

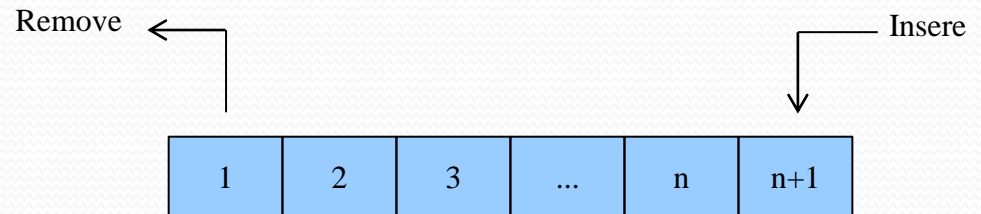
# Estrutura de Dados

Aula 7 e 8  
Filas e suas Aplicações

Prof. Leticia Winkler

# 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 ao 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

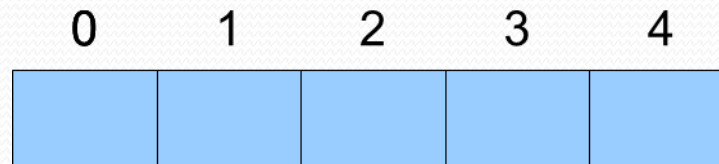
- Supondo uma fila com capacidade para 5 valores reais (5 nós).

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

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

```
int inicio, // armazena o índice do dado que está no início da fila
```

```
    fim; // armazena o índice do elemento que está no fim da fila
```



# Inicializar a Fila

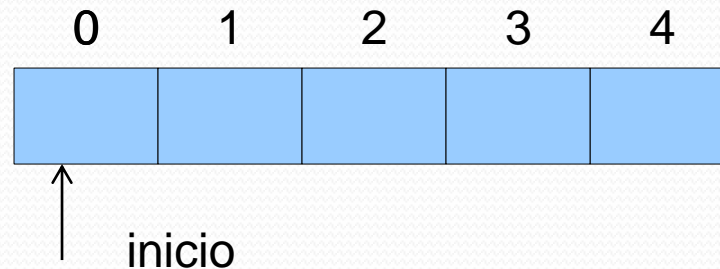
- Determina o status inicial da fila, a fim de prepará-la para a inserção de dados.

- Na main:

**início = 0;**

**fim = -1;**

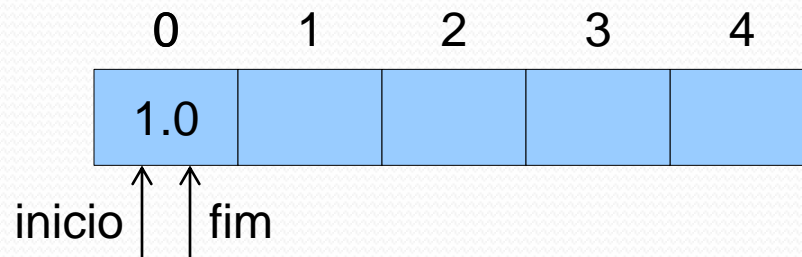
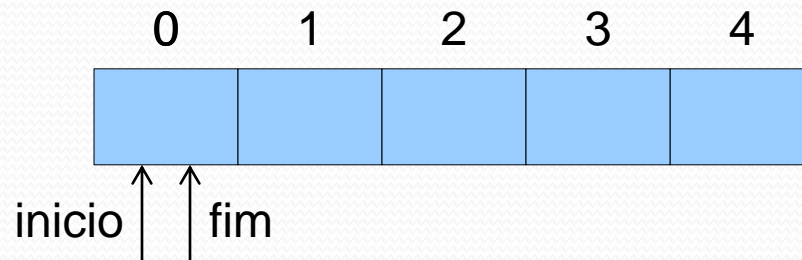
- Por que 0 e -1 ?



- Quando o primeiro valor for enfileirado, o início e o fim serão iguais a zero, pois ele (o valor) é o primeiro e último da fila.
  - O fim deverá ser -1 (índice impossível) para caracterizar fila sem dados.
  - O fim igual a -1 garantirá o status de fila sem dados. Assim, quando ocorrer a 1ª. inserção, o valor inserido já estará adequadamente no início da fila - área de índice zero.

