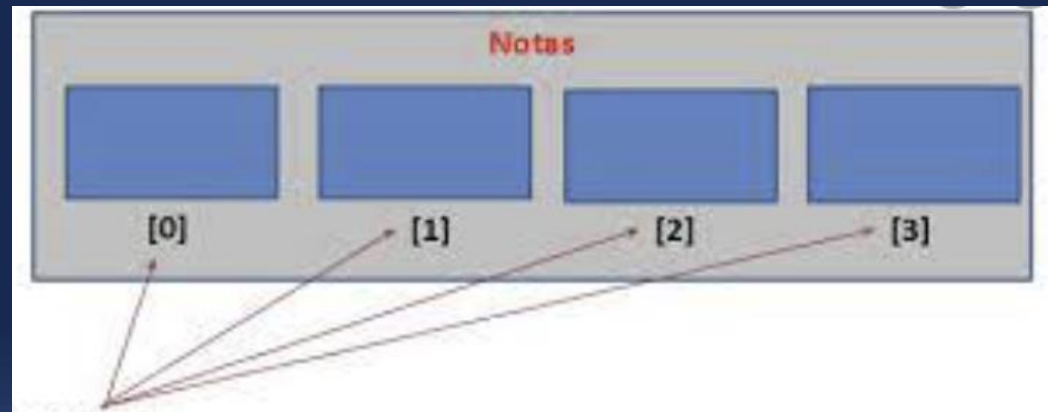




Programação Avançada e Estrutura de Dados

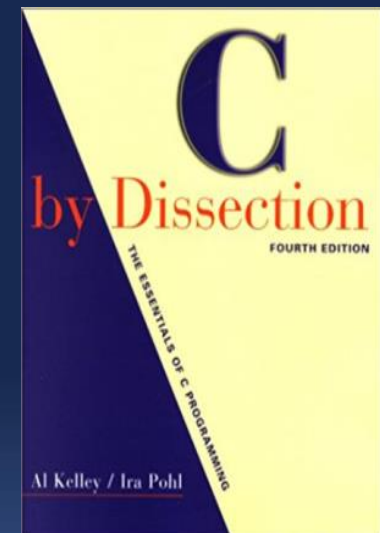
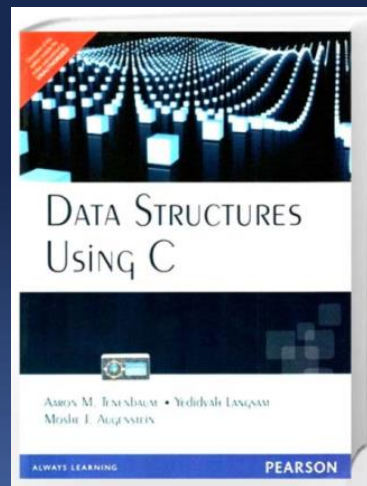
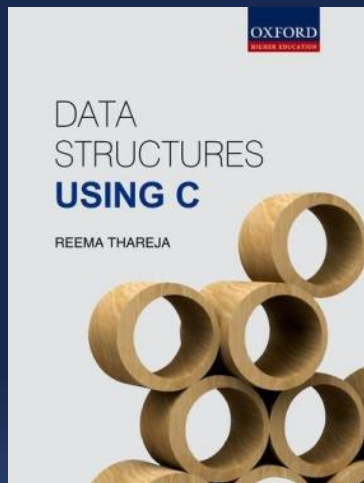
Unidade 5 – Arrays



Prof. Aparecido V. de Freitas
Doutor em Engenharia
da Computação pela EPUSP
aparecido.freitas@online.uscs.edu.br
aparecidovfreitas@gmail.com

Bibliografia

- ✓ Data Structures using C - Oxford University Press - 2014
- ✓ Data Structures Using C - A. Tenenbaum, M. Augensem, Y. Langsam, Pearson 1995
- ✓ C By Dissection - Kelley, Pohl - Third Edition - Addison Wesley



