

SSC0101 - ICC1 – Teórica

---

Introdução à Ciência da Computação I

**Tipos de Dados Avançados**  
**Vetores e Matrizes**

Prof. Vanderlei Bonato: [vbonato@icmc.usp.br](mailto:vbonato@icmc.usp.br)

Prof. Claudio Fabiano Motta Toledo: [claudio@icmc.usp.br](mailto:claudio@icmc.usp.br)

---

# Sumário

---

- Vetores
  - Vetores em Algoritmos
  - Vetores em C
  
- Matrizes
  - Matrizes em Algoritmos
  - Matrizes em C
  
- Exercícios

# Vetores

---

- **Vetores e matrizes** são estruturas de dados que armazenam diversos valores de um mesmo tipo.
- **Vetores:** Estrutura de dados composta, homogênea e unidimensional.

	0	1	2	3	4	5	6	7	8	9
notas	65.6	70.8	99.0	23.5	78.9	64.3	82.0	92.3	26.4	36..5

	0	1	2	3	4
vogais	a	e	i	o	u

# Vetores em Algoritmos

---

- Declaração:

DECLARE nome\_vetor[tamanho\_vetor] tipo

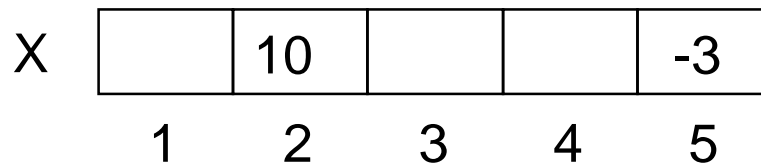
- Exemplo:

DECLARE X[5] NUMÉRICO

- Atribuição

X[2] ← 10

X[5] ← -3



# Vetores em Algoritmos

---

- Inicialização de um vetor:
  - Deve-se atribuir valores a todas as suas posições.
- Exemplo: Algoritmo para inicialização  
PARA  $i \leftarrow 1$  ATÉ 5 FAÇA  
  INÍCIO  
    ESCREVA “Digite o valor de  $x[$ ,  $i$ ,  $]=$ “  
    LEIA  $X[i]$   
  FIM

# Vetores em Algoritmos

Tela		Memória					
i=1	Digite o valor de x[ 1 ]= 10	X	10				
			1	2	3	4	5
i=2	Digite o valor de x[ 2 ]= -2	X	10	-2			
			1	2	3	4	5
i=3	Digite o valor de x[ 3 ]= 0	X	10	-2	0		
			1	2	3	4	5
i=4	Digite o valor de x[ 4 ]= 67	X	10	-2	0	67	
			1	2	3	4	5
i=5	Digite o valor de x[ 5 ]= -88	X	10	-2	0	67	-88
			1	2	3	4	5

# Vetores em C

---

- **Declaração:**

<tipo> <nome\_variável> [<tamanho>]

- **Limitantes e tamanho:**

limitante\_inferior = 0.

limitante\_superior = tamanho-1.

tamanho = limitante\_superior + 1.

- **Exemplo:**

```
int x[10];          //x[0], x[1], x[2],...,x[9]
```

```
float notas[10]; //notas[0], notas[1], notas[2],...,notas[9]
```

```
char vogais[5]; //vogais[0], vogais[1],...,vogais[5]
```

# Vetores em C

---

- Recomenda-se definir o tamanho de um vetor usando uma constante.
- A constante poderá ser utilizada tanto na declaração do vetor quanto na condição de parada dos laços que percorrem o mesmo.
- Exemplo: 

```
#define N 100  
int a[N]; // a[0], a[1], ..., a[99]
```
- Normalmente, utiliza-se um laço “for” para processar os elementos de um vetor.
- Exemplo: 

```
for(i=0; i<N; ++i)  
    sum += a[i];
```



# Vetores em C

---

- Exemplos de inicialização de vetores:

```
float f[5] = {0.0, 1.0, 2.0, 3.0, 4.0};
```

- Quando a lista de valores é menor que o número de elementos, os valores remanescentes são iniciados com valor zero.

```
float a[100] = {0};
```

- Inicia todos os elementos com valor zero.

```
int a[ ] = {2, 3, 5, -7};  $\Leftrightarrow$  int a[4] = {2, 3, 5, -7};
```

- O tamanho do número de elementos do vetor é determinada pela quantidade de valores inicializados.

