SENAI CIMATEC

SISTEMA FIEB

TECNOLOGIA E INOVAÇÃO EM PROL DA INDÚSTRIA

SENAI CIMATEC

SISTEMA FIEB

Técnico em Informática

Programação de aplicativos – 140h

Profª: Francisleide Almeida

Variáveis Indexadas

Variáveis indexadas

- Existem diversas situações na área de processamento de dados que existe a necessidade de se armazenar um grande conjunto de dados na memória RAM do computador. Muitas dessas situações os dados estão relacionados.
- Ex:
  - Ler todos os nomes dos alunos da turma de LPE e ordena-los alfabeticamente.
  - Ler todas as notas dos alunos desta turma e classifica-los imprimindo os que tiveram média acima de sete.

- Nestes casos é inviável utilizarmos uma variável para cada aluno ou nota, sendo assim, as linguagens de programação incluem um mecanismo chamado **variável indexada** que permite realizar este armazenamento com uma única variável. O termo *indexada* provém da maneira como esta individualização é feita: por meio de **índices**.

Variáveis indexadas

- Variáveis indexadas são denominadas arrays que se dividem em duas partes:
  - Arrays Unidimensionais : também chamadas vetores (um único índice é usado);
  - Arrays Multidimensionais : duas ou mais dimensões (possui dois ou mais índices).
- ***Operações Básicas com Variáveis Indexadas***

- Do mesmo modo que acontece com variáveis simples, também é possível operar com variáveis indexadas. Contudo, não é possível operar diretamente com o conjunto completo, mas com cada um de seus componentes isoladamente. O acesso individual a cada componente de um conjunto é realizado pela especificação de sua posição no mesmo por meio de um ou mais índices (no caso de uma matriz).

Variáveis indexadas

- **Exemplo 01:** Ler três números inteiros, achar a média e imprimir os números maiores que a média **VARIÁVEIS INDEXADAS**

Solução Tradicional:

```
Algoritmo <Programa_Imprime>
Início
        Inteiro A1, A2, A3;
        Real Media;

        Ler A1;
        Ler A2;
        Ler A3;
        Media ← (A1 + A2 + A3) / 3;
        Se (A1 > Media) então
             Escreva A1;
        Fim_Se
        Se (A2 > Media) então
             Escreva A2;
        Fim_Se
        Se (A3 > Media) então
             Escreva A3;
        Fim_Se
Fim
```

Modelo de Memória

A1	A2	A3	Media
5	7	8	6.6

Obs.: Imagine o mesmo exercício para 100 valores.

- O uso de variáveis indexadas nos dá a possibilidade de combinar o nome de uma variável com um índice numérico. O nome da variável associado com o valor numérico está relacionado a uma posição de memória.

```
Algoritmo <Programa_10>
Início
        Real X1, X2, X3, ..., X10, Media;

        Ler X1;
        Ler X2;
        Ler X3;
             :
             :
        Ler X10;
        Media ← (X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10) / 10;
        Escreve Media;
        Escreve X1;
        Escreve X2;
             :
             :
        Escreve X10;
Fim
```

```
1ª Entrada de Dados
Aux[0] = 1;
Aux[1] = 5;
Aux[2] = 3;
        :
        :
Aux[9] = 45;

2ª Entrada de Dados (Mudança de Valores)
Aux[0] = 15;
Aux[1] = 25;
Aux[9] = 2;
```

Vetores e Matrizes

- Neste tipo de estrutura, os valores armazenados devem pertencer ao mesmo tipo.

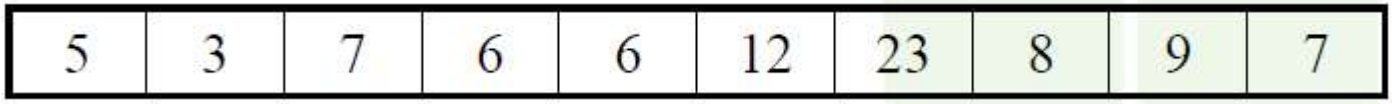
- •Entre outros nomes que estas estruturas recebem, iremos chamá-las de **Vetores** e **Matrizes**.