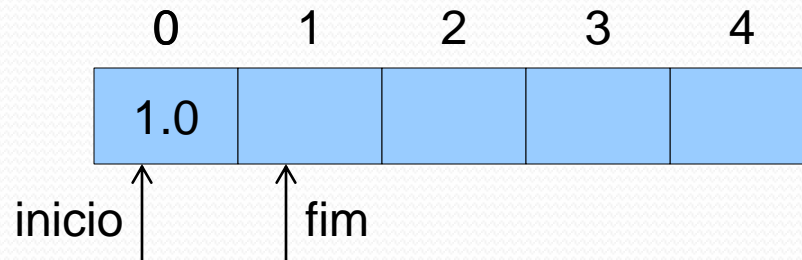
# Enfileirar (enqueue)

- Consiste em inserir um valor no fim da fila, desde que a fila não esteja cheia.
- Para enfileirar, é preciso ajustar o fim e então, armazenar o dado no vetor.
- Parâmetros : vetor de elementos, valor a ser inserido e o fim da fila
- Retorno : Não há
- Enfileirar o 1.0

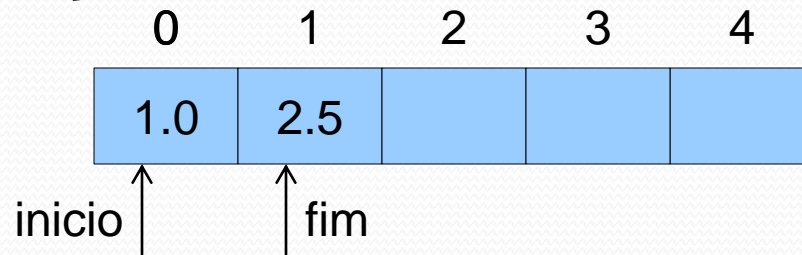


# Enfileirar

- Enfileirar 2.5
  - Atualiza o fim



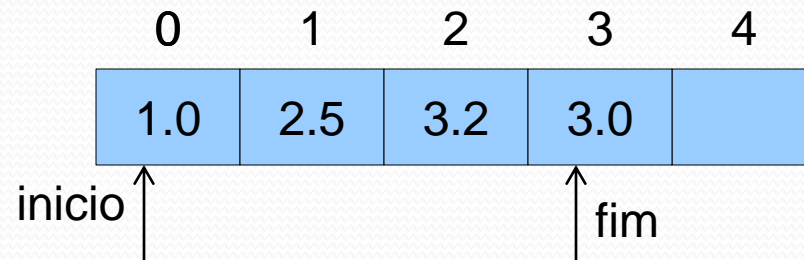
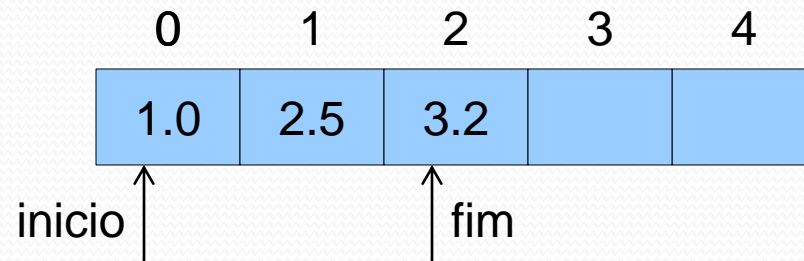
- Inserir o 2.5





# Enfileirar

- Enfileirar 3.2
- Enfileirar 3.0



# Código da Função Enfileirar

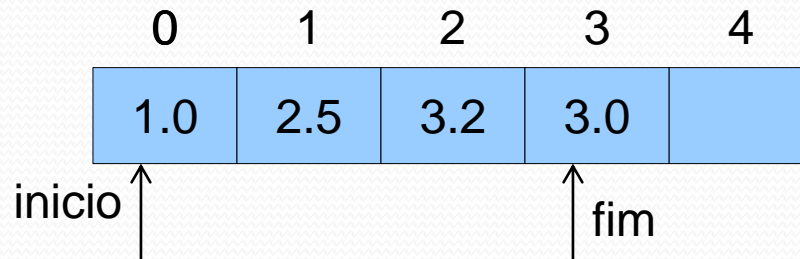
```
void enqueue (float f[], float valor, int &fim) {  
    if (fim == TAM - 1) { // testa fila cheia – é uma operação  
        cout << “ERRO : fila cheia.” << endl;  
        return; // sai da função  
    }  
    fim++; // ajusta fim  
    f[fim] = valor; // armazena o valor fim da fila  
}
```