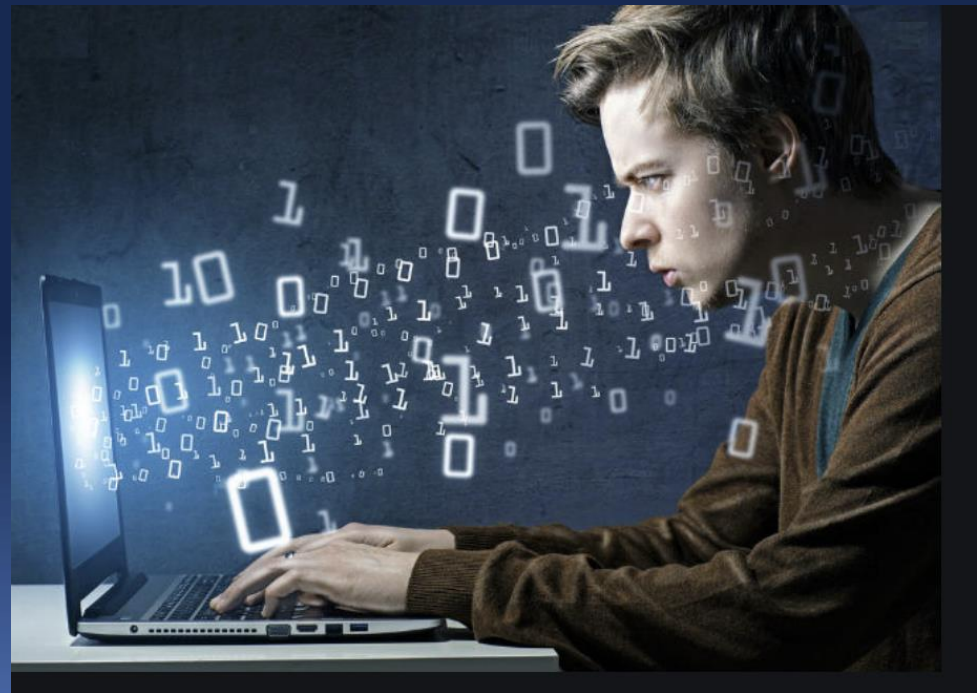
Introdução

- ▣ Um programador recebeu uma incumbência de escrever um programa **na linguagem C** que irá manipular **1000** números...



O que fazer ?

- ❖ Com os **tipos básicos** vistos nas unidades anteriores, cada identificador corresponde à uma **única variável**.
- ❖ Mas, como proceder para manusearmos um conjunto de valores do mesmo tipo ?
- ❖ Por exemplo: os primeiros **1000** números primos.



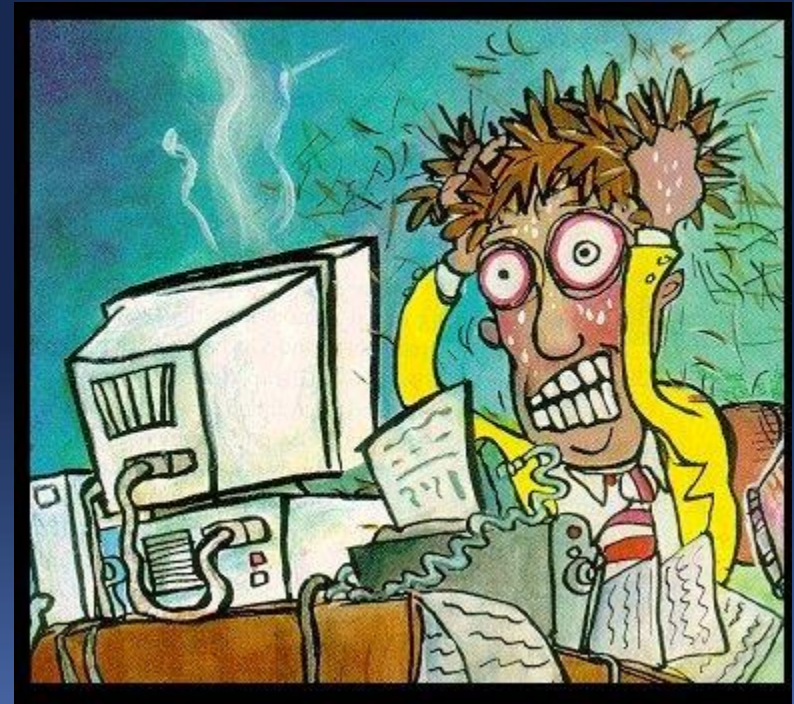
Uma alternativa...

