



MCTA028-15: Programação Estruturada

Aula 4: Vetores (Primeira Parte)

Wagner Tanaka Botelho

wagner.tanaka@ufabc.edu.br / wagtanaka@gmail.com

Universidade Federal do ABC (UFABC)

Centro de Matemática, Computação e Cognição (CMCC)

Introdução

➤ Por que utilizar?

- Até agora os **algoritmos** implementados armazenavam um **único** valor por vez;
- Sempre que um **novo valor** é **atribuído** a uma variável, o **valor anterior** é perdido, como por exemplo:

```
Ex_01.c X
1  #include<stdio.h>
2
3  void main(){
4      int x = 10;
5      printf("x = %i\n", x);
6
7      x = 30;
8      printf("x = %i\n", x);
9  }
```

	MEMÓRIA
x	10 30



```
Ex_02.c X Ex_01.c X
1  #include<stdio.h>
2
3  void main(){
4      float n1=0, n2=0, n3=0, n4=0, n5=0, media=0;
5
6      printf("Digite as notas de 5 alunos:\n");
7
8      scanf("%f", &n1);
9      scanf("%f", &n2);
10     scanf("%f", &n3);
11     scanf("%f", &n4);
12     scanf("%f", &n5);
13
14     media = (n1+n2+n3+n4+n5)/5;
15     printf("A media eh: %.2f.\n", media);
16
17     if(n1 > media){
18         printf("A nota %.2f eh maior do que a media.\n", n1);
19     }
20
21     if(n2 > media){
22         printf("A nota %.2f eh maior do que a media.\n", n2);
23     }
24
25     if(n3 > media){
26         printf("A nota %.2f eh maior do que a media.\n", n3);
27     }
28
29     if(n4 > media){
30         printf("A nota %.2f eh maior do que a media.\n", n4);
31     }
32
33     if(n5 > media){
34         printf("A nota %.2f eh maior do que a media.\n", n5);
35     }
36 }
```

Agora, vamos implementar um algoritmo que leia as notas de uma turma com **CINCO** estudantes e depois imprima as notas **MAIORES** do que a **MÉDIA** da **TURMA**.

Imagina se tivermos que implementar um algoritmo para uma turma com 100 alunos?

100 variáveis

Agora, vamos implementar um algoritmo que leia as notas de uma turma com **CEM** estudantes e depois imprima as notas **MAIORES** do que a **MÉDIA** da **TURMA**.

100 scanf()

n1...n100

100 ifs

Você considera viável
implementar desta forma?

```
#include<stdio.h>

void main(){
    float n1=0,n2=0,n3=0,n4=0,n5=0,n6=0,n7=0,...,n100=0,media=0;
    printf("Digite as notas de 5 alunos:\n");
    scanf("%f", &n1);
    scanf("%f", &n2);
    scanf("%f", &n3);
    scanf("%f", &n4);
    scanf("%f", &n5);
    scanf("%f", &n6);
    scanf("%f", &n7);
    ....
    ....
    ....
    scanf("%f", &n100);
    media = (n1+n2+n3+n4+n5+n6+n7+.....+n100)/100;
    printf("A media eh: %.2f.\n", media);

    if(n1 > media){
        printf("A nota %.2f eh maior do que a media.\n", n1);
    }
    if(n2 > media){
        printf("A nota %.2f eh maior do que a media.\n", n2);
    }
    if(n3 > media){
        printf("A nota %.2f eh maior do que a media.\n", n3);
    }
    if(n4 > media){
        printf("A nota %.2f eh maior do que a media.\n", n4);
    }
    if(n5 > media){
        printf("A nota %.2f eh maior do que a media.\n", n5);
    }
    if(n6 > media){
        printf("A nota %.2f eh maior do que a media.\n", n5);
    }
    if(n7 > media){
        printf("A nota %.2f eh maior do que a media.\n", n5);
    }
    ....
    ....
    ....
    if(n100 > media){
        printf("A nota %.2f eh maior do que a media.\n", n5);
    }
}
```

Vetores

- As **variáveis** podem ser **compostas** e formadas por **uma** ou **mais** posições, podendo armazenar **um** ou **mais** dados:

Índices

→

0

1

2

3

4

5

6

7

8

9

do

→

vet

5

10

1

2

34

3

15

5

31

-1

- As variáveis devem ser do **MESMO** tipo;
- Um **vetor** é definido como tendo um **número fixo de células**, com a mesma configuração;
- Cada **célula** armazena **um**, e **somente um**, dos **valores** de **dados** do **vetor**;
- Cada **célula** de um **vetor** possui sua própria **posição**, ou **ÍNDICE**, através do qual pode ser referenciada.

