

**UNISINOS** - UNIVERSIDADE DO VALE DO RIO DOS SINOS  
CIÊNCIAS EXATAS E TECNOLÓGICAS – Curso: Informática / Ciência da Computação

## Programação II

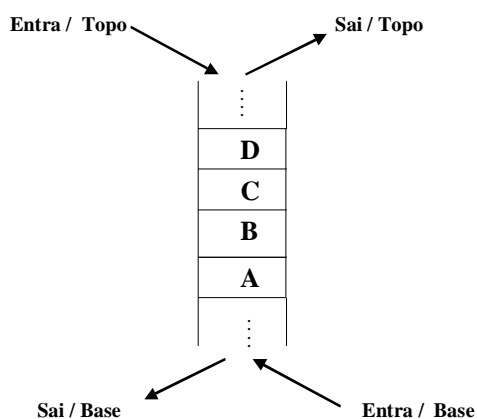
Disciplina: Linguagem de Programação PASCAL  
Professor responsável: *Fernando Santos Osório*  
Semestre: 2004/2  
Horário: 63

E-mail: *osorio@exatas.unisinos.br*  
Web:  
*http://www.inf.unisinos.br/~osorio/prog2.html*  
Xerox : *Pasta 54 – LAB. II (Xerox do “Alemão”)*

### Deque – Double Ended QUEues

As estruturas de dados podem ser organizadas de formas diferentes e especiais, de acordo com a **maneira com que os dados são inseridos e retirados**. Um tipo particular de estruturas de dados, é o **DEQUE**, onde a inserção pode ocorrer no início ou no final do deque (início/final do vetor) e a retirada dos dados também pode ocorrer no início ou no final do deque (início/final do vetor). Note que uma característica importante dos deque é que as operações de inserção e remoção de elementos ocorrem sempre nas extremidades, onde a princípio não devem ser realizadas inserções ou remoções de dados que se encontram no interior do mesmo. Esta noção de início/fim do vetor pode ser um pouco particular se considerarmos que usualmente os deque, assim como as filas circulares, serão implementados em uma estrutura de dados chamada “*deque circular*”, que descreveremos a seguir. Este tipo de estrutura de dados, os deque, podem “simular” tanto uma fila como uma pilha, pois possuem as operações elementares que permitem realizar as inserções e remoções que caracterizam estes dois tipos de estruturas de dados. Os deque são também muito usados em diversos tipos de aplicações computacionais, dada a sua flexibilidade no que diz respeito a inserção e remoção dos dados.

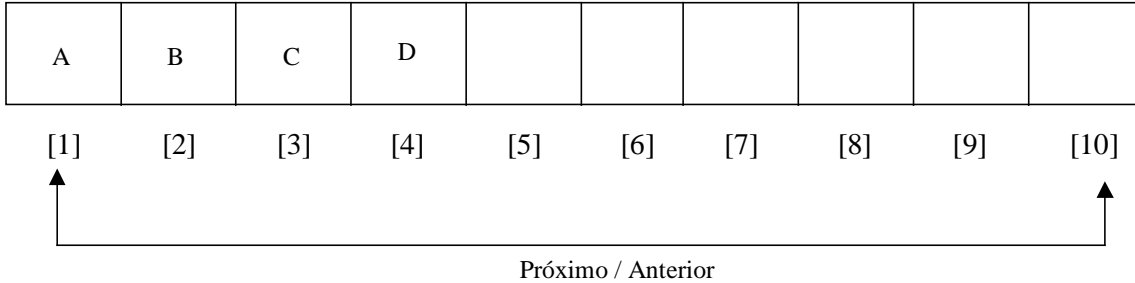
Manipulação de Deques => Entra = Insere no início ou final do deque (em uma das extremidades)  
<= Sai = Retira do início ou final do deque (em uma das extremidades)



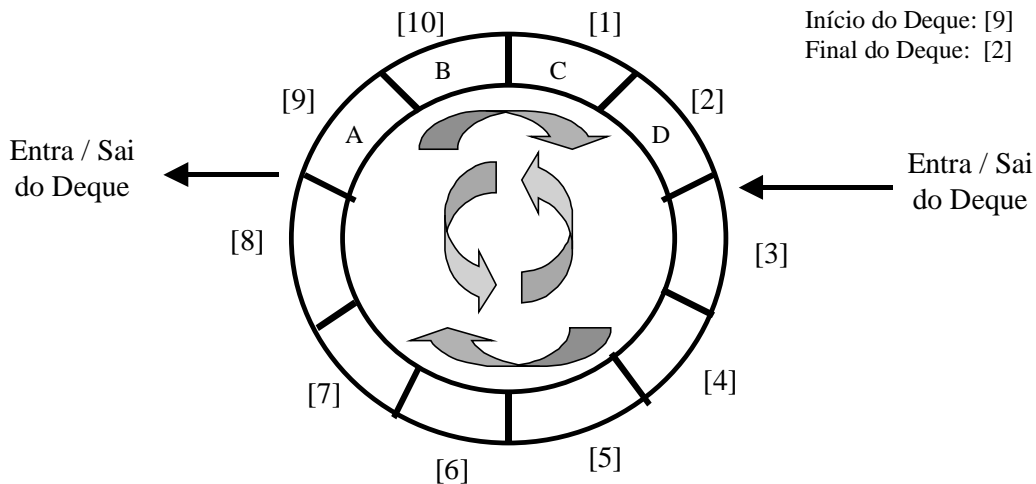
Os deque podem usar vetores estáticos para armazenar os dados, mas como pode ser constatado na figura acima, se formos inserindo e retirando dados de um vetor estático, podemos rapidamente alcançar o “topo” ou a “base” deste vetor. Uma solução para este problema é a criação dos chamados DEQUES CIRCULARES.

