



Algoritmos e Programação II

Prof. Joilson dos Reis Brito

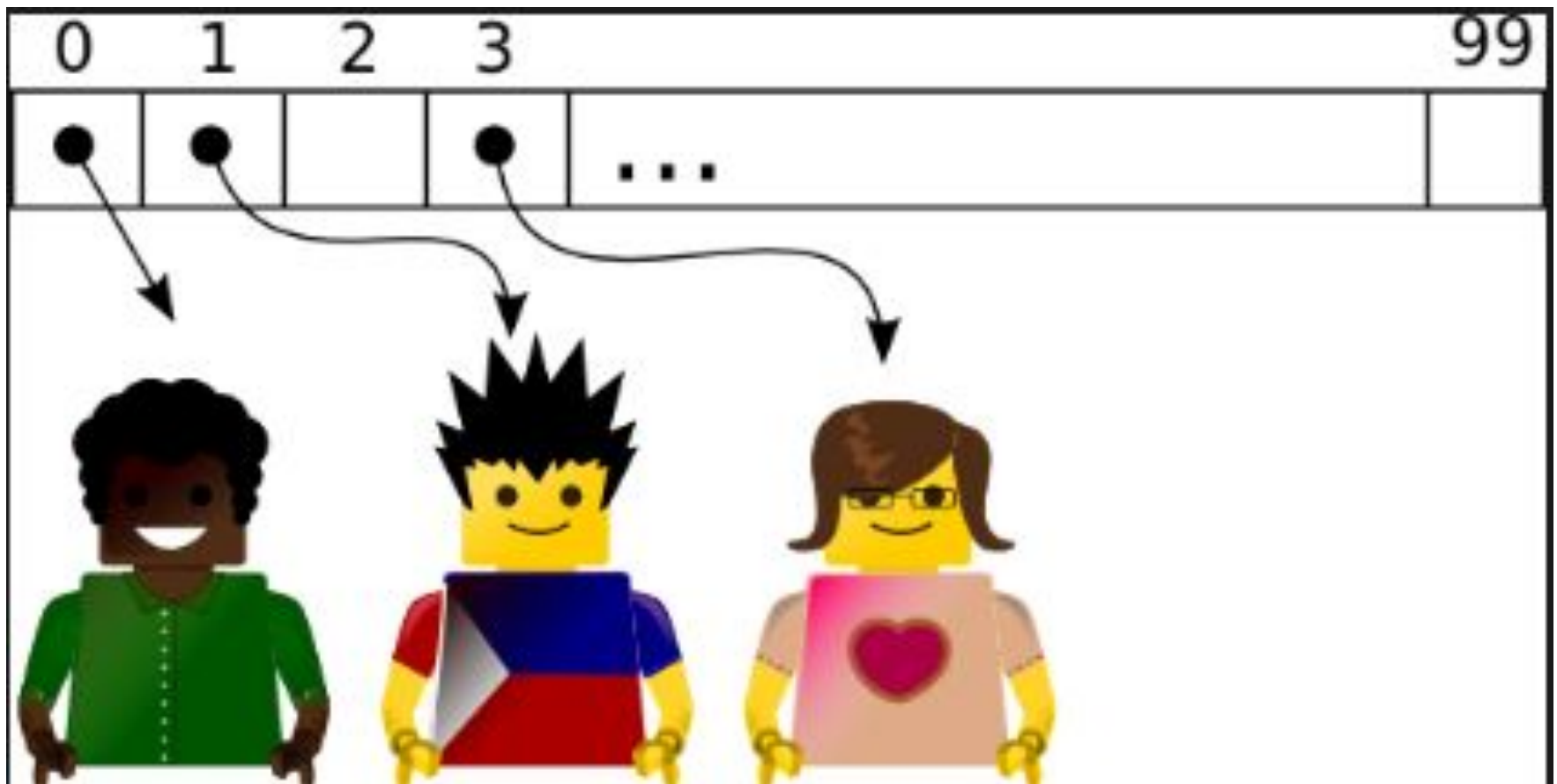
Agenda da Aula

- Variável Composta Homogênea – Vetor
- Exemplos
- Exercício

Monitoria

- 18h às 19h 3a e 5a

VETOR (ARRAY)