# Vetores em C

---

- A contagem das posições no vetor começa em 0.
- Incluir mais elementos que o tamanho definido para o vetor é uma fonte de erros.
- Os valores excedentes são atribuídos a uma parte não alocada da memória.
- O espaço alocado em memória é aquele definido quando o vetor foi declarado.

# Vetores em C

---

- Um vetor do tipo “char” pode armazenar “string”
- Note que uma string sempre termina com o caracter *null* (“\0”)
- Exemplo:

```
char nome[4] = "Ana";
```

```
char sobrenome[] = {'H','i','t','s'};
```

```
printf("%s,%d\n",nome,strlen(nome));
```

```
printf("%s,%d\n",sobrenome,strlen(sobrenome));
```

Está correto?



```
char sobrenome[] = {'H','i','t','s','\0'};
```

# Funções para trabalhar com strings

---

- Manipulação de strings:

`strlen();`

`strcat();`

`strcmp();`

`strcpy();`

- Entrada e saída

`gets();`

`puts();`

- Veja mais funções em:

[http://www.acm.uiuc.edu/webmonkeys/book/c\\_guide/](http://www.acm.uiuc.edu/webmonkeys/book/c_guide/)

# Exemplo com vetor

---

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 3
int main(int argc, char *argv[])
{
    int nota[MAX]; nota[0] = 10; nota[1] = 20; nota[2] = 30;
    //int nota[MAX] = {10,20,30};
    //int nota[] = {10,20,30};
    int media,x,acc=0;
    for(x=0; x<MAX;x++){
        acc += nota[x];
        printf("%d\n",nota[x]);
    }
    media = acc/MAX;
    printf("%d\n",media);
    system("PAUSE");
    return 0;
}
```

Note que em C o primeiro elemento do vetor é o índice [0]

# Matrizes

---

- Estrutura de dados composta, homogênea e multidimensional.
- Formada por uma sequência de variáveis do mesmo tipo, com o mesmo identificador (mesmo nome) e alocadas sequencialmente na memória.
- As variáveis são distinguidas pelos índices que referenciam sua localização dentro da estrutura.
- Há um índice para cada uma das dimensões da matriz.

# Matrizes em Algoritmos

- Declaração:

DECLARE nome\_matriz[dim\_1, dim\_2,...,dim\_N] tipo

- Exemplo:

DECLARE tabela[3,6] NUMÉRICO

DECLARE pagina[3,6,2] NUMÉRICO

- Atribuição:

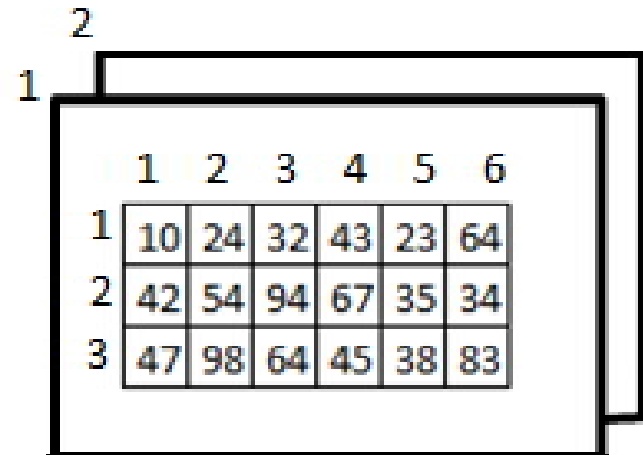
tabela[2,3] ← 94

tabela[3,2] ← 98

	1	2	3	4	5	6
1	10	24	32	43	23	64
2	42	54	94	67	35	34
3	47	98	64	45	38	83

pagina[2,3,1] ← 94

pagina[3,2,1] ← 98



# Matrizes em Algoritmos

---

- Inicialização de uma matriz:
  - Todas as posições da matriz devem ser identificadas.
- Exemplo: Algoritmo para inicialização

