**UNISINOS - UNIVERSIDADE DO VALE DO RIO DOS SINOS**
**CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS (C6/6) – Curso: Informática**

# ***PROGRAMAÇÃO I – AULA 11***

**Disciplina: Linguagem de Programação PASCAL**
**Professor responsável:** ***Fernando Santos Osório***
**Semestre: 2001/2**
**Horário: 63**

**E-mail:** ***osorio@exatas.unisinos.br***
**Web:**
***http://inf.unisinos.br/~osorio/prog1.html***
**Xerox : *Pasta 54* (Xerox do C6/6)**

## 1. Vetores Unidimensionais - ARRAY:

Variáveis estruturadas, compostas por um conjunto de variáveis iguais:
- Estruturas homogêneas = Série de elementos iguais => *Vetores / Array / Matriz*

Os arrays unidimensionais (vetores) ou arrays multidimensionais (matrizes) são compostos por um número fixo de elementos do mesmo tipo, onde cada elemento é acessado individualmente através do seu índice. O índice é um valor inteiro e positivo que permite indicar qual dos 'N' elementos do vetor que estamos fazendo referência em um certo momento. Os índices são indicados entre colchetes, por exemplo, Vetor[6] usualmente indica o acesso ao sexto elemento de uma variável chamada de 'Vetor'.

Os vetores servem para simplificar a declaração de várias variáveis iguais, por exemplo: imagine um programa que manipule com 10 notas de um aluno... ao invés de termos variáveis do tipo Nota1, Nota2, Nota3, Nota4, ..., Nota10, bastaria termos um único vetor chamado Nota, com índices que variam de 1 até 10. Assim sendo poderíamos acessar diretamente as notas através de Nota[1], Nota[2], ... Nota[10]. A vantagem é que o nome da variável fica fixo, enquanto podemos fazer variar apenas o índice, que pode inclusive ser dado por uma outra variável. Para ler as 10 notas poderíamos usar um FOR, onde o contador seria usado como índice do vetor. Exemplo:

```
FOR Cont := 1 TO 10  DO  READLN ( Nota[Cont] );
```

* SINTAXE:

***A declaração é feita assim...***

<nome_do_vetor> : ARRAY [ <início> **..** <fim> ] OF <tipo_da_variável> ;

***Usando os vetores...***

<nome_do_vetor> [ <índice> ] := <valor>;

<nome_da_variável> := <nome_do_vetor> [ <índice> ] ;

Write (<nome_do_vetor> [ <índice> ] ) ;

Readln (<nome_do_vetor> [ <índice> ] ) ;

**Exemplos:**

```
PROGRAM Exemplo;
VAR
    Nota   :  ARRAY  [ 1.. 10 ]  OF  REAL;
    Indice :  INTEGER;
BEGIN
    FOR  Indice := 1  TO  10
    DO  BEGIN
            Writeln ('Entre com a nota nro. ', Indice, ' : ' );
            ReadLn ( Nota[Indice] );
        END;
    WriteLn ('A primeira nota fornecida foi :  ', Nota [1] );
    WriteLn ('A última nota fornecida foi: ', Nota[10] );
END.
```

**Podemos usar também um array de registros...**

```
TYPE
    Reg_Pessoa  =  RECORD
                     Nome      : String;
                     Sexo      : Char;
                     Endereco  : String;
                     Telefone  : String;
                     Salario   : Real;
                     Disciplina: ARRAY [1..10]  OF  String;
                   END;
...
VAR
    Professor : ARRAY [ 1 .. 100 ]  OF  Reg_Pessoa;
    Aluno     : ARRAY [ 1 .. 30 ]   OF  Reg_Pessoa;
    GrauA     : ARRAY [ 1 .. 30 ]   OF  Real;
    GrauB     : ARRAY [ 1 .. 30 ]   OF  Real;
    Media     : ARRAY [ 1 .. 30 ]   OF  Real;
...
BEGIN
    Professor[1].Nome:='Fulano';
    Professor[1].Disciplina[1] := 'Matemática';
    ...
    Aluno[ Atual ].Nome := 'Estudantino';
    Aluno[ Atual ].Disciplina[ Indice ] := Aluno [ Outro ].Disciplina[ 1];
    ...
    FOR  Atual := 1  TO 30
    DO  Media[ Atual ] := ( GrauA[Atual]  +  GrauB[Atual] ) / 2.0;
    ...
END.
```