Vetor - Conceito

Na disciplina de Algoritmos e Programação I, apresentamos o conceito de Variável que é uma posição na memória do computador que pode armazenar apenas um valor que pode ser numérico, caracter ou string e que recebe um nome (identificador) para poder ser utilizada.

```
float PrecoProduto;
```

```
int Idade;
```

Vetor - Conceito

Em algumas situações precisaremos armazenar na memória do computador várias informações com o mesmo significado, como preços de vários produtos e, portanto, necessitaremos de muitas variáveis para armazenar estes preços. Nestes casos a utilização de variáveis convencionais tornará o programa muito extenso e de difícil compreensão.

Ex:

```
float PrecoProduto1,PrecoProduto2,PrecoProduto3,  
PrecoProduto4,PrecoProduto5,..., PrecoProduto20;
```

Vetor - Conceito

Para resolver este problema, a informática utilizou os conceitos de matriz da álgebra linear para armazenar muitas informações em uma única estrutura de dados e não mais em várias variáveis.

Esta estrutura se chama **VETOR (ARRAY)**

Vetor - Conceito

Antes de definirmos o que é um vetor vamos ilustrar uma situação na qual a utilização de variáveis convencionais não traz bons resultados no programa.

Vetor - Conceito

Imagine que você precisa fazer um programa para ler 20 notas de provas, calcular a média das notas e escrever as notas que ficaram acima da média.

Neste programa precisaremos ler todas as notas para calcularmos a média e depois precisaremos comparar cada nota com a média, portanto cada nota lida deve ficar armazenada em uma variável. Considerando que são 20 notas, temos 20 variáveis e todos os comandos que utilizarem estas variáveis serão repetidos 20 vezes.

Vetor - Conceito

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    float Nota1,Nota2,Nota3,Nota4,Nota5,Nota6,Nota7,Nota8,Nota9,Nota10;
    float Nota11,Nota12,Nota13,Nota14,Nota15,Nota16,Nota17,Nota18,Nota19,Nota20, Media;
    printf("1a Nota: "); scanf("%f",&Nota1);
    printf("2a Nota: "); scanf("%f",&Nota2);
    printf("3a Nota: "); scanf("%f",&Nota3);
    printf("4a Nota: "); scanf("%f",&Nota4);
    printf("5a Nota: "); scanf("%f",&Nota5);
    printf("6a Nota: "); scanf("%f",&Nota6);
    printf("7a Nota: "); scanf("%f",&Nota7);
    printf("8a Nota: "); scanf("%f",&Nota8);
    printf("9a Nota: "); scanf("%f",&Nota9);
    printf("10a Nota: "); scanf("%f",&Nota10);
    printf("11a Nota: "); scanf("%f",&Nota11);
    printf("12a Nota: "); scanf("%f",&Nota12);
    printf("13a Nota: "); scanf("%f",&Nota13);
    printf("14a Nota: "); scanf("%f",&Nota14);
    printf("15a Nota: "); scanf("%f",&Nota15);
    printf("16a Nota: "); scanf("%f",&Nota16);
    printf("17a Nota: "); scanf("%f",&Nota17);
    printf("18a Nota: "); scanf("%f",&Nota18);
    printf("19a Nota: "); scanf("%f",&Nota19);
    printf("20a Nota: "); scanf("%f",&Nota20);
```

Vetor - Conceito

```
Media = (Nota1+Nota2+Nota3+Nota4+Nota5+Nota6+Nota7+Nota8+Nota9+Nota10
        +Nota11+Nota12+Nota13+Nota14+Nota15+Nota16+Nota17+Nota18+Nota19+Nota20)/20;
printf("Media da Turma: %.2f\n",Media);
printf("Notas acima da media:\n");
if(Nota1>Media)
    printf("Nota 1 : %.1f\n",Nota1);
if(Nota2>Media)
    printf("Nota 2 : %.1f\n",Nota2);
if(Nota3>Media)
    printf("Nota 3 : %.1f\n",Nota3);
if(Nota4>Media)
    printf("Nota 4 : %.1f\n",Nota4);
if(Nota5>Media)
    printf("Nota 5 : %.1f\n",Nota5);
if(Nota6>Media)
    printf("Nota 6 : %.1f\n",Nota6);
if(Nota7>Media)
    printf("Nota 7 : %.1f\n",Nota7);
if(Nota8>Media)
    printf("Nota 8 : %.1f\n",Nota8);
if(Nota9>Media)
    printf("Nota 9 : %.1f\n",Nota9);
if(Nota10>Media)
    printf("Nota 10 : %.1f\n",Nota10);
```

Vetor - Conceito

