



# Estruturas de dados homogêneas

---

“Três classes de pessoas são infelizes:  
as que não sabem e não perguntam,  
as que sabem e não ensinam,  
as que ensinam e não fazem.”

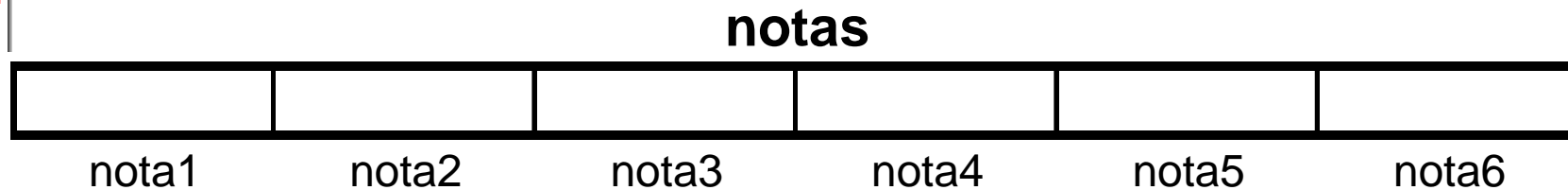
V.Beda

# Sumário

- Vectores em C



# Vectores



- Conjunto de valores todos do mesmo tipo aos quais se dá um nome.
- Cada dado individual é chamado de elemento
- Pode conter elementos de qualquer tipo
  - int , char, float, double, . . . , outros
- Os elementos ocupam posições contíguas de memória.
- O nome do vector representa o conjunto de elementos.



# Vectores

- Declaração
  - tipo nome\_variável [numero de elementos];
    - char nota[7];
- Acesso
  - Indexado pelo operador [ ]
    - nota[2]
  - O compilador não verifica os índices
    - nota[-1] = 5 ;
    - nota[7] = 15;

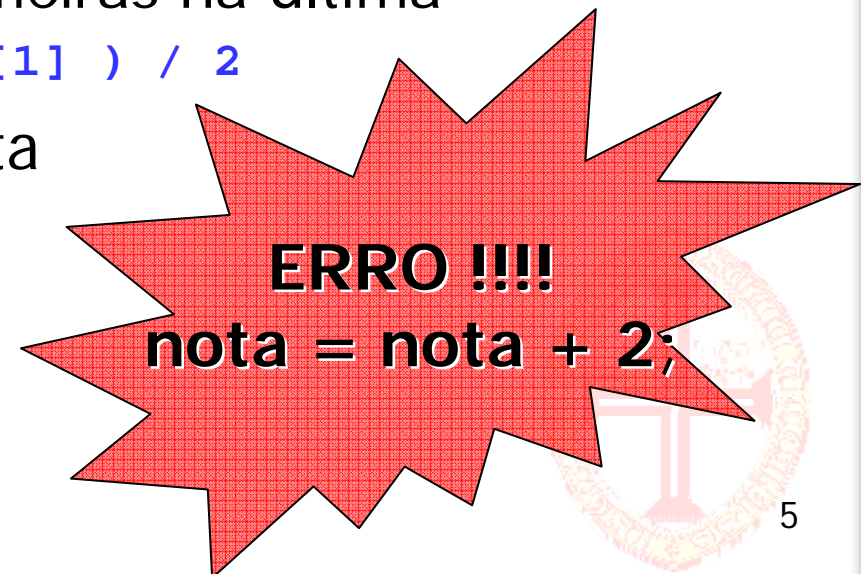
	0	1	2	3	4	5	6	7	8
<b>c</b>		<b>idade</b>			<b>sexo</b>				
	c:0	c:1	24	c:3	c:4	m'	c:5	c:6	c:7
<b>d</b>	<b>Peso</b>				<b>altura</b>				
	d:0	74,32		d:4	1,75			d:8	
<b>e</b>	<b>nota[0]</b>	<b>nota[1]</b>	<b>nota[2]</b>	<b>nota[3]</b>	<b>nota[4]</b>	<b>nota[5]</b>	<b>nota[6]</b>		
	e:0	e:1	e:2	e:3	e:4	e:5	3	e:6	e:8
	<b>nota</b>								