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

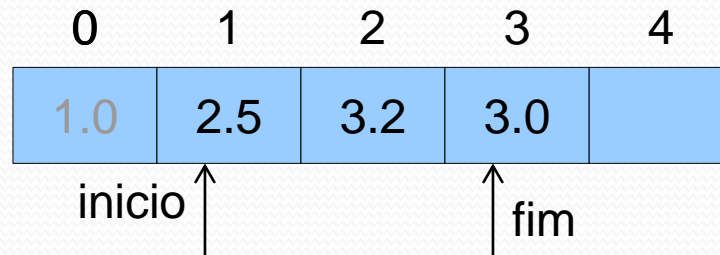
# Desenfileirar (dequeue)

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

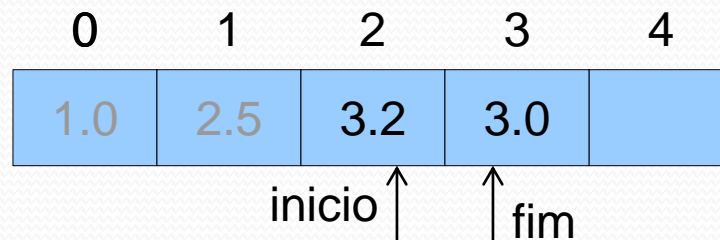
# Desenfileirar



- Desenfileirar



- Desenfileirar



- Na realidade a remoção de um elemento da fila é realizada apenas alterando-se a informação da posição do início.

