



PROGRAMAÇÃO DE COMPUTADORES (PC)

ENGENHARIA DA COMPUTAÇÃO

Vetores

Profa. Dra. Lúcia F. A. Guimarães



- **Vetores**

- Em Computação, vetores são utilizados para armazenar na memória um número finito de valores (numéricos ou não-numéricos) do MESMO TIPO.
- São Identificados por um *tipo, nome e tamanho*, podendo ser inicializados em suas declarações ou não.
- `int a[5] = {1, 2, 4, 8, 16};`

1	a[0]
2	a[1]
4	a[2]
8	a[3]
16	a[4]



- **Vetores**

- O tipo representa o conteúdo que armazena, podendo ser inteiros, reais, caracteres, “strings”, estruturas e classes. Exemplos:
 - `int u[5];`
 - `float v[3];`
 - `char w[4];`
 - `char x[6][max_sz_string];` (em C não temos strings, mas sim uma sequência de caracteres).
 - `minhaStruct y[7];`
 - `minhaClasse z[9];`



- **Inicialização:**

- **Por inicialização:**

```
int vet[3] = {1, 7, -3};
```

- **Por atribuição:**

```
int vet[3];
```

```
vet[0] = 1;
```

```
vet[1] = 7;
```

```
vet[2] = -3;
```

- **Por leitura:**

```
int vet[3];
```

```
for (int i = 0; i < 3; i++) std::cin >> vet[i];
```

- **Vetores**

- Cada elemento ocupa uma posição (índice) específica na memória, que pode ser acessada através do operador colchetes ([]).
- Em C, os índices de um vetor se iniciam com 0 e vai até (tamanho)-1.
 - Portanto um vetor de 6 posições terá os índices variando de 0 a 5.

INDICE	0	1	2	3	4	5
CONTEÚDO	9	7	10	5	8	3



• Operador colchetes []

- O operador colchetes serve para acessar a posição de um vetor, que pode operar de duas formas:
 - **Get:** o operador colchetes **retorna** o conteúdo da posição indicada. Funciona como *get* sempre que **não** estiver envolvido com um operador de atribuição de conteúdo, como, por exemplo, os operadores = e >> (no caso de `std::cin`);
 - `std::cout << vet[2];` (Funcionando como *get*)
 - **Set:** o operador colchetes indica a posição em que seu conteúdo deve ser **atribuído**. Funciona como *set* sempre que estiver envolvido com um operador de atribuição de conteúdo.
 - `vet[2] = “algo do tipo armazenado”;` (Funcionando como *set*)



- **ATENÇÃO!!!**

- A linguagem C NÃO FAZ NENHUMA VERIFICAÇÃO nos índices de um vetor.
- O acesso a uma posição fora dos limites de um vetor pode acarretar na perda definitiva de informações importantes do sistema operacional.



- **Exercícios**

1. Fazer um programa que faz a leitura de dois vetores X e Y, ambos com N reais (N deve ser lido tb), com tamanho máximo igual a 20. Efetue a multiplicação dos respectivos pares de elementos de X por Y, gerando um terceiro vetor Z. Imprima os vetores X, Y e Z.
2. Fazer um programa que faz a leitura de um vetor X com N inteiros, com tamanho máximo igual a 20 e N deverá ser lido. Multiplique cada elemento do vetor por um inteiro K (**K também deverá ser lido**). Imprimir o vetor lido, o valor da variável K e o vetor multiplicado por K.



- **Exercícios**

3. Fazer um programa que faz a leitura de um vetor X com N reais , com tamanho máximo igual a 20 e N deverá ser lido. Calcule o somatório de todos os elementos do vetor. Imprimir o vetor lido e o valor do somatório.
4. Fazer um programa que faz a leitura de um vetor X , com N inteiros, com tamanho máximo igual a 20 e N deverá ser lido. Gere um segundo vetor com os valores do vetor X em ordem inversa. Imprimir o vetor X e vetor com a ordem inversa.



- **Exercícios**

5. Ler um vetor de inteiros 10 posições. Escreva a seguir o valor e a posição do maior e menor elementos lidos.
6. Fazer um programa em C para ler um vetor de inteiros positivos de 15 posições. Imprimir a quantidade de números pares e a quantidade de múltiplos de 5
7. Ler um vetor A com 10 elementos inteiros correspondentes as idades de um grupo de pessoas. Escreva um programa que determine e escreva a idade média dos elementos lidos e quantos elementos são menores que a média.