Vetores

- São estruturas lineares e estáticas, ou seja, são compostas por um número finito e pré-determinado de valores.

vetor1[5 3 7 6 6 12 23 8 9 7]

vetor1



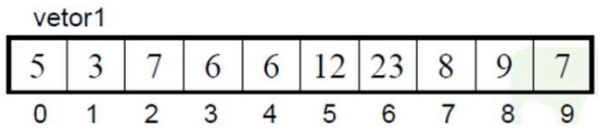
Vetores

- Estrutura homogênea e estática
  - Unidimensional
- Exemplo:
  - Prédio com um apartamento por andar
- Todos elementos pertencem ao mesmo tipo de dado.;

Posicionamento no vetor

- Levando em consideração que a primeira posição do vetor seja 0, teremos

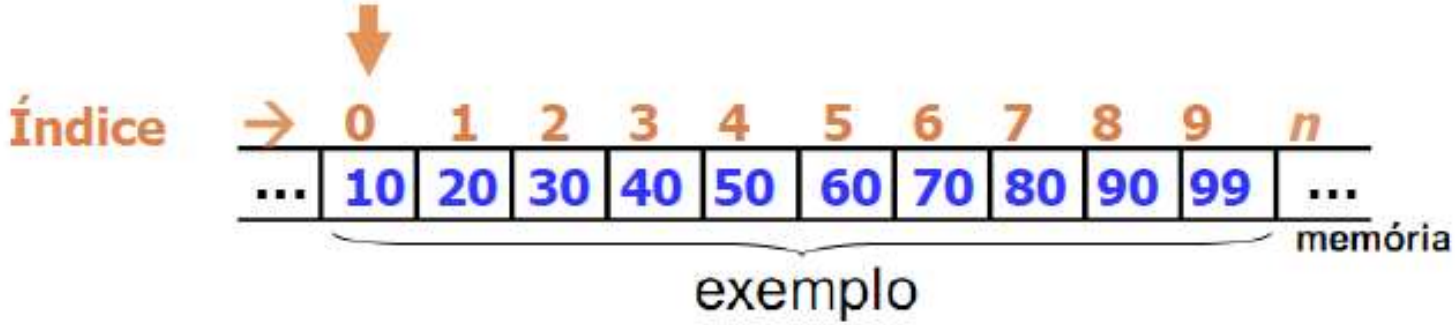
vetor1[0] = 5
vetor1[1] = 3
vetor1[2] = 7
vetor1[3] = 6
vetor1[4] = 6
...
vetor1[9] = 7



- Para fazer referência a um valor de um elemento contido em um vetor, usamos a notação vetor[índice], que serve tanto para obter quanto para definir o valor de um elemento específico, dada sua posição Note que os elementos são numerados a começar do zero, e, portanto, se o número de elementos é , o índice ou posição do último elemento será N - 1.

Posicionamento no vetor

- Índices (iniciam em "*0*", até "*n*");
- Índices utilizados para Recuperar/Inserir valores.



- Forma geral para se declarar um vetor:
- **tipo_da_variável nome_da_variável** [tamanho];

Posicionamento no vetor

- **Fique atento:**
  - Os limites do vetor:
    - Ex:

```
int vetorDois[10];
int x; vetorDois[11] = 32;
x = vetorDois[13];
```

Inicialização do vetor

- Um vetor pode ser inicializado na declaração:

```
int vetor[5] = {1,5,7,8,9}
```

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

```
int vetor[] = {1,5,7,8,9}
```

Vetores

- Valores das posições podem ser modificados no programa:

```
#include <stdio.h>
int main()  {
   int sal[]={0,1,2,3,4};
   sal[0]=100;
   sal[1]=300;
   sal[3]=900;
  printf ("%d\n",sal[0]);
 printf ("%d\n",sal[1]);
 printf ("%d\n",sal[2]);
 printf ("%d\n",sal[3]);
 printf ("%d\n",sal[4]);
    getchar();
 return (0);  }
```

Exercício1

- Construa um programa que declare e receba um vetor de inteiros com 10 elementos com números fornecidos pelo usuário, através da entrada padrão e depois exiba os índices e seus valores armazenados.

Exercício2

- Construa um programa que declare e receba um vetor de inteiros com 10 elementos e que o conteúdo de cada posição do vetor será o seu índice ao quadrado. Criei um ***for*** para receber o conteúdo e outro para imprimir.

Exercício3

- Faça um programa que leia um vetor tamanho 10 com dados obtidos a partir do usuário e exiba os valores recebidos de forma invertida.

Exercício4

- Faça um programa que leia um vetor de 10 posições e crie um segundo vetor substituindo os valores negativos por 1. Exiba os resultados do segundo vetor.