```
PARA i ← 1 ATÉ 3 FAÇA
```

```
  INÍCIO
```

```
    PARA j ← 1 ATÉ 5 FAÇA
```

```
      INÍCIO
```

```
        ESCREVA "Digite o valor de x[" , i, " , " , j, "]" = "
```

```
        LEIA X[ i,j]
```

```
      FIM
```

```
    FIM
```



# Matrizes em Algoritmos

i	j	Tela	Memória
1	1	Digite o valor de $x[1,1] = 8$	X 1 8 -2 10 1 34
	2	Digite o valor de $x[1,2] = -2$	2
	3	Digite o valor de $x[1,3] = 10$	3
	4	Digite o valor de $x[1,4] = 1$	1 2 3 4 5
	5	Digite o valor de $x[1,5] = 34$	
2	1	Digite o valor de $x[2,1] = 21$	X 1 8 -2 10 1 34
	2	Digite o valor de $x[2,2] = -43$	2 21 -43 13 8 -2
	3	Digite o valor de $x[2,3] = 13$	3
	4	Digite o valor de $x[2,4] = 8$	1 2 3 4 5
	5	Digite o valor de $x[2,5] = -2$	
3	1	Digite o valor de $x[3,1] = -4$	X 1 8 -2 10 1 34
	2	Digite o valor de $x[3,2] = -10$	2 21 -43 13 8 -2
	3	Digite o valor de $x[3,3] = 25$	3 -4 -10 25 -5 0
	4	Digite o valor de $x[3,4] = -5$	1 2 3 4 5
	5	Digite o valor de $x[3,5] = 0$	

# Matrizes em Algoritmos

---

- Exemplo: Algoritmo para inicialização  $X[4,3,2]$   
PARA  $i \leftarrow 1$  ATÉ 4 FAÇA  
INÍCIO  
PARA  $j \leftarrow 1$  ATÉ 3 FAÇA  
INÍCIO  
PARA  $k \leftarrow 1$  ATÉ 2 FAÇA  
INÍCIO  
ESCREVA "Digite o valor de  $x[$ ",  $i$ ,  $,$ ,  $,$ ,  $j$ ,  $]=$ "  
LEIA  $X[ i,j]$   
FIM  
FIM  
FIM

# Matrizes em Algoritmos

i	j	K	Tela																																									
1	1	1	Digite o valor de $x[1,1,1] = 2$	<p>PROFUNDIDADE 1</p> <table border="1"> <tr><td>1</td><td>2</td><td>-1</td><td>15</td></tr> <tr><td>2</td><td></td><td></td><td></td></tr> <tr><td>3</td><td></td><td></td><td></td></tr> <tr><td>4</td><td></td><td></td><td></td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> </table>	1	2	-1	15	2				3				4					1	2	3	<p>PROFUNDIDADE 2</p> <table border="1"> <tr><td>5</td><td>0</td><td>8</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> </table>	5	0	8														1	2	3
		1	2		-1	15																																						
	2																																											
	3																																											
	4																																											
		1	2		3																																							
5	0	8																																										
	1	2	3																																									
2	Digite o valor de $x[1,1,2] = 5$																																											
2	1	Digite o valor de $x[1,2,1] = -1$																																										
	2	Digite o valor de $x[1,2,2] = 0$																																										
3	1	Digite o valor de $x[1,3,1] = 15$																																										
	2	Digite o valor de $x[1,3,2] = 8$																																										
2	1	1	Digite o valor de $x[2,1,1] = -25$	<p>PROFUNDIDADE 1</p> <table border="1"> <tr><td>1</td><td>2</td><td>-1</td><td>15</td></tr> <tr><td>2</td><td>-25</td><td>6</td><td>7</td></tr> <tr><td>3</td><td></td><td></td><td></td></tr> <tr><td>4</td><td></td><td></td><td></td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> </table>	1	2	-1	15	2	-25	6	7	3				4					1	2	3	<p>PROFUNDIDADE 2</p> <table border="1"> <tr><td>5</td><td>0</td><td>8</td></tr> <tr><td>3</td><td>9</td><td>11</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> </table>	5	0	8	3	9	11											1	2	3
		1	2		-1	15																																						
	2	-25	6		7																																							
	3																																											
	4																																											
		1	2		3																																							
5	0	8																																										
3	9	11																																										
	1	2	3																																									
2	Digite o valor de $x[2,1,2] = 3$																																											
2	1	Digite o valor de $x[2,2,1] = 6$																																										
	2	Digite o valor de $x[2,2,2] = 9$																																										