**15**

# Utilização de vectores

- Definir um vector chamado nota com dez elementos do tipo inteiro
  - `int nota[10];`
- Colocar a nota 10 na primeira nota
  - `nota[0] = 10;`
- Colocar a nota 5 na segunda nota
  - `nota[1] = 5;`
- Colocar a média das duas primeiras na última
  - `nota[9] = (nota[0] + nota [1] ) / 2`
- Somar dois valores a cada nota

```
for(i=0; i< 10; i++)  
{  
    nota[i] +=2;  
}
```



# Utilização de vectores

- Ler as notas

```
for( i=0; i< 10; i++)  
{  
    scanf("%d", &nota[i] ;  
}
```

- Calcular a média

```
int soma=0;  
for( i=0; i< 10; i++)  
{  
    Soma += nota[i] ;  
}  
double media = (double) soma / 10.0;
```



# Utilização de vectores

- Nota máxima

```
int notaMaxima=nota[0];
for(i=1; i< 10; i++)
{
    if(notaMaxima < nota[i])
    {
        notaMaxima = nota[i];
    }
}
```



## Exercício

- Para avaliar a regularidade de vendas de um trabalhador, escreva um programa que realize a leitura as comissões pagas a um trabalhador durante um ano, calcule a média de comissões pagas e o seu desvio padrão.

$$dp = \sqrt{\frac{\sum (x[i] - \text{media})^2}{N}}$$





# Resolução

## Declaração de variáveis

```
double comissao[12]; //comissões do trabalhador
int i; // variavel auxiliar
```

## Introdução dos valores

```
printf("Introduza as Comissões n");
for(i=0; i< 12; i++){
    printf("mes %d :", i+1);
    scanf("%f",&comissao[i]);
}
```

0 1 2 3 4 5 6 7 8 9 10 11

--	--	--	--	--	--	--	--	--	--	--	--

## ecra

### Introduza as comissoes

mes 1: 10

mes 2: 15

mes 3: 5

mes 4: 8

mes 5: 10

mes 6: 12

mes 7: 16

mes 8: 8

mes 9: 5

mes 10: 12

mes 11: 11

mes 12: 10

# Resolução

## Cálculo da média

```
double soma = 0.0;
for( int i=0 ; i < 12 ; i++)
    soma += comissao[i];
double media = soma / 12.0;
printf( " Media : %lf",media);
```

ecra

Media : 10.1667

## Desvio Padrão

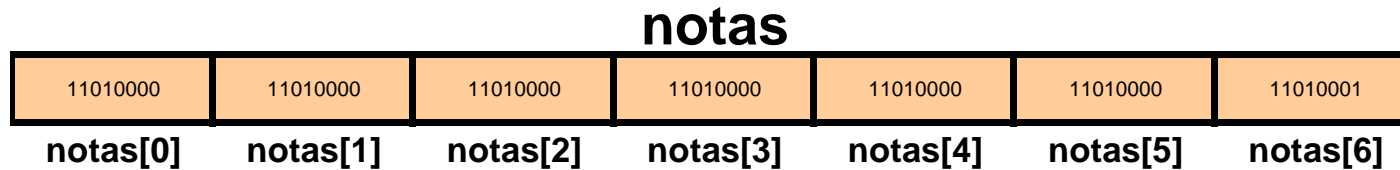
```
double desv_pad = 0.0;
for( int i=0 ; i < 12 ; i++)
    desv_pad += pow( comissao[i] - media , 2.0 );
desv_pad = sqrt ( desv_pad / 12.0 );
printf(" Desvio Padrão :%lf",desv_pad);
```

$$dp = \sqrt{\frac{\sum (x[i] - media)^2}{N}}$$

ecra

Desvio Padrão : 3.26173

# Inicialização automática de vectores



- Quando não inicializados os arrays contêm **LIXO**.