# Código da Função Desenfileirar

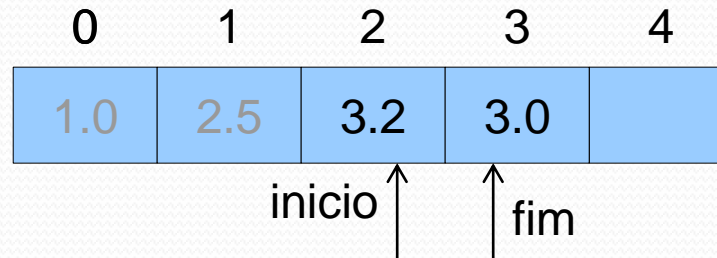
```
int dequeue (float f[], int &inicio, fim, float &valor) {  
    if (inicio > fim) { // testa fila vazia - operação  
        cout << "ERRO : fila vazia." << endl;  
        return 0; // sinaliza fracasso  
    }  
    valor = f[inicio]; // retira do início  
    inicio++; //ajusta o início  
   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

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

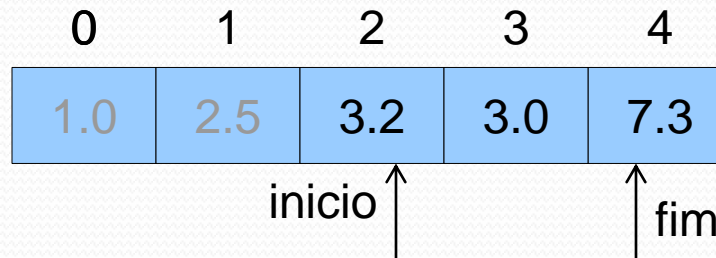
# Animação da Manipulação de uma Fila

<http://www.cosc.canterbury.ac.nz/mukundan/dsal/QueueAppl.html>

# Desenfileirar/Enfileirar



- Suponha agora que se deseje:
  - Enfileirar o 7.3



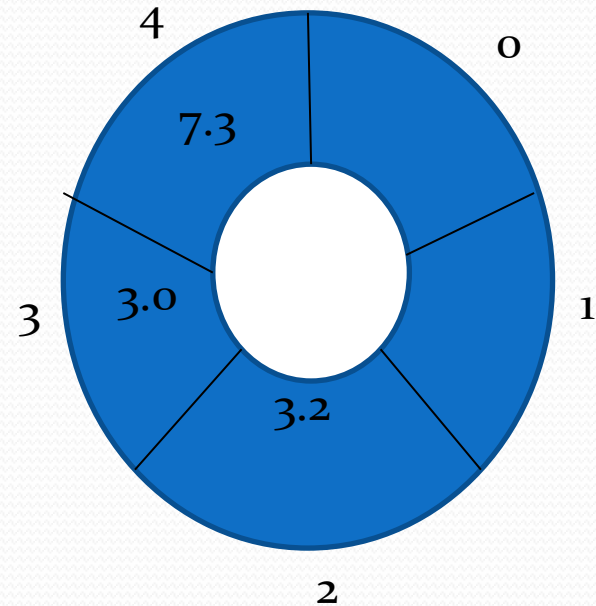
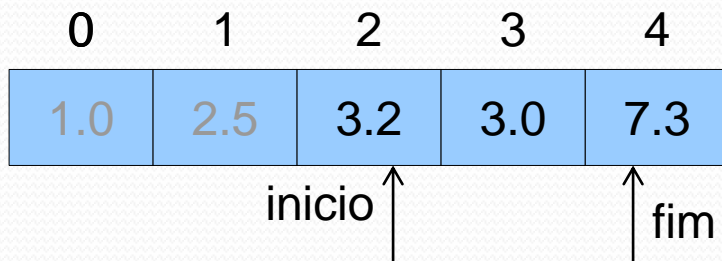
- E, enfileirar o 5.4?
- O vetor acabou, mas há espaço no início da fila!!!!

# Solução

- Deslocar os elementos para o início da fila (exercício)
- Uso de uma FILA CIRCULAR
  - Solução mais empregada
  - Tem menor custo computacional
  - Mais fácil implementação

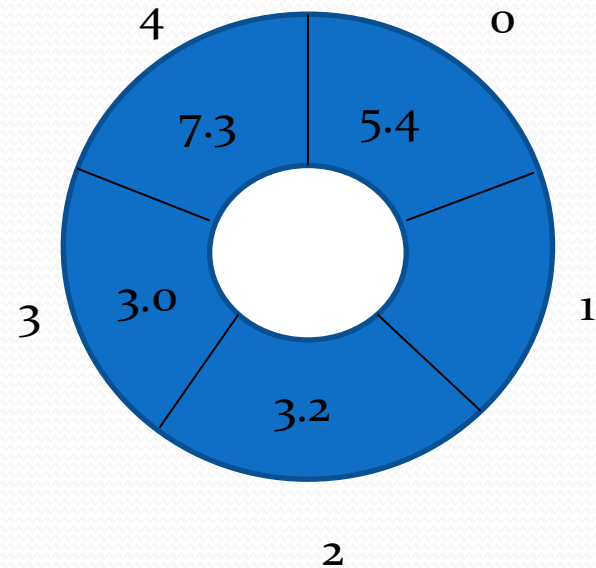
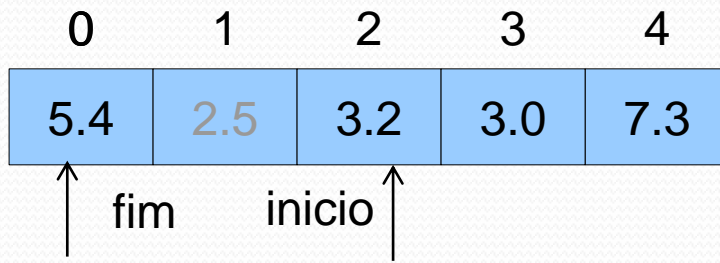


# Fila Circular



# Fila Circular

- enfileirar o 5.4



# Animação da Manipulação de uma Fila Circular

<http://www.cosc.canterbury.ac.nz/mukundan/dsal/CQueueAppl.html>

# Criar Fila Circular

- Para facilitar incluímos uma informação – quantidade de elementos existente na fila circular (*nItens*)
- 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 que está no início da fila
```

```
    fim, // armazena o índice do elemento que está no fim da fila
```

```
    nItens; // armazena a quantidade de elementos existente na fila
```

# Inicializar a Fila Circular

- Determina o status inicial da fila, a fim de prepará-la para a inserção de dados.
- Na 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.
- Para enfileirar, 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 da Função Enfileirar - Fila Circular