```
;
if(Nota11>Media)
    printf("Nota 11 : %.1f\n",Nota11);
if(Nota12>Media)
    printf("Nota 12 : %.1f\n",Nota12);
if(Nota13>Media)
    printf("Nota 13 : %.1f\n",Nota13);
if(Nota14>Media)
    printf("Nota 14 : %.1f\n",Nota14);
if(Nota15>Media)
    printf("Nota 15 : %.1f\n",Nota15);
if(Nota16>Media)
    printf("Nota 16 : %.1f\n",Nota16);
if(Nota17>Media)
    printf("Nota 17 : %.1f\n",Nota17);
if(Nota18>Media)
    printf("Nota 18 : %.1f\n",Nota18);
if(Nota19>Media)
    printf("Nota 19 : %.1f\n",Nota19);
if(Nota20>Media)
    printf("Nota 20 : %.1f\n",Nota20);
system("pause");
return 0;
}
```

Vetor - Conceito

No programa acima percebemos que os comandos que foram utilizados para ler uma nota `printf("1a Nota: "); scanf("%f",&Nota1);` foram repetidos 20 vezes.

O comando que testa se a nota está acima da média também foi repetido 20 vezes.

```
if(Nota1>Media)
    printf("Nota 1 : %.1f\n",Nota1);
```

Este fato aumenta muito o tamanho, dificultando a legibilidade e as futuras manutenções do programa.

Para resolver este problema foi criada uma estrutura de dados que é capaz de armazenar um conjunto de dados do mesmo tipo e que tem o mesmo significado, este tipo de dados é chamado em programação de **Vetor (ou Array)**.

Vetor - Conceito

Um vetor é um agrupamento de componentes de mesmo tipo dispostos sequencialmente(primeiro,segundo, terceiro, quarto).

A referência a um elemento do vetor é feita pelo nome do vetor mais o índice do elemento.

Resumindo:

um vetor é como se fosse uma variável capaz de armazenar mais de uma informação do mesmo tipo, por exemplo, várias notas sendo armazenada em uma variável.

Como temos várias notas em uma variável, precisamos de um índice para podermos dizer qual das notas estamos utilizando em um dado momento dentro do programa.

notas



Vetor - Declaração

```
int Numeros[n];
```

Onde :

int é o tipo dos elementos do vetor, ou seja, o vetor será composto de números inteiros.

Numeros é o nome do vetor.

n é o número de elementos do vetor.

Utilizando a sintaxe acima vamos declarar dois vetores:

Um vetor para armazenar 30 idades de pessoas.

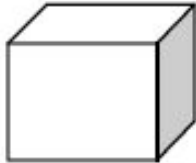
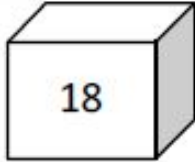
```
int Idades[30];
```

Um vetor para armazenar 55 preços de produtos.