- ❖ Criar **1000** variáveis, cada uma com um determinado nome...



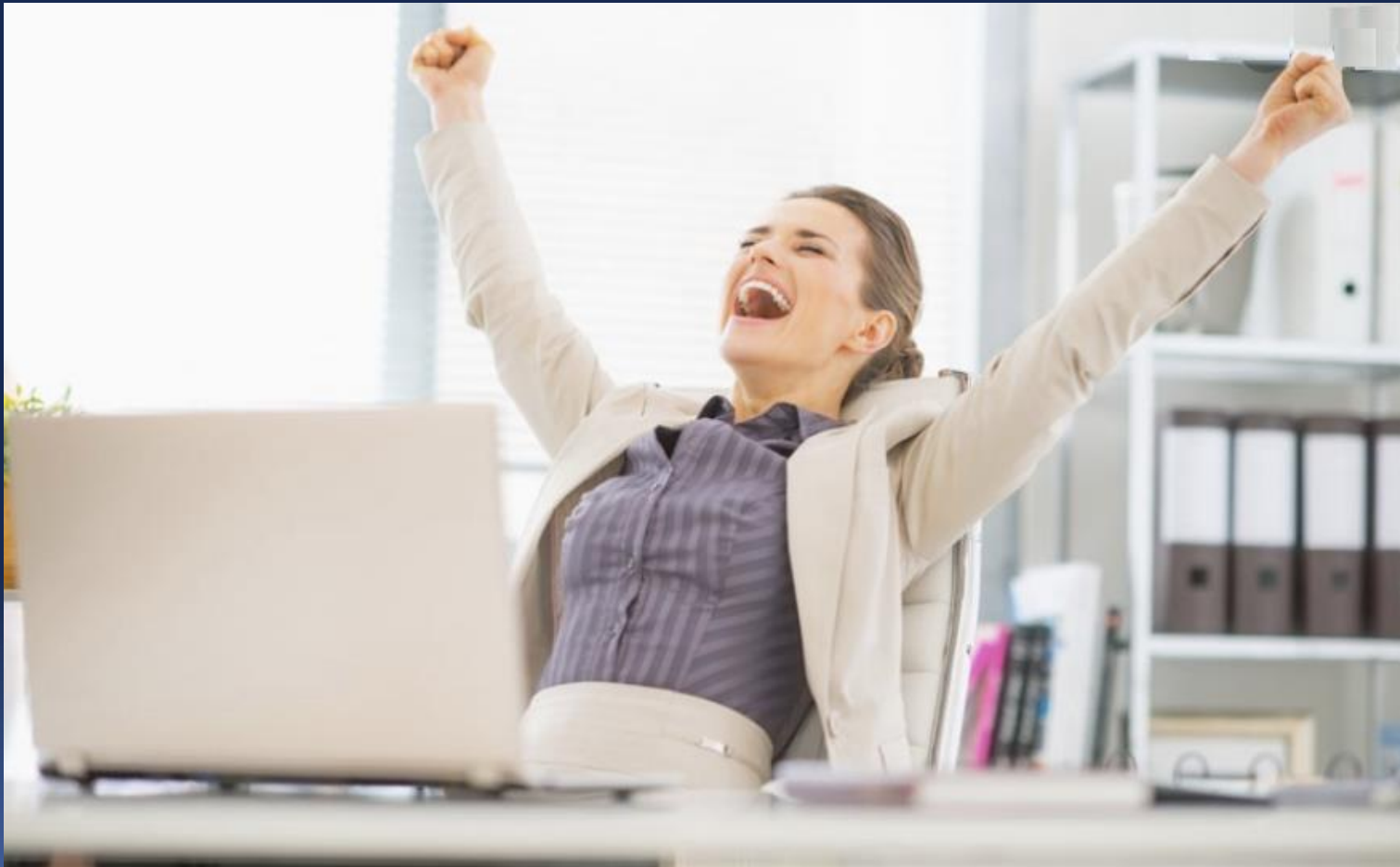
Alternativa inviável ...

