 5 valores reais (5 nós).

**#define TAM 5** ou **int const tam = 5;**

**float** fila[TAM]; // fila propriamente dita - armazena os dados

**int** inicio, // armazena o índice do dado no início da fila

fim, // armazena o índice do elemento no fim da fila

nltens; // armazena a quantidade de elementos existente na fila



# Inicializar Fila circular

- Determina o status inicial da fila, a fim de prepará-la para a inserção de dados.
- Na função principal (main):

inicio = 0;

fim = -1;

nlens = 0;



# Enfileirar Elemento na Fila circular

- Consiste em **inserir** um valor no fim da fila, desde que a fila não esteja cheia.
- É preciso **ajustar o fim** (controlando para o vetor não "estourar") e então, armazenar o dado no vetor.
- Caso haja **espaço** e o **último elemento** se encontra na **última posição** do vetor, o elemento será inserido na **primeira posição** do vetor.
- **Parâmetros :**
  - vetor de elementos;
  - valor a ser inserido;
  - fim da fila;
  - número de elementos existente na fila.
- **Retorno :** Não há



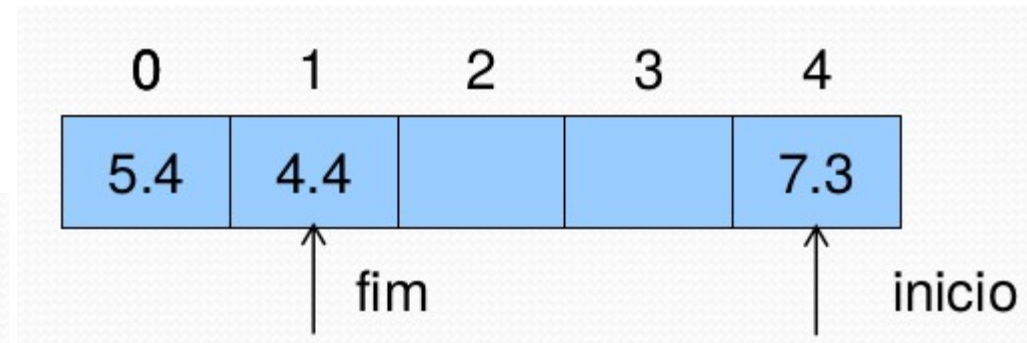
# Código Função Enfileirar Fila circular

```
void enqueue (float f[], float valor, int *fim, int *nltens) {  
    if (nltens == TAM) { // testa fila cheia  
        printf("ERRO : Fila cheia! \n");  
        return; // sai da função  
    }  
    (*fim)++; // ajusta fim  
    if ((*fim) == TAM) { // estourou o vetor  
        (*fim) = 0;  
    }  
    f[(*fim)] = valor; // armazena o valor fim da fila  
    (*nltens)++;  
}
```