3	1	Digite o valor de $x[2,3,1] = 7$																																										
	2	Digite o valor de $x[2,3,2] = 11$																																										

# Matrizes em Algoritmos

i	j	K	Tela																																						
3	1	1	Digite o valor de $x[3,1,1] = 23$	PROFUNDIDADE 1 <table border="1"> <tr><td>1</td><td>2</td><td>-1</td><td>15</td></tr> <tr><td>2</td><td>-25</td><td>6</td><td>7</td></tr> <tr><td>3</td><td>23</td><td>-5</td><td>19</td></tr> <tr><td>4</td><td></td><td></td><td></td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> </table>	1	2	-1	15	2	-25	6	7	3	23	-5	19	4					1	2	3	PROFUNDIDADE 2 <table border="1"> <tr><td>5</td><td>0</td><td>8</td></tr> <tr><td>3</td><td>9</td><td>11</td></tr> <tr><td>-2</td><td>46</td><td>1</td></tr> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> </table>	5	0	8	3	9	11	-2	46	1					1	2	3
		1	2		-1	15																																			
	2	-25	6		7																																				
	3	23	-5		19																																				
	4																																								
		1	2		3																																				
5	0	8																																							
3	9	11																																							
-2	46	1																																							
	1	2	3																																						
2	Digite o valor de $x[3,1,2] = -2$																																								
2	1	Digite o valor de $x[3,2,1] = -5$																																							
	2	Digite o valor de $x[3,2,2] = 46$																																							
3	1	Digite o valor de $x[3,3,1] = 19$																																							
	2	Digite o valor de $x[3,3,2] = 1$																																							
4	1	1	Digite o valor de $x[4,1,1] = 14$	PROFUNDIDADE 1 <table border="1"> <tr><td>1</td><td>2</td><td>-1</td><td>15</td></tr> <tr><td>2</td><td>-25</td><td>6</td><td>7</td></tr> <tr><td>3</td><td>23</td><td>-5</td><td>19</td></tr> <tr><td>4</td><td>14</td><td>5</td><td>10</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> </table>	1	2	-1	15	2	-25	6	7	3	23	-5	19	4	14	5	10		1	2	3	PROFUNDIDADE 2 <table border="1"> <tr><td>5</td><td>0</td><td>8</td></tr> <tr><td>3</td><td>9</td><td>11</td></tr> <tr><td>-2</td><td>46</td><td>1</td></tr> <tr><td>27</td><td>4</td><td>65</td></tr> <tr><td></td><td>1</td><td>2</td><td>3</td></tr> </table>	5	0	8	3	9	11	-2	46	1	27	4	65		1	2	3
		1	2		-1	15																																			
	2	-25	6		7																																				
	3	23	-5		19																																				
	4	14	5		10																																				
		1	2		3																																				
5	0	8																																							
3	9	11																																							
-2	46	1																																							
27	4	65																																							
	1	2	3																																						
2	Digite o valor de $x[4,1,2] = 27$																																								
2	1	Digite o valor de $x[4,2,1] = 5$																																							
	2	Digite o valor de $x[4,2,2] = 4$																																							
3	1	Digite o valor de $x[4,3,1] = 10$																																							
	2	Digite o valor de $x[4,3,2] = 65$																																							

# Matrizes em C

---

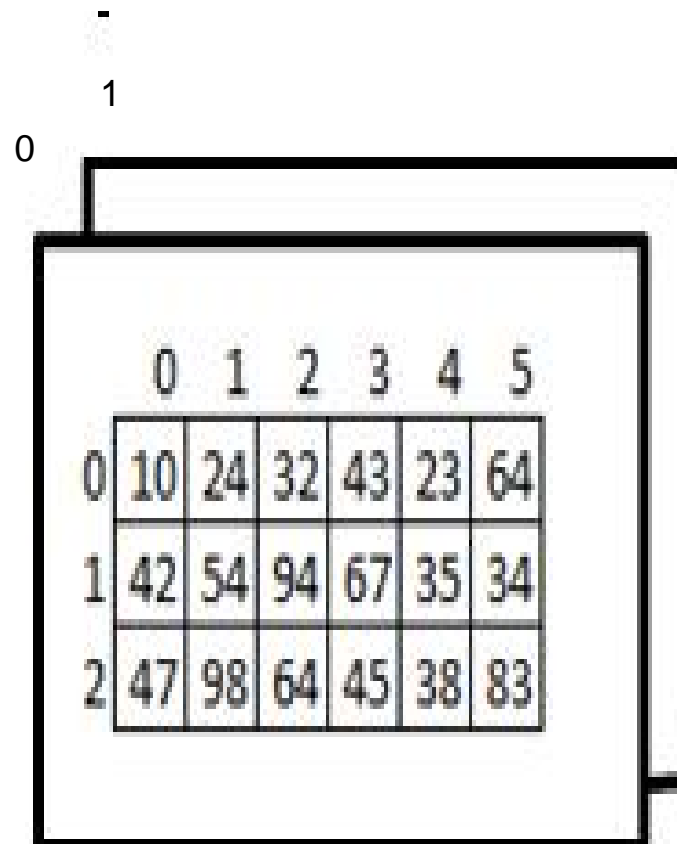
- Declaração:  
tipo nome\_variável[tamanho\_dim1] [tamanho\_dim2]  
[tamanho\_dim3].. [tamanho\_dimM];
  - Exemplo de declaração de uma matriz:  
int tabela [3][5];
    - Tabela com 3 linhas e 5 colunasint paginas[3][5][2];
    - Estrutura com 3 linhas, 5 colunas e 2 tabelas de profundidade
-

# Matrizes em C

	0	1	2	3	4	5
0	10	24	32	43	23	64
1	42	54	94	67	35	34
2	47	98	64	45	38	83

tabela[1][2] = 94

tabela[2][1] = 98



pagina [1][2][0] = 94

pagina [2][1][0] = 98

# Matrizes em C

- Exemplos de inicializações:

```
int tabela[3][5] = { {10,24,32,43,23,64},  
                    {42,54,94,67,35,34},  
                    {47,98,64,45,38,83} };
```

	0	1	2	3	4	5
0	10	24	32	43	23	64
1	42	54	94	67	35	34
2	47	98	64	45	38	83

Dica: pense as inicializações da direita para a esquerda.

```
float mat_A[1][2][3] = {  
    { {5.2,0.9,1.3}, {0.8,4.5,2.3} }  
};
```

```
float mat_B[1][2][3][4] = {  
    {  
        { {5.2,0.9,1.3,4.2}, {0.8,4.5,2.3,6.4}, {3.2,3.4,6.3,9.0} },  
        { {8.1,3.4,6.3,7.1}, {2.3,6.1,0.3,9.2}, {1.1,3.5,0.1,7.2} }  
    }  
};
```

# Matrizes em C

---

- Exemplos de inicializações (cont.):
  - `int a[2][3] = {1,2,3,4,5,6};`  $\Leftrightarrow$  `int a[2][3] = {{1,2,3}, {4,5,6}};`  
 $\Leftrightarrow$  `int a[ ][3] = {{1,2,3},{4,5,6}};`
  - `int a[2][][2][3] = { { {1,1,0}, {2,0,0} },`  
`{ {3,0,0}, {4,4,0} } };`  
 $\Leftrightarrow$  `int a[ ][2][3] = { { {1,1}, {2} }, { {3}, {4,4} } };`
  - `int a[2][2][3] = {0};` //inicia todas as posições com zero



```

1  #include<stdio.h>
2  #include<string.h>
3  #include<stdlib.h>
4
5  void main(){
6
7  int ln, cl, pf;
8
9  //Iniciando Matriz
10 float matriz[2][3][4] = {
11     { {5.2,0.9,1.3,4.2}, {0.8,4.5,2.3,6.4}, {3.2,3.4,6.3,9.0}},
12     { {8.1,3.4,6.3,7.1}, {2.3,6.1,0.3,9.2}, {1.1,3.5,0.1,7.2}}
13 };
14 //Percorrendo Matriz
15 for(ln=0; ln<2; ln++)
16     for(cl=0; cl<3; cl++)
17         for(pf=0; pf<4; pf++)
18             printf("matriz[%d][%d][%d]=%3.1f\n", ln, cl, pf, matriz[ln][cl][pf]);
19 }

```

