

# Vetores e Strings

## 1. Vetores

Vetores nada mais são que matrizes unidimensionais. É importante notar que vetores, matrizes bidimensionais e matrizes de qualquer dimensão são caracterizadas por terem todos os elementos pertencentes ao mesmo tipo de dado. Forma geral para se declarar um vetor:

***tipo\_da\_variável nome\_da\_variável [tamanho];***

# Vetores e Strings

## 1. Vetores (continuação)

Exemplo:

```
float exemplo [20];
```

Na linguagem C a numeração dos índices começa sempre em zero. Isto significa que, no exemplo acima, os dados serão indexados de 0 a 19. Para acessá-los vamos escrever:

```
exemplo[0]
```

```
exemplo[1]
```

```
.
```

```
.
```

```
.
```

```
exemplo[19]
```

# Vetores e Strings

## 1. Vetores (continuação)

Mas ninguém o impede (programador) de escrever:

```
exemplo[30]  
exemplo[103]
```

Pois o C não verifica se o índice que você usou está dentro dos limites válidos. Este é um cuidado que você deve tomar. Se o programador não tiver atenção com os limites de validade para os índices ele corre o risco de ter variáveis sobreescritas ou de ver o computador travar. Bugs terríveis podem surgir.

# Vetores e Strings

## 1. Vetores (continuação)

Exercício: Construa um programa que declare um vetor de inteiros com 10 elementos, o inicialize, com números fornecidos pelo usuário, através da entrada padrão e que através de uma pesquisa nos elementos do vetor, retorne na saída padrão os elementos de menor e maior valor, respectivamente.

# Vetores e Strings

## 1. Vetores (continuação)

OBS.: Um vetor pode ser inicializado na declaração, exemplo:

```
int vetor[10]={0,1,2,3,4,5,6,7,8,9};
```

E ainda pode-se deixar em aberto o número de elementos, que será preenchido pelo números de elementos na inicialização, que ocorre no momento da declaração.

```
int vetor[]={0,1,2,3,4,5,6,7,8,9};
```

~~int vetor[];~~

# Vetores e Strings

## 2. Vetores Multidimensionais

*tipo\_da\_variável nome\_da\_variável [d1][d2]...[dn];*

Exemplo:

*float matriz [3][4]={1,2,3,4,5,6,7,8,9,10,11,12};*

*float matriz [ ][2]={1,2,3,4,5,6,7,8,9,10,11,12};*

~~*float matriz [ ][ ]={1,2,3,4,5,6,7,8,9,10,11,12};*~~

## Vetores e Strings

Exemplo:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int matriz1[2][3],matriz2[3][2],i,j;
```

```
    for (i=0;i<2;i++)
```

```
        for (j=0;j<3;j++)
```

```
        {
```

```
            printf ("\nEntre com matriz1[%d][%d]=",i,j);
```

```
            scanf ("%d",&matriz1[i][j]);
```

```
            printf ("\nEntre com matriz2[%d][%d]=",j,i);
```

```
            scanf ("%d",&matriz2[j][i]);
```

```
        }
```

```
    }
```

# Vetores e Strings

## 2. Vetores Multidimensionais (continuação)

Exercício: Construa um programa, na linguagem C, que declare uma matriz 8x3 de números em ponto flutuante, a inicialize com valores fornecidos pelo usuário através da entrada padrão e a retorne na saída padrão com o layout a seguir:

```
|      X.XX      X.XX ...      X.XX |
|-----|-----|.....|
| 10 X.XX  10 X.XX ...      X.XX |
|      .        .        .        . |
|      .        .        .        . |
|      .        .        .        . |
```