Um DEQUE CIRCULAR é uma estrutura de dados que simula de modo virtual um deque com duas extremidades onde o início e o fim do vetor são unidos. A figura abaixo demonstra um exemplo de como poderia ser visualizado um deque circular usando um vetor de 10 elementos:

Deque Circular => Deque: Array [1..10] of Integer;      Início do Deque: [1] - Final do Deque: [4]



Deque Circular => Deque: Array [1..10] of Integer;      Representação “Circular” do vetor



Portanto, em um DEQUE CIRCULAR o elemento seguinte ao último elemento do vetor é virtualmente o primeiro elemento deste vetor, assim como, o elemento vizinho e anterior ao primeiro elemento deste vetor passa a ser o último elemento do vetor. Com isto, não teremos mais problemas relativos ao deslocamento do início/fim do deque que ocorre quando os elementos são inseridos ou retirados do deque. O deque, por aceitar inserção e remoção nas duas extremidades deve permitir um “giro” do início nos dois sentidos (horário e anti-horário), assim como o final que também deve se deslocar nos dois sentidos. O uso de um deque circular, assim como no caso das filas circulares, vai também implicar na necessidade de diferenciar entre deque completamente vazio e deque completamente cheio.

Vamos definir a seguir um conjunto de **rotinas genéricas** para manipulação de deques circulares, que simplificarão o seu uso e sua adaptação para uso em outras aplicações.

Definição dos Dados:

```
Const
    DEQUE_TAM = 10;
```

Type

```
TDeque_Dado = Integer;  
TDeque = Record  
    Dado: Array [1..DEQUE_TAM] of TDeque_Dado;  
    Inicio, Fim: Integer;  
    Qtde: Integer;  
End;
```

Definição das Rotinas:

**Procedure Deque\_Inicializa (Var D: TDeque);**

**Function Deque\_Inserir\_Inicio (Var D: TDeque; Dado: TDeque\_Dado): Boolean;**

**Function Deque\_Inserir\_Final (Var D: TDeque; Dado: TDeque\_Dado): Boolean;**

**Function Deque\_Retira\_Inicio (Var D: TDeque; Var Dado: TDeque\_Dado): Boolean;**

**Function Deque\_Retira\_Final (Var D: TDeque; Var Dado: TDeque\_Dado): Boolean;**

**Procedure Deque\_Exibe (Var D: TDeque);**

**Procedure Deque\_Esvazia (Var D: TDeque);**

**Function Deque\_Vazio (D: TDeque): Boolean;**

**Function Deque\_Cheio (D: TDeque): Boolean;**

**Function Deque\_Quantidade (D: TDeque): Integer;**

**Procedure Deque\_Grava (Var ArqTxt: Text; D: TDeque);**

**Function Deque\_Le (Var ArqTxt: Text; Var D:TDeque);**

**Function Deque\_Procura\_Dado (Var D: TDeque; Dado: TDeque\_Dado;  
 Var Posicao: Integer): Boolean;**

## EXERCÍCIOS:

- 1) Faça a implementação das rotinas genéricas conforme foi descrito acima, que irão permitir a manipulação de um deque, com inserção e remoção tanto no início quanto no final do deque.
- 2) Faça um programa usando as rotinas genéricas de manipulação de deque, que permita manipular um deque de números reais. Implemente um menu, contendo as seguintes opções para o usuário: (1) Inserir um valor no início do deque; (2) Inserir um valor no final do deque; (3) Retirar um valor do início do deque; (4) Retirar um valor do final do deque; (5) Listar o conteúdo do deque na tela; (6) Remover todos os valores do deque; (7) Fim da execução do programa.
- 3) Implemente novamente as rotinas de manipulação de pilha e de fila, usando como rotinas de base as rotinas genéricas de manipulação de deques. As rotinas de inserir/retirar pilha/fila devem conter chamadas para as rotinas de inserção e remoção de dados no deque.