

# INTRODUÇÃO À PROGRAMAÇÃO

Prof.<sup>a</sup> Priscilla Abreu

[priscilla.braz@rj.senac.br](mailto:priscilla.braz@rj.senac.br)



# Introdução à Programação



## Roteiro de Aula

- Objetivo da aula
- Vetores unidimensionais

# Introdução à Programação



## Objetivo da aula

Revisar o funcionamento e manipulação de vetores.

# Introdução à Programação



## INTRODUÇÃO

- Exemplos:
  - Tipos primitivos:
    - `int idade;`
    - `float altura;`
  - Tipos compostos:
    - `float notas[50];`
    - `aluno alunos[50];`



ALUNO
Nome
Matricula
Nota1
Nota2
Endereço

# Introdução à Programação



## VETOR

- Declaração de um vetor em C:

```
int nome_vetor[tamanho];
```

Quantidade de posições de memória para essa variável

Tipo dos dados do conjunto

Identificador ou nome do conjunto

Dimensão, ou tamanho máximo do conjunto

# Introdução à Programação



## VOLTANDO À SITUAÇÃO...

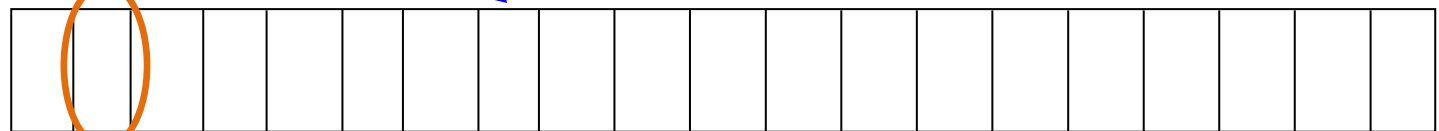
Solução para o problema das múltiplas variáveis de mesmo tipo: um vetor.

Nome: comum a todas as variáveis

Notas

(...)

Nota na  
posição 1:  
estaremos  
acessando a  
2ª nota.



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Através da posição acessamos cada elemento do vetor.

# Introdução à Programação



## ACESSANDO UM ELEMENTO...

Coloca-se o nome da variável e entre [] coloca-se o índice, que indica a posição do elemento.

O índice é uma constante inteira, uma variável inteira ou um cálculo que resulte em valor inteiro.

Exemplos:

- `notas[1] = 10;`
- `vetor[5] = 30.4;`
- `i = 0;`
- `notas[i] = 7.5;`

# Introdução à Programação



## ACESSANDO VÁRIOS ELEMENTOS...

9.5	10	8	9.4	3.5	2.9	7	8	6.8	10
notas[0]	notas[1]	notas[2]	notas[3]	notas[4]	notas[5]	notas[6]	notas[7]	notas[8]	notas[9]

```
for (i=0; i<=9; i++){  
    printf("Informe a nota %d:", i+1 );  
    scanf("%f", &notas[i]);  
}
```



# Introdução à Programação



## EXEMPLO

```
#include <stdio.h>
#define max 20
int main(){
    int valores[max];
    int i;
    for (i=0;i<max;i++){
        printf("Digite um número:");
        scanf("%d",&valores[i]);
    }
    for (i=0;i<max;i++){
        printf("Valor[%d]: %d \n",i,valores[i]);
    }
}
```

# Introdução à Programação



## EXERCÍCIO 1

Faça um programa que leia um vetor de 20 inteiros, conte quantos números pares estão contidos no vetor e imprima essa quantidade bem como todo o vetor de números.

# Introdução à Programação



## EXERCÍCIO 1 – SOLUÇÃO

```
#include <stdio.h>
int main(){
    int vet[20];
    int i, pares=0;
    for (i=0;i<20;i++){
        printf("Valor:");
        scanf("%d",&vet[i]);
        if (vet[i]%2==0)
            pares++;
    }
```

# Introdução à Programação



## EXERCÍCIO 1 – SOLUÇÃO

```
printf("\nQuantidade de números pares:
%d",pares);
for (i=0;i<20;i++){
    printf("Valor %d: %d\n", i+1, vet[i]);
}
}
```

# Introdução à Programação



## EXERCÍCIO EM GRUPO

Escreva um programa que armazene 10 números inteiros em um vetor e exiba o maior valor armazenado.

# Introdução à Programação



## PRATICANDO...

Considere o seguinte vetor:

**VET**

12	18	3	41	15
----	----	---	----	----

**COMO VOCÊ ENCONTRA O MAIOR VALOR???**

# Introdução à Programação



## PRATICANDO...

Considere o seguinte vetor:

**VET**

12	18	3	41	15
----	----	---	----	----

E uma variável que terá por objetivo armazenar o maior valor, até o momento, do vetor.