**ERRO !!!!!!!!**

```
int notas[6];
int i;
for(i=0; i< 6; i++)
printf("%d\n",notas[i]);
```

**ecra**

```
0
4202572
2147303424
844745036
1
256
```

# Inicialização automática de vectores

- Inicialização
  - `tipo var[n] = { valor_1, valor_2, . . . , valor_n};`
- Exemplos
  - `int nota[7] = { 10 , 5, 18, 6, 19 ,17,18};`
  - `char vogal[5] = { 'a' , 'e', 'i', 'o', 'u'};`

notas						
10	5	18	6	19	17	18
notas[0]	notas[1]	notas[2]	notas[3]	notas[4]	notas[5]	notas[6]

# Inicialização automática

- Omitir a dimensão na inicialização
  - `char vogal[] = { 'a' , 'e' , 'i' , 'o' , 'u' };`
    - O número de elementos é o número de valores da inicialização
- Valores insuficientes
  - `int nota[7] = { 10 , 5 };`
    - Os valores são introduzidos nos primeiros elementos e os restantes recebem o valor ZERO

notas						
10	5	0	0	0	0	0
notas[0]	notas[1]	notas[2]	notas[3]	notas[4]	notas[5]	notas[6]



# Inicialização automática

## Exemplos correctos:

- `Int v[3] = {1,2,3};`
- `Int v[10] = {0};`
- `Int v[] = {1,2,3};`

## Exemplos Incorrectos:

- `Int v[];`
- `Int v[3] = {};`
- `Int v[2] = {1,3,5};`
- `Int v = {1,2,3};`

## Passagem de vectores para funções

### Escrever um vector de 10 elementos

```
void escreve1(int a[10]){  
    int i;  
    for(i=0; i < 10; i++)  
        printf("%d\t", a[i]);  
}
```

### Escrever um vector de 20 elementos

```
void escreve2(int a[20]){  
    int i;  
    for(i=0; i < 20; i++)  
        printf("%d\t", a[i]);  
}
```

```
int main(int argc, char* argv[])  
{  
    int a[10]= { 1,2,3,4,5};  
    int b[20]= { 1,2,3,4,5,6,7};  
    escreve1(a);  
  
    escreve2(b);  
    return 0;  
}
```



## Passagem de vectores parametrizados

### Escrever um vector de inteiros

```
void escreve( int a[] , int elementos)
{ int i;
  for(i=0; i < elementos; i++)
    printf("%d\t", a[i]);
}
```

```
int main(int argc, char* argv[])
{
  int a[10]= { 1,2,3,4,5};
  int b[20]= { 1,2,3,4,5,6,7};
  escreve(a,10);
  escreve(b,20);
  return 0;
}
```





## Exercício

- Pretende-se um programa que permita extrair informação acerca das notas de um aluno. O programa deve pedir a nota das dez disciplinas e de seguida calcular a nota máxima, a nota média, o número de notas negativas, o número de notas positivas e se está apto para passar de ano (o aluno passa de ano se tiver no máximo 3 notas negativas).
- Modularização
  - Introduzir as notas
  - Calcular a média
  - Calcular o máximo
  - Calcular mínimo
  - Calcular negativas
  - Verificar se está aprovado.
  - Programa principal



# Resolução do Exercício

## Introduzir notas

```
void IntroduzNotas( int notas[] , int elementos){  
    int i;  
    printf("Introducao de Notas \n");  
    for(i=0 ; i < elementos ; i++)  
    {  
        printf("%d Nota:", i + 1);  
        scanf("%d", &notas[i]);  
    }  
}
```

## Calculo da média

```
double Media( int notas[] , int elementos){  
    double soma=0.0;  
    int i;  
    for(i=0 ; i < elementos ; i++)  
        soma += notas[i];  
    return soma / elementos;  
}
```

# Resolução do Exercício

## Nota máxima

```
int NotaMaxima( int notas[] , int elementos) {  
    int maximo = notas[0];  
    int i;  
    for( i=1 ; i < elementos ; i++) {  
        if( notas[i] > maximo )  
            maximo = notas[i];  
    }  
    return maximo;  
}
```

notas

10	5	18	6	19	17	18
notas[0]	notas[1]	notas[2]	notas[3]	notas[4]	notas[5]	notas[6]

# Resolução do Exercício

## Número de Negativas

```
int NumeroNegativas( int notas[] , int elementos){  
    int i;  
    int negativas=0;  
    for(i=0 ; i < elementos ; i++) {  
        if( notas[i] < 10 )  
            negativas ++;  
    }  
    return negativas;  
}
```

notas

10	5	18	6	19	17	18
notas[0]	notas[1]	notas[2]	notas[3]	notas[4]	notas[5]	notas[6]