Declaração

Vetores: Declaração

➤ Na Linguagem C, a declaração de um *array/vetor* é:

```
tipo_dado nome_array[tamanho];
```

➤ Para declarar um *vetor* do tipo `float` com as *notas* de *100 alunos* do exemplo apresentado nos primeiros slides:

```
float notas[100];
```

Armazenando Um Elemento

Vetores: Armazenando Um Elemento

- Para **armazenar** um elemento em uma posição do vetor, é necessário utilizar um **índice** que indique a **posição** deste elemento;
- Por exemplo, para **armazenar** um **valor** na **SEGUNDA POSIÇÃO** do vetor `notas`, tem-se o seguinte comando:

`notas[1] = 8.5;`

0	1	2	99
5.0	8.5	2.5					9.0
- Os **índices** começam em **ZERO** e vão até o **número** de **posições reservadas**, menos um;
- No vetor `notas`, os índices válidos vão de **ZERO** à **NOVENTA E NOVE**.

Vetores: Armazenando Um Elemento Pelo Teclado

```
float notas[100];
```

Nome do vetor e o ÍNDICE!!

```
scanf("%f", &notas[0]);
```

Tipo de dado
do vetor notas[]

Vetores: Armazenando Elementos Pelo Teclado

- Um **vetor/array** possui um **índice**:
- Portanto, pode-se usar **ESTRUTURAS de REPETIÇÃO** (**for/while**) para executar **repetidamente** um mesmo comando para **CADA** posição do vetor.

```
float notas[100];  
  
for(int i=0; i<100;i++){  
    scanf("%f", &notas[i]);  
}
```

Acessando Um Elemento

Vetores: Acessando Um Elemento

- O acesso ao **valor** de cada elemento de um vetor deve ser realizado utilizando o **ÍNDICE**:

	0	1	2	3	4	99
notas	5.0	8.5	2.5	6.7	3.4															9.0

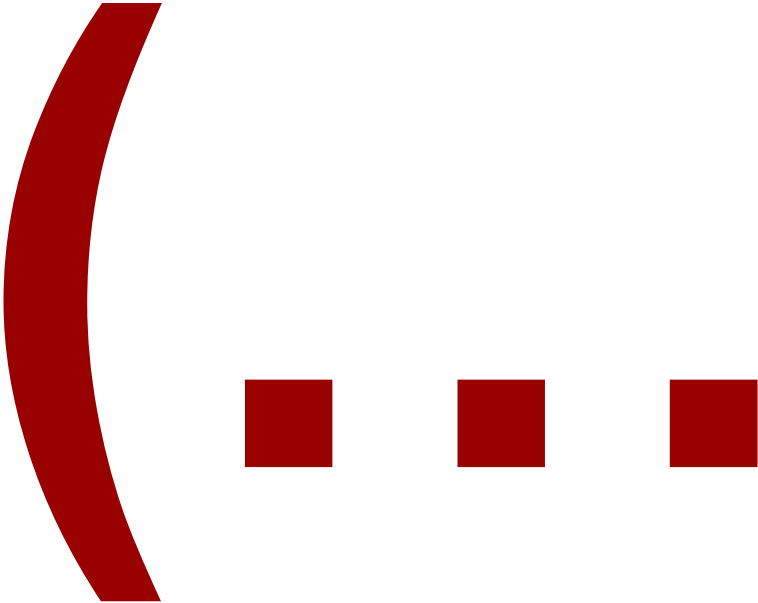
- Na **posição [0]** “índice **0**” está armazenada a nota 5.0;
- Na **posição [2]** “índice **2**” está armazenada a nota 2.5;

Vetores: Acessando os Elementos

- O acesso aos elementos de um **vetor/array** é realizado pelo **índice**:
- Por meio das **ESTRUTURAS de REPETIÇÃO** (**for/while**):

```
float notas[100];
```

```
for(int i=0; i<100;i++){  
    printf("%f", notas[i]);  
}
```



Declarar

```
int a;
```

```
int a[8];
```

Armazenar Valores

```
a=10;
```

```
a[0]=10;
```

```
a[1]=3;
```

```
a[2]=4;
```

```
....
```

Armazenar Valores pelo Teclado

```
scanf("%i", &a);
```

```
scanf("%i", &a[0]);
```

```
scanf("%i", &a[1]);
```