```

matriz[0][0][0]=5.2  matriz[1][0][0]=8.1
matriz[0][0][1]=0.9  matriz[1][0][1]=3.4
matriz[0][0][2]=1.3  matriz[1][0][2]=6.3
matriz[0][0][3]=4.2  matriz[1][0][3]=7.1
matriz[0][1][0]=0.8  matriz[1][1][0]=2.3
matriz[0][1][1]=4.5  matriz[1][1][1]=6.1
matriz[0][1][2]=2.3  matriz[1][1][2]=0.3
matriz[0][1][3]=6.4  matriz[1][1][3]=9.2
matriz[0][2][0]=3.2  matriz[1][2][0]=1.1
matriz[0][2][1]=3.4  matriz[1][2][1]=3.5
matriz[0][2][2]=6.3  matriz[1][2][2]=0.1
matriz[0][2][3]=9.0  matriz[1][2][3]=7.2

```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <conio.h>
4
5  int main()
6  {
7      int i, j, k, m;
8      float mat_B[1][2][3][4] = {
9          {
10             {
11                 {5.2,0.9,1.3,4.2}, {0.8,4.5,2.3,6.4}, {3.2,3.4,6.3,9.0}},
12                 {8.1,3.4,6.3,7.1}, {2.3,6.1,0.3,9.2}, {1.1,3.5,0.1,7.2}}
13             }
14         };
15
16     for(i=0; i<1; i++)
17         for(j=0; j<2; j++)
18             for(k=0; k<3; k++)
19                 for(m=0; m<4; m++)
20                     printf("\nmat_B[%d][%d][%d][%d]=%f", i, j, k, m, mat_B[i][j][k][m]);
21
22     return 0;
23 }
24
25
26

```

```

mat_B[0][0][0][0]=5.200000
mat_B[0][0][0][1]=0.900000
mat_B[0][0][0][2]=1.300000
mat_B[0][0][0][3]=4.200000
mat_B[0][0][1][0]=0.800000
mat_B[0][0][1][1]=4.500000
mat_B[0][0][1][2]=2.300000
mat_B[0][0][1][3]=6.400000
mat_B[0][0][2][0]=3.200000
mat_B[0][0][2][1]=3.400000
mat_B[0][0][2][2]=6.300000
mat_B[0][0][2][3]=9.000000
mat_B[0][1][0][0]=8.100000
mat_B[0][1][0][1]=3.400000
mat_B[0][1][0][2]=6.300000
mat_B[0][1][0][3]=7.100000
mat_B[0][1][1][0]=2.300000
mat_B[0][1][1][1]=6.100000
mat_B[0][1][1][2]=0.300000
mat_B[0][1][1][3]=9.200000
mat_B[0][1][2][0]=1.100000
mat_B[0][1][2][1]=3.500000
mat_B[0][1][2][2]=0.100000
mat_B[0][1][2][3]=7.200000

```

# Matrizes em C

---

- Numa declaração do tipo

```
int a[7][9][2]
```

o compilador irá alocar espaço para 7x9x2 valores inteiros contíguos.

- O mapeamento desses valores faz com que  
$$a[i][j][k] \Leftrightarrow *(&a[0][0][0] + 9*2*i + 2*j + k)$$

# Exercício I

---

- Escreva um programa que encontre o maior e o menor número de um vetor de inteiros de 30 elementos e mostre as suas respectivas posições;
  - Inicialize o vetor com números randômicos
    - Ex:

```
int x;  
x = rand();
```

## Exercício II

---

- Faça um programa que preencha um primeiro vetor com dez números inteiros e um segundo vetor com cinco números inteiros. O programa deverá mostrar uma lista dos números do primeiro vetor com seus respectivos divisores armazenados no segundo vetor, bem como suas posições
  - Para saber se um número inteiro é divisível por outro, deve-se calcular o resto da divisão
    - Exemplo em C

```
int x = 10, y = 2;
printf("%d\n",x%y);
```

## Exercício III

---

- Escreva um programa que gere duas matrizes A e B quadradas de ordem 10 e que realize as seguintes operações:
  - transposta de A ( $A^t$ );
  - produto de  $AxB$ ;

Imprima o conteúdo de A e B e o resultados das operações

# Referências

---

Ascencio AFG, Campos EAV. Fundamentos de programação de computadores. São Paulo : Pearson Prentice Hall, 2006. 385 p.

Kelley, A.; Pohl, I., *A Book on C: programming in C*. 4ª Edição. Massachusetts: Pearson, 2010, 726p.

Kernighan, B.W.; Ritchie, D.M. C, *A Linguagem de Programação: padrão ANSI*. 2ª Edição. Rio de Janeiro: Campus, 1989, 290p.

MIZRAHI, V. V., *Treinamento em Linguagem C – Módulo 1 e Módulo 2*, Makron Books, 1990.

---

# FIM Aula 10

---