# Resolução do Exercício

**Aprovado**

```
int Aprovado(int notas[], int elementos)
{
    return NumeroNegativas( notas, elementos) <= 3;
}
```

**notas**

10	5	18	6	19	17	18
notas[0]	notas[1]	notas[2]	notas[3]	notas[4]	notas[5]	notas[6]



# Resolução do Exercício

## Programa principal

```
int main(int argc, char* argv[])
{
    int notasPaulo[10];
    IntroduzNotas( notasPaulo, 10);
    int nm = NotaMaxima( notasPaulo, 10);
    printf("\n Maximo    :%d",nm);;

    double media = Media( notasPaulo, 10);
    printf("\n Media    :%lf", media);;

    printf("\n Negativas : %d",NumeroNegativas(notasPaulo, 10));
    if ( Aprovado( notasPaulo, 10) > 3 )
        printf("\n O Paulo esta aprovado");
    else
        printf("\n O Paulo reprovou");
}
```

## Definição de vectores através de constantes

- Notas de um aluno `int nota[10]`
  - E se o aluno tiver 25 disciplinas
- Definição de constantes
  - Especificador `const`
    - `const int MAX=25;`
  - Directiva `#define`
    - `#define MAX 25`
- Definição do vector
  - `int nota[MAX];`

### Diferenças entre `const` e `#define`

- A constante existe em memória
- A `#define` existe apenas no programa fonte

# Resolução do Exercício

## Programa principal

```
int main(int argc, char* argv[])
{
    const int MAX = 10;
    int notasPaulo[MAX];
    IntroduzNotas( notasPaulo, MAX);
    int nm = NotaMaxima( notasPaulo, MAX);
    printf("\n Maximo   :%d",nm);

    double media = Media( notasPaulo, MAX);
    printf("\n Media    :%lf", media);

    printf("\n Negativas : %d",NumeroNegativas(notasPaulo, MAX));
    if ( Aprovado( notasPaulo, MAX) > 3 )
        printf("\n O Paulo esta aprovado");
    else
        printf("\n O Paulo reprovou");
}
```



## Matrizes (vetores multidimensionais)

- Não existe limite para o número de dimensões de um vector
- Sintaxe
  - tipo nome\_variavel[dim\_1][dim\_2]...[dim\_n]
- Exemplos

- Int notas[20][3]

- Char galo[3][3]

- Galo[0]

- Galo[1]

- Galo[2]

0		
	X	
0		X

0		
---	--	--

	X	
--	---	--

0		X
---	--	---

**Galo[2][0]='O'**  
**Galo[2][2]='X'**

## Matrizes (vetores multidimensionais)

- Inicialização

- `char galo[3][3] = {{'O', ' ', ' '}, {' ', 'X', ' '}, {'O', ' ', 'X'}};`

O		
	X	
O		X

- Passagem como parâmetro

- **Apenas se pode omitir a primeira dimensão**

- `void mostra( char galo[3][3]);`
    - `void mostra( char galo[][3], int dim);`
    - `void mostra( char *galo[3], int dim);`



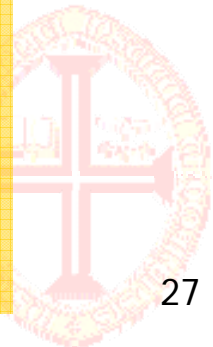
# Jogo do galo

## Ler Jogo

```
void LerJogo(int jogo[][3] , int elementos) {  
    int x, y;  
    for( y=0 ; y < elementos ; y++ ) {  
        for( x =0 ; x < 3 ; x ++  
            scanf("%d",&jogo[y][x]);  
        }  
    }  
}
```

## Imprimir Jogo

```
void MostraJogo(int jogo[][3] , int elementos) {  
    int x , y ;  
    for( int y=0 ; y < elementos ; y++ ) {  
        for( int x =0 ; x < 3 ; x ++  
            printf("%d", &jogo[y][x]);  
            printf("\n");  
        }  
    }  
}
```



# Restrições ao uso de matrizes

- Devolução
  - Uma função não pode devolver uma matriz
- Não é permitida a leitura ou escrita de uma só vez
  
- Não é permitido a utilização de operadores.
  - Aritméticos ou lógicos
    - **NotasTotal = NotasPaulo + NotasMaria;**
  - Disponíveis apenas para cada um dos elementos
    - **NotasTotal[0] = NotasPaulo[0] + NotasMaria[0];**