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

```
....
```

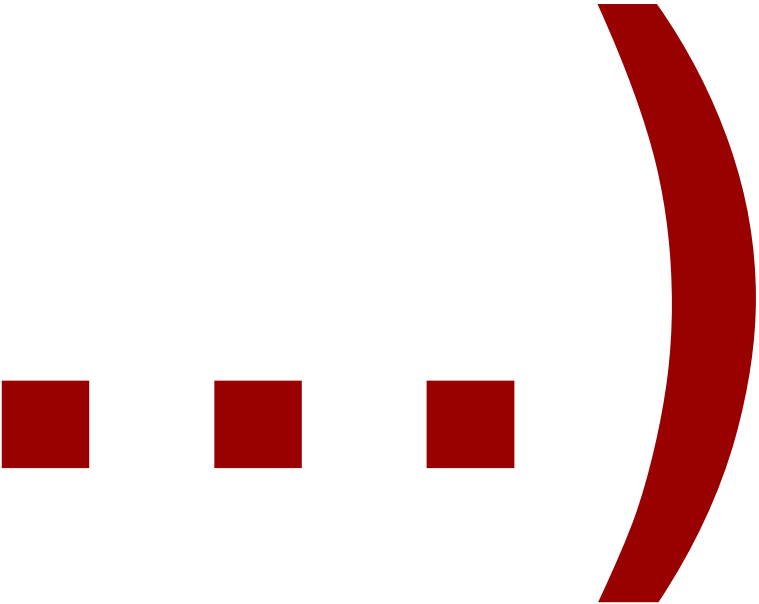
Operações

```
a = a+1;
```

```
a = a-1;
```

```
a[1] = a[1]+1;
```

```
a[2] = a[2]-1;
```

Exemplo

Implementar um algoritmo que leia as notas de n alunos, calcule a MÉDIA e depois IMPRIMA as notas MAIORES do que a MÉDIA da TURMA. O n será definido pelo usuário.

```

1 #include<stdio.h>
2
3 void main(){
4     float media=0, soma=0;
5     int qtd = 0;
6
7     printf("Digite a quantidade de alunos:\n");
8     scanf("%i", &qtd);
9
10    float notas[qtd];
11
12    for(int i = 0;i < qtd ;i++){
13        scanf("%f", &notas[i]);
14
15        soma = soma + notas[i];
16    }
17
18    media = soma/qtd;
19    printf("Media da turma eh: %.2f \n", media);
20
21    for(int i = 0;i < qtd ;i++){
22        if(notas[i] > media){
23            printf("A nota %.2f eh maior do que a media
24        }
25    }
26 }

```

linha	media	soma	qtd	i	notas[]
4	0	0			
5			0		
8			3		
10					⁰ ¹ ² [, ,]
12				0	
13					⁰ ¹ ² [5.0, ,]
15		5.0			
12				1	
13					⁰ ¹ ² [5.0, 8.5,]
15		13.5			
12				2	
13					⁰ ¹ ² [5.0, 8.5, 6.0]
15		19.5			
12				3	
18	6.5				
19	{6.5}				

```

1 #include<stdio.h>
2
3 void main(){
4     float media=0, soma=0;
5     int qtd = 0;
6
7     printf("Digite a quantidade de alunos:\n");
8     scanf("%i", &qtd);
9
10    float notas[qtd];
11
12    for(int i = 0;i < qtd ;i++){
13        scanf("%f", &notas[i]);
14
15        soma = soma + notas[i];
16    }
17
18    media = soma/qtd;
19    printf("Media da turma eh: %.2f \n", media);
20
21    for(int i = 0;i < qtd ;i++){
22        if(notas[i] > media){
23            printf("A nota %.2f eh maior do que a media.\n", notas[i]);
24        }
25    }
26 }

```

linha	media	soma	qtd	i	notas[]
	{6.5}	19.5	3	3	⁰ [5.0, ¹ 8.5, ² 6.0]
21				0	
21				1	
23					[1] = {8.5}
21				2	
21				3	

Referências

- Slides do Prof. Luiz Rozante;
- SALES, André Barros de; AMVAME-NZE, Georges. Linguagem C: roteiro de experimentos para aulas práticas. 2016;
- BACKES, André. Linguagem C Completa e Descomplicada. Editora Campus. 2013;
- SCHILDT, Herbert. C Completo e Total. Makron Books. 1996;
- DAMAS, Luís. Linguagem C. LTC Editora. 1999;
- DEITEL, Paul e DEITEL, Harvey. C Como Programar. Pearson. 2011.