**Chamada** : enqueue (fila,valor,&fim, &nltens);



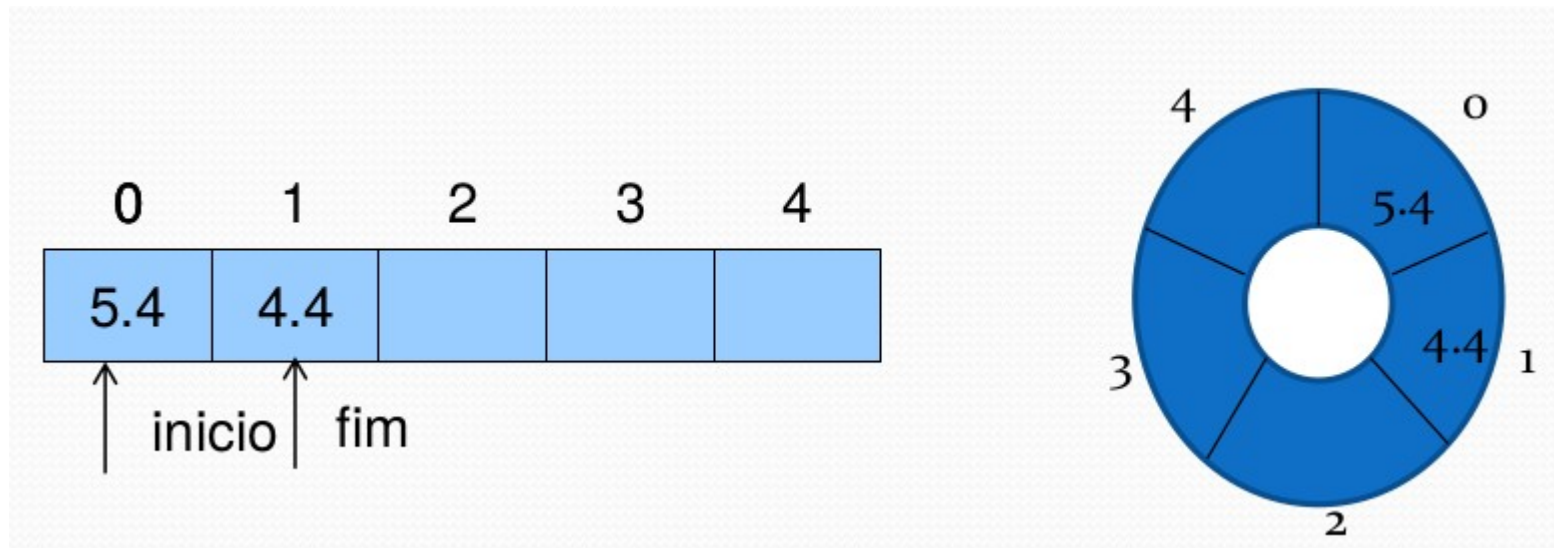
# Desenfileira





# Desenfileirar Fila circular

## Desenfileira





# Desenfileirar Fila circular

- Consiste em **remover** um valor do **início** da fila, desde que a fila não esteja **vazia**.
- Para desenfileirar, é preciso **ajustar o início** (controlando quando o elemento da última posição do vetor for removido , neste caso, o fim passa a ser a primeira posição do vetor).
- **Parâmetros** :
  - vetor de elementos;
  - início fila;
  - número de elementos existente na fila.
- **Retorno** : Não há





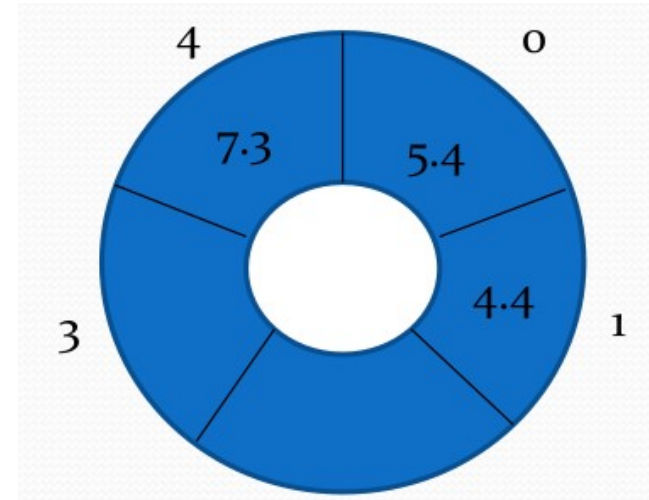
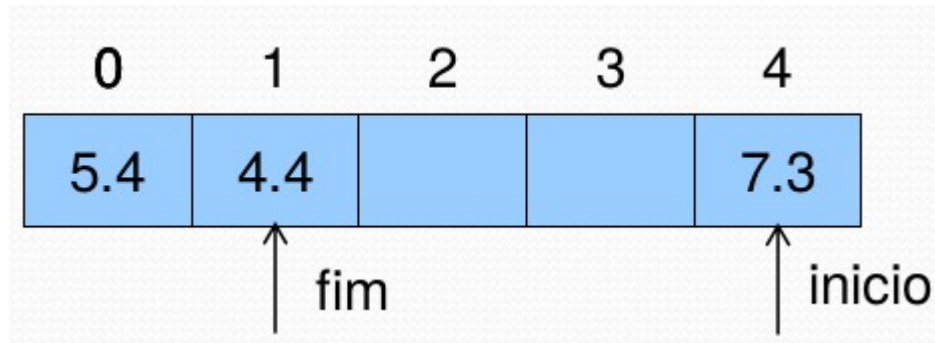
# Código Desenfileirar Fila circular

```
void dequeue (float f[], int *inicio, int *nItens) {  
    if ((*nItens) == 0) { // testa se a fila está vazia  
        printf("ERRO : Fila vazia! \n");  
        return; // sai da função  
    }  
    (*inicio)++; // ajusta o inicio  
    if ((*inicio) == TAM) { // se estourou o vetor ...  
        (*inicio) = 0; // vai para primeira posição do vetor  
    }  
    (*nItens)--; // um item retirado  
}
```



# Mostrar elementos

- Supondo a seguinte situação



- Mostrar os elementos não é simplesmente percorrer o vetor da primeira posição (posição 0) até a última posição (no exemplo posição 4)
- O que deseja-se é percorrer o vetor do início da fila (no exemplo, posição 4) até o fim da fila (no exemplo posição 1)
- Lembrar que só se pode mostrar uma fila que NÃO esteja vazia



# Mostrar elementos

```
void displayQueue(float f[], int inicio, int fim, int nltens) {  
    if (nltens == 0) { // testa se a fila esta vazia  
        printf("ERRO: fila vazia! \n");  
        return; // sai da função  
    }  
    for (int cont=0, i= inicio; cont < nltens; cont++){  
        printf("%f ", f[i++]);  
        if (i == TAM)  
            i=0;  
    }  
    printf("\n\n");  
}
```

**Chamada:** displayQueue (fila, inicio, fim, nltens);



# Outras Funções

- Verificar se a fila está vazia

// retorna verdadeiro se a fila está vazia

```
bool isEmpty(float f[], int nltens) {  
    return (nltens==0);  
}
```