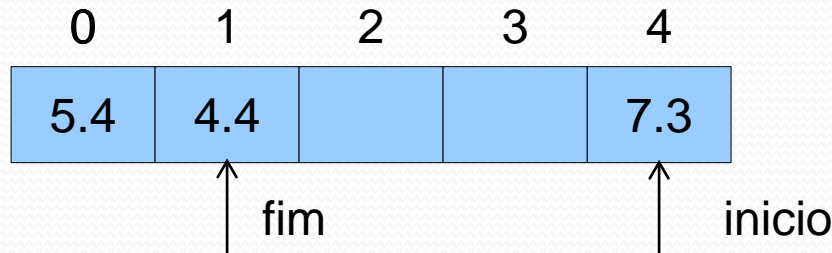
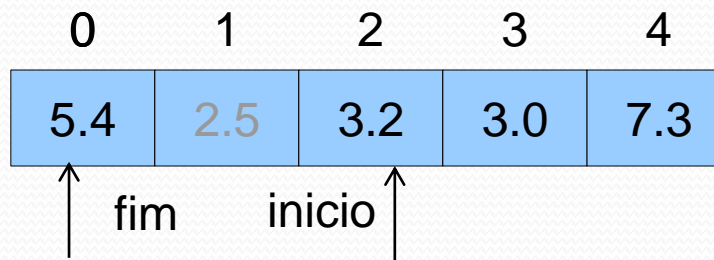
```
void enqueue (float f[], float valor, int &fim, int &nItens) {  
    if (nItens == TAM) { // testa fila cheia  
        cout << "ERRO : fila cheia." << endl;  
        return; // sai da função  
    }  
    fim++;    // ajusta fim  
    if (fim == TAM) { // estourou o vetor  
        fim = 0;  
    }  
    f[fim] = valor; // armazena o valor fim da fila  
    nItens++;  
}
```

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

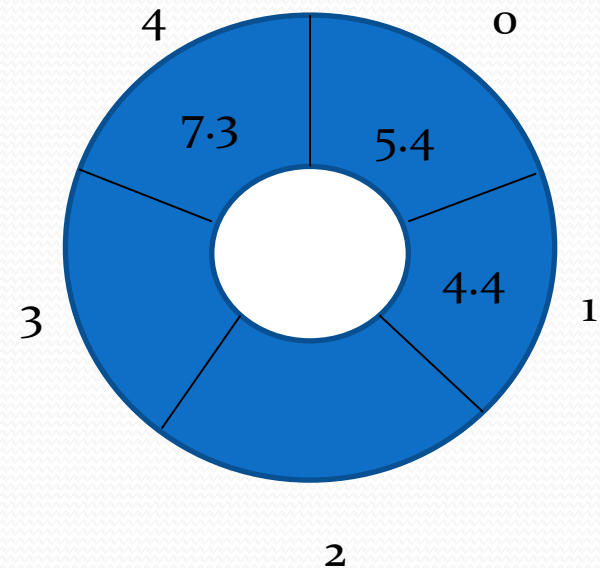
# Desenfileirar Elemento na Fila Circular

- Supondo as seguintes operações:

- Enfileira o 4.4
- Desenfileirar
- Desenfileirar



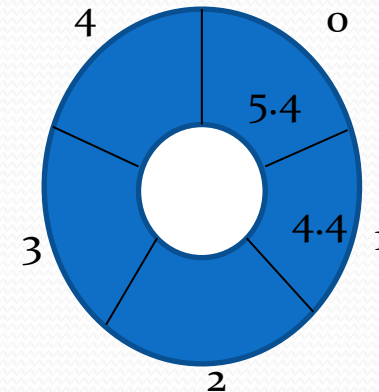
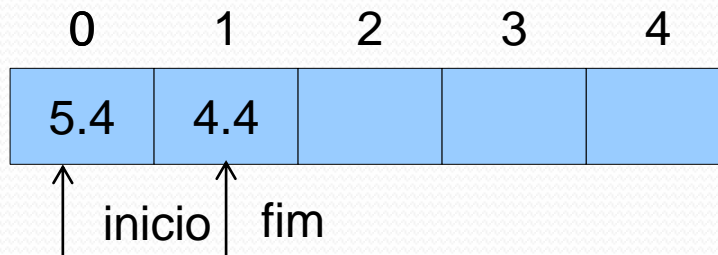
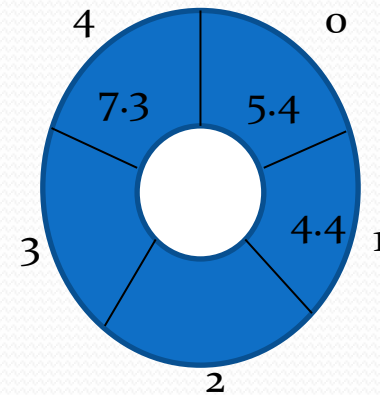
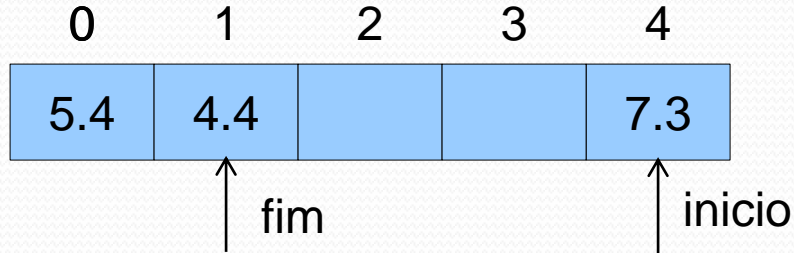
- Desenfileirar???