```
float Precos[55];
```

Vetor - Representação

Fazendo uma representação gráfica de uma variável, apenas para efeitos didáticos, poderíamos representar uma variável comum, do tipo int, por uma caixa onde podemos colocar um número inteiro de cada vez, conforme abaixo.

Comando em C++	Representação Gráfica do Comando
<code>int IDADE;</code>	IDADE 
<code>IDADE = 18;</code>	IDADE 

Vetor - Leitura/ Impressão

Quando estudamos a variáveis mostramos que existem cinco operações básicas que fazemos com uma variável:

Definir uma variável.

int Idade;

Atribuir valor a uma variável.

Idade = 8;

Ler uma variável.

scanf("%d", &Idade);

Escrever uma variável.

printf("Idade : %d\n", Idade);

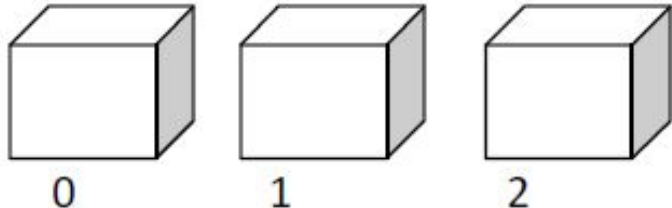
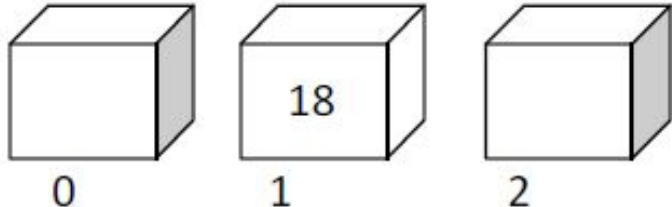
Utilizar o valor da variável em uma expressão aritmética:

Meses = Idade * 12;

Vetor - Representação

Utilizando a mesma representação para Vetor, um vetor é uma variável que representa várias caixas, do mesmo tipo.

Para obtermos o valor de uma das caixas utilizamos o nome do vetor e o índice da caixa.

Comando em C++	Representação Gráfica do Comando
<code>int IDADES[3];</code>	IDADES 
<code>IDADES[1] = 18;</code>	IDADES 

IMPORTANTE: O índice do vetor, no C/C++, começa do 0, portanto se definirmos um vetor Idades de 3 posições seus índices serão

Idades[0]

Idades[1]

Idades[2]

Vetor - Leitura/ Impressão

Estas mesmas operações, feitas com a estrutura de vetor, ficariam assim:

Definir uma variável.

```
int Idades[10];
```

Atribuir valor a uma variável.

```
Idades[4]=8;
```

Ler uma variável.

```
scanf("%i",&Idades[2]);
```

Escrever uma variável.

```
printf("Idade 5 : %d\n",Idades[4]);
```

Vetor - Atividade Prática

Sendo o vetor Idades, de 10 elementos, igual a :

Conteúdo	2	6	8	3	10	9	15	21	33	44
Índice	0	1	2	3	4	5	6	7	8	9

E as variáveis $X = 1$ e $Y = 3$, escreva o valor correspondente conteúdo do elemento do vetor:

A) Idades[X+1]

B) Idades[X+Y]

C) Idades[X*4]

D) Idades[Idades[X]]

E) Idades[Idades[X]+3]

E) Idades[Idades[Y]*10]

Vetor - Utilização

Para reduzir o tamanho do programa utilizamos uma estrutura de repetição para trabalhar com vetores, normalmente o **for**.

Assim podemos fazer operações com cada elemento do vetor utilizando o comando de repetição **for**, estudado em Algoritmos e Programação I, para gerar o índice do vetor e colocar dentro do bloco de comando do comando **for** todos os comandos necessários para realizar operações que se repetiam no programa anterior, evitando que se escreva os comandos novamente, porque o próprio comando **for** repetirá o bloco de comandos.

Vetor - Utilização- Exemplo 1

Sem Vetor

```
#include<stdio.h>
int main()
{
    int Numero1,Numero2,Numero3,Numero4,Numero5;
    printf("Informe o numero 1: ");
    scanf("%i",&Numero1);
    printf("Informe o numero 2: ");
    scanf("%i",&Numero2);
    printf("Informe o numero 3: ");
    scanf("%i",&Numero3);
    printf("Informe o numero 4: ");
    scanf("%i",&Numero4);
    printf("Informe o numero 5: ");
    scanf("%i",&Numero5);
    system("pause");
    return 0;
}
```

Com Vetor