**Observações:**

- Em Pascal, podemos definir arrays que não começam obrigatoriamente na primeira posição (índice inicial diferente de 1). Exemplo: VAR TABELA:ARRAY [10..30] OF Real ;

- Quando manipulamos os arrays, devemos trabalhar com um elemento por vez, ou seja, devemos ler, escrever ou atribuir valores indicando o índice de qual elemento especificamente estamos nos referenciando;

- Quando usamos uma variável como índice de um array, um erro muito comum é o estouro da faixa especificada, ou seja, suponha que um array 'Vetor' tenha sido definido com índices que variam de 1 até 10, e que em nosso programa vamos usar o seguinte comando 'Vetor[Indice] := 3;'. Você consegue imaginar o que acontecerá se o valor da variável 'Indice' for um valor superior à 10, como por exemplo 45 ? TEREMOS GRANDES PROBLEMAS (O programa deve "enlouquecer" num caso como este).

- Uma solução para evitar este tipo de erros é o uso de uma opção de compilação do Turbo Pascal que permite verificar, sempre antes de acessar um dado em um vetor, se o índice deste vetor é valido. Esta opção se encontra no menu "OPTIONS – COMPILER", onde você deve marcar um 'x' na opção "RANGE CHECKING".

# EXERCÍCIOS – AULA 11

1. Faça um programa que crie um vetor de inteiros de 10 posições, leia os valores deste vetor e exiba o vetor na tela de trás para frente (na ordem inversa na qual os dados foram fornecidos). Faça um programa onde o vetor possa ser facilmente alterado de 10 posições para 30 posições, por exemplo.

2. Faça um programa que leia 10 números reais. Troque os valores do vetor dois à dois, ou seja, os elementos cujos índices são pares do vetor devem ser trocados com os elementos de índice ímpar vizinhos a estes. Exiba na tela o vetor resultante desta troca dos valores, e também a soma total dos elementos contidos neste vetor.

3. Faça um programa que crie um vetor de 26 elementos do tipo caracter. Cada elemento do vetor deve conter uma letra do alfabeto, onde o seu índice é dado pela ordem da letra no alfabeto (exemplo: 'A' = 1, 'B' = 2, 'C' = 3, ...) Exibir os elementos deste vetor que possuam o índice par na tela.

4. Use o programa anterior para copiar do $10^{o}$ ao $20^{o}$ caracter para uma string. Exibir o conteúdo desta string na tela. Pedir para o usuário entrar com uma nova string, cujos caracteres deverão ser copiados para dentro do vetor a partir da $8^{a}$ posição deste. Exibir como ficou o vetor após a sua alteração.

5. Faça um programa para ler 10 números inteiros, colocando-os em um vetor. Crie dois outros vetores, um deles onde você deve armazenar apenas os números positivos (maior ou iguais a zero) e outro onde você deve armazenar apenas os números negativos. Coloque os números positivos no vetor de positivos e os números negativos no vetor de negativos. Mostre na tela como ficaram os três vetores.

6. Baseado no programa anterior, faça com que o novo vetor de números positivos tenha os números iguais à zero desprezados (este valores devem ser "eliminados" do novo vetor). Mostre na tela os três vetores, indicando a quantidade de zeros que havia vetor de positivos e a quantidade total de números atualmente armazenados no vetor de positivos (sem os zeros) e no vetor de negativos.

7. Faça um programa que leia dois vetores de números compostos por 5 elementos que são fornecidos de maneira ordenada (números em ordem crescente). Crie um terceiro vetor que é a união dos dois primeiros vetores, sendo que este novo vetor de 10 elementos também deve ser um vetor onde os seus elementos estão ordenados. Exiba os vetores e a soma total dos elementos para cada um dos três vetores.

8. Altere o programa anterior para desprezar os números iguais, caso estes existam. Sendo assim o vetor final não deve possuir números iguais armazenados no vetor.

9. Faça um programa para o controle de notas dos alunos de uma escola (similar ao exercício desenvolvidos nas aulas anteriores), usando *registros para descrever os dados relativos ao aluno* e criando uma *sub-rotina para a entrada de dados e uma outra sub-rotina para o cálculo da média*. Ler o nome do aluno e suas duas notas A e B, e após calcular a média ponderada entre estas notas (A tem peso 1 e B tem peso 2). Repetir este procedimento para uma turma composta por 10 alunos. Ao terminar a entrada de todos os dados, limpar a tela e exibir as notas e as respectivas médias dos 10 alunos na tela. ***Não utilize variáveis globais!***