# Desenfileirar Elemento na Fila Circular

- Desenfileirar???



# Desenfileirar Elemento na Fila Circular

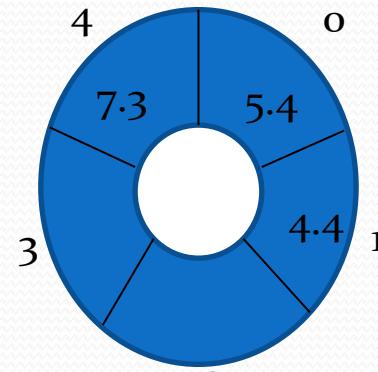
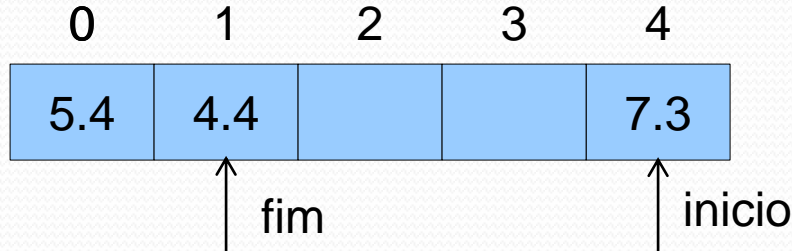
- Consiste em remover um valor do início da fila, desde que a fila não esteja vazia.
- Para desenfileirar, é preciso ajustar o início (controlando quando o elemento da última posição do vetor for removido, neste caso, o fim passa a ser a primeira posição do vetor).
- Parâmetros : vetor de elementos,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

# Código da Função Denfileirar - Fila Circular

```
void dequeue (float f[], int &inicio, int &nItens) {  
    if (nItens == 0) { // testa se a fila está vazia  
        cout << "ERRO : fila vazia." << endl;  
        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  
}  
• Chamada na main  
dequeue(fila, inicio, nItens);
```

# Mostrar Elementos da Fila Circular

- 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

# Código para Mostrar Elementos da Fila Circular

```
void displayQueue(float f[], int inicio, int fim, int &nItens) {  
    if (nItens == 0) { // testa se a fila está vazia  
        cout << "ERRO : fila vazia." << endl;  
        return; // sai da função  
    }  
    for (int cont=0, i= inicio; cont < nItens; cont++){  
        cout << f[i++] << " ";  
        if (i == TAM) {  
            i=0;  
        }  
    }  
    cout << "\n\n";  
}  
• Chamada na main  
  displayQueue (fila, inicio, fim, nItens);
```

# Outras Funções

- Verificar se a fila está vazia

**// retorna verdadeiro se a fila está vazia**

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

- Verificar se a fila está cheia

**// retorna verdadeiro se a fila está cheia**

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

# Exercício #1

Questões de Concurso (pilha e fila)

# Questão #1

- Considere a estrutura de dados fila, do tipo FIFO. Entidades são inseridas nessa estrutura com a operação `push( )` e removidas com a operação `pop( )`. A opção a seguir que mostra o conteúdo ordenado da fila após a sequência de operações

`push(8), push(7), push(5), push(2), pop( ), push(8), push(7), pop( ), push(5), push(2), pop( ), pop( )` é:

- (A) 8578
- (B) 8758
- (C) 8752
- (D) 2875
- (E) 2758



# Questão #2

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

# Questão #3

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;

# Questão #4

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.

# Questão #5

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.

# Questão #6

- 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.

# Questão #7

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.

# Questão #8

- 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.

# Questão #9

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 definição
  - 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).



# Exercício #2

- 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.