- Verificar se a fila está cheia

// retorna verdadeiro se a fila está cheia

```
bool isFull(float f[], int nltens) {  
    return (nltens == TAM);  
}
```

# Exercícios - Questões de concursos (pilha e fila)



- Empresa de Desenvolvimento Urbano - EMDUR – 2007  
Uma das estruturas de dados utilizadas na programação de computadores funciona conforme o princípio conhecido como FIFO – “First In First Out” e uma como LIFO – “Last In First Out”. Essas estruturas são denominadas, respectivamente:
  - (A) Lista Circular e Árvore
  - (B) Árvore e Lista Linear
  - (C) Pilha e Lista Circular
  - (D) Lista Linear e Fila
  - (E) Fila e Pilha



# Exercícios

- Concurso Público - 2006 Prefeitura de Várzea Paulista

A representação a seguir refere-se a um conjunto de elementos armazenados em um array. A remoção de um elemento desse conjunto segue a regra “o primeiro elemento que entra é o primeiro elemento que sai (FIFO)”.

A representação acima refere-se a uma:

- (A) pilha;
- (B) fila;
- (C) lista encadeada;
- (D) árvore;
- (E) lista binária;



# Exercícios

- FCC - 2010 - MPE-RN - Analista de Tecnologia da Informação - Engenharia de Software
- Último dado armazenado é o primeiro a ser recuperado caracteriza a estrutura de dados do tipo:
- a) árvore.
  - b) pilha.
  - c) string.
  - d) fila.
  - e) boolean.



# Exercícios

- FCC - 2008 - MPE-RS - Técnico em Informática  
- Área Sistemas

Respeitando as ordens de inserção e de retirada dos dados, uma estrutura de:

- a) fila é também denominada LIFO ou LILO.
- b) fila é também denominada FIFO ou FILO.
- c) fila é também denominada FIFO ou LIFO.
- d) pilha é também denominada FIFO ou FILO
- e) pilha é também denominada LIFO ou FILO.





# Exercícios

- As estruturas do tipo LIFO (Last-In-First-Out) e FIFO (First-In-First-Out) são classificadas, respectivamente, como:
  - A) pilha e fila;
  - B) e lista;
  - C) lista e pilha;
  - D) grafo e ;
  - E) pilha e grafo.



# Exercícios

- BNDS 2002 – Analista de Sistemas (Desenvolvimento)

Considere os processos de inserção e remoção de elementos de uma determinada estrutura de dados. Dois tipos especiais de estruturas de dados que, sob o ponto de vista da inserção e remoção de elementos, são classificadas como estruturas do tipo LIFO – Last-In-First-Out e FIFO – First-In-First-Out são, respectivamente,

- (A) B-Tree e lista.
- (B) pilha e fila.
- (C) lista e pilha.
- (D) grafo e B-Tree.
- (E) fila e grafo.



# Exercícios

Qual das afirmações abaixo melhor se aplica a estruturas de dados lineares?

- A) Pilhas e filas são estruturas do tipo FIFO e FILO, respectivamente;
- B) Uma fila pode ser entendida como uma lista duplamente ligada;
- C) Uma pilha pode ser vista como um caso especial de uma fila;
- D) Nas listas ligadas circulares, o último elemento deve ter uma referência para acesso ao primeiro elemento da lista;
- E) Nenhuma das anteriores.



# Exercícios

TRE/MG – Analista de Sistemas – 2005

“ É uma lista linear em que todas as inserções de novos elementos são realizadas numa extremidade da lista e todas as remoções de elementos são feitas na outra extremidade da lista”.

Esta é uma definição de:

- a) fila que é uma estrutura de dados do tipo FIFO (First In First Out).
- b) pilha que é uma estrutura de dados do tipo FILO (First In Last Out).
- c) fila circular que é uma estrutura de dados do tipo FILO (First In Last Out).
- d) pilha ordenada que é uma estrutura de dados do tipo FIGO (First In Goback Out).
- e) fila que é uma estrutura de dados do tipo LIFO (Last In First Out).



# Exercícios

No conjunto de operações de uma fila circular estão disponíveis as funções:

- Enfileirar (enqueue) – insere um elemento na fila (após o último);
- Desenfileirar (dequeue) – remove o primeiro elemento da fila;
- Mostrar a fila (showQueue) – mostra os valores de cada um dos elementos da fila;
- Verificar se a fila está vazia (isEmpty) – opcional – retorna true, se não existirem elementos na fila;
- Verificar se a fila está cheia (isFull) ) – opcional – retorna true, se não houver espaço para inserir um elemento na fila;

Necessita-se de uma operação para inserir um elemento antes do primeiro.

- Implemente uma função chamada furaFila, que recebe como parametro o vetor, que representa a fila propriamente dita; o valor a ser inserido; a posição do primeiro elemento dentro do vetor e a quantidade de elementos na fila (nItens) e realiza a operação desejada.
- A função de furar a fila só poderá ser realizada se a fila não estiver cheia.