```
#include<stdio.h>

int main()
{
    int Numero[5],l;
    for(l=0;l<5;l++)
    {
        printf("Informe o numero %i: ",l);
        scanf("%i",&Numero[l]);
    }
    return 0;
}
```

Vetor - Utilização- Exemplo 2

// programa que atribui o valor 1 a todos os elementos de um vetor de inteiros e imprime o vetor.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int X;
    int Numeros[10];
    for(X=0;X<10;X++)
    {
        Numeros[X]=1;
    }
    system("cls");
    printf("\nVetor Numeros:\n");
    for(X=0;X<10;X++)
    {
        printf("Numeros[%i]: %d\n",X,Numeros[X]);
    }
    system("pause");
    return 0;
}
```

Vetor - Utilização- Exemplo 3

// programa que atribui os valores pares, a partir do número 2, aos elementos do um vetor de inteiros e imprime o vetor.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int X;
    int Numeros[10], NumeroPar;
    NumeroPar = 2;
    for(X=0;X<10;X++)
    {
        Numeros[X]=NumeroPar;
        NumeroPar = NumeroPar + 2;
    }
    system("cls");
    printf("\nVetor Numeros:\n");
    for(X=0;X<10;X++)
    {
        printf("Numeros[%i]: %d\n",X,Numeros[X]);
    }
    system("pause");
    return 0;
}
```


Vetor - Utilização- Exemplo 4

// programa que atribui o valor 0 para os 5 primeiros elementos do vetor e o valor 1 para os demais elementos de um vetor de inteiros e imprime o vetor.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int X;
    int Numeros[10], NumeroPar;
    for(X=0;X<10;X++)
    {
        if(X<=4)
            Numeros[X]=0;
        else
            Numeros[X]=1;
    }
    system("cls");
    printf("\nVetor Numeros:\n");
    for(X=0;X<10;X++)
    {
        printf("Numeros[%i]: %d\n",X,Numeros[X]);
    }
    system("pause");
    return 0;
}
```

Vetor - Utilização- Exemplo 5

// programa que lê vetor de inteiros e imprime na ordem lida e na ordem inversa.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int X;
    int Numeros[10];
    for(X=0;X<10;X++)
    {
        printf("%iº numero: ",X+1);
        scanf("%i",&Numeros[X]);
    }
    system("cls");
    printf("\nVetor impresso na ordem de leitura:\n");
    for(X=0;X<10;X++)
    {
        printf("Numeros[%i]: %d\n",X,Numeros[X]);
    }
    printf("\nVetor impresso ao inverso da ordem de leitura:\n");
    for(X=9;X>=0;X--)
    {
        printf("Numeros[%i]: %d\n",X,Numeros[X]);
    }
    return 0;
}
```

Vetor - Utilização- Exemplo 6

```
// programa que lê vetor de Notas, calcula média e mostra notas acima da média
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int Indice;
    float Notas[20];
    float Media, Total;
    Total = 0;
    for(Indice=0;Indice<20;Indice++)
    {
        printf("%dª nota: ",Indice+1);
        scanf("%f",&Notas[Indice]);
        Total = Total + Notas[Indice];
    }
    Media = Total/20.0;
    printf("Media da Turma: %.2f\n", Media);
    printf("Notas acima da media: \n");
    for(Indice=0;Indice<20;Indice++)
    {
        if(Notas[Indice] > Media)
            printf("Nota %d: %.2f\n",Indice+1,Notas[Indice]);
    }
    return 0;
}
```

Vetor - Utilização- Exemplo 7

```
// programa que lê vetor de inteiros, localiza e imprime o maior valor do vetor.
int main()
{
    int Array[8];
    int l, MaiorValor;
    for(l=0;l<8;l++)
    {
        printf("%iº numero: ",l+1);
        scanf("%i",&Array[l]);
    }
    MaiorValor = Array[0]; // inicializa MaiorValor com 1º número do vetor
    for(l=1;l<8;l++) //laço for pode iniciar no índice 1 pois o índice 0 já foi usado
    {
        if(Array[l] > MaiorValor)
        {
            MaiorValor = Array[l];
        }
    }
    printf("\nMaior Valor do Array: %d\n",MaiorValor);
    return 0;
}
```

Atividade prática

Copie os códigos fonte dos exemplos apresentados nos slides, cole no DEV-C, compile e execute para verificar o resultado de cada programa.