**MAIOR**

# Introdução à Programação



## PRATICANDO...

Considerando que o vetor será percorrido, podemos supor que – inicialmente – o maior valor é o primeiro elemento do vetor.

**MAIOR**

12
----

Agora, vamos comparar cada elemento do vetor com essa variável e atualizá-la a cada vez que encontrarmos um valor maior.

**VET**

12	18	3	41	15
----	----	---	----	----



# Introdução à Programação



## PRATICANDO...

Vamos comparar cada elemento do vetor com essa variável e atualizá-la a cada vez que encontrarmos um valor maior.

**MAIOR**

12
----

**cont = 1**

**VET**

12	18	3	41	15
----	----	---	----	----



**vet[i] > maior ???**

# Introdução à Programação



## PRATICANDO...

Vamos comparar cada elemento do vetor com essa variável e atualizá-la a cada vez que encontrarmos um valor maior.

**MAIOR**



**cont = 1**

**VET**

12	18	3	41	15
----	----	---	----	----



**vet[i] > maior ???**

# Introdução à Programação



## PRATICANDO...

Vamos comparar cada elemento do vetor com essa variável e atualizá-la a cada vez que encontrarmos um valor maior.

**MAIOR**

18
----

**cont = 2**

**VET**

12	18	3	41	15
----	----	---	----	----



**vet[i] > maior ???**

# Introdução à Programação



## PRATICANDO...

Vamos comparar cada elemento do vetor com essa variável e atualizá-la a cada vez que encontrarmos um valor maior.

**MAIOR**

18
----

**cont = 3**

**VET**

12	18	3	41	15
----	----	---	----	----



**vet[i] > maior ???**

# Introdução à Programação



## PRATICANDO...

Vamos comparar cada elemento do vetor com essa variável e atualizá-la a cada vez que encontrarmos um valor maior.

**MAIOR**



**cont = 3**

**VET**

12	18	3	41	15
----	----	---	----	----



**vet[i] > maior ???**

# Introdução à Programação



## PRATICANDO...

Vamos comparar cada elemento do vetor com essa variável e atualizá-la a cada vez que encontrarmos um valor maior.

**MAIOR**

41

**cont = 4**

**VET**

12

18

3

41

15



**vet[i] > maior ???**

# Introdução à Programação



## PRATICANDO...

Vamos comparar cada elemento do vetor com essa variável e atualizá-la a cada vez que encontrarmos um valor maior.

**MAIOR**

41
----

**VET**

12	18	3	41	15
----	----	---	----	----

# Introdução à Programação



## SOLUÇÃO – EXERCÍCIO 2

```
#include <stdio.h>
#define max 10
int main(){
    int vet[max];
    int cont, maior;
    for(cont=0;cont<max;cont++){
        printf("Informe o valor para o vetor na
posição %d: ", cont+1);
        scanf("%d",&vet[cont]);
    }
    maior = vet[0];
```



# Introdução à Programação



## SOLUÇÃO – EXERCÍCIO 2

```
for(cont=0;cont<max;cont++){  
    if(vet[cont]>maior){  
        maior = vet[cont];  
    }  
}  
printf("\nVETOR \n");  
for(cont=0; cont<max;cont++){  
    printf("%d\n",vet[cont]);  
}  
printf("\nMaior valor: %d.",maior);  
}
```

# Introdução à Programação



## EXERCÍCIO 3

Considerando o exercício anterior, o que precisaria ser adaptado para que a busca pelo maior valor seja feita por uma função?

A função deve receber o conjunto de dados por parâmetro e retornar o maior valor.

# Introdução à Programação



## EXERCÍCIO 3 – solução

```
#include <stdio.h>
#include <locale.h>
#define max 10
int maior(int vet[]){
    int cont,ma;
    ma=vet[0];
    for(cont=0;cont<max;cont++){
        if(vet[cont]>ma)
            ma = vet[cont];
    }
    return ma;
}
```

# Introdução à Programação



## EXERCÍCIO 3 – solução

```
int main(){
    setlocale(LC_ALL,"Portuguese");
    int vet[max], cont;
    for(cont=0;cont<max;cont++){
        printf("%dº valor: ",cont+1);
        scanf("%d",&vet[cont]);
    }
    printf("\nMaior      valor      informado:
%d\n",maior(vet));
}
```

# Introdução à Programação



## ATIVIDADE – ENTREGA até 20/11

Faça um programa que lê 10 números inteiros e os armazena em um vetor, e calcula e mostra dois vetores resultantes:

- o primeiro vetor resultante deve conter os números positivos;
- o segundo deve conter os números negativos.

Cada vetor resultante vai ter, no máximo, 10 posições, que podem não ser completamente utilizadas, sendo que as posições vazias devem ficar ao final do vetor.