- O programa teria **1 000 variáveis** ...
- A **tabela de símbolos** certamente seria **difícil** de ser **manipulada...**



Outra alternativa...

❖ Empregar arrays ...



Arrays

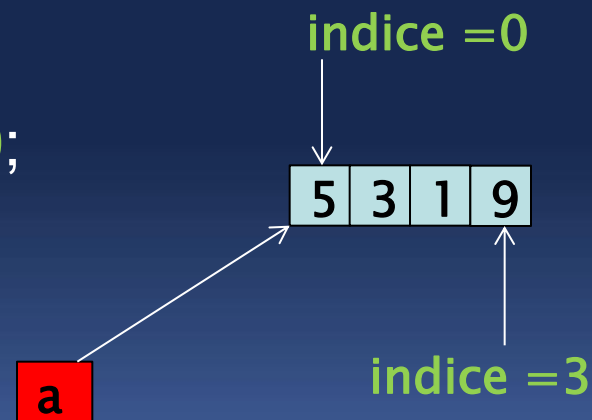
- Trata-se de automatizar a declaração de um grande número de dados de um mesmo tipo simples.
- As variáveis assim declaradas são acessadas por meio de um índice de tipo de dados, por exemplo: **int**.

- Declaração:

- ✓ `int a[4];`

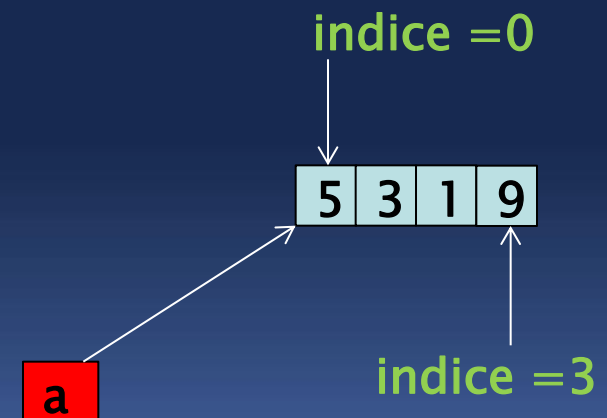
- ✓ Índice da primeira posição = 0;

- ✓ Índice da última posição = 3;



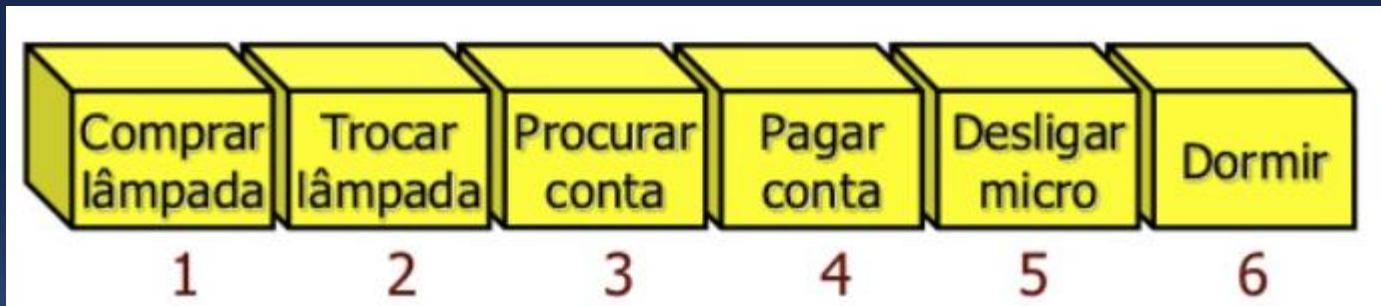
Arrays

- Declaração:
 - ✓ `int a[4];`
 - ✓ Índice da primeira posição = 0;
 - ✓ Índice da última posição = 3;
- Atribuição:
 - ✓ `a [1] = 3;`
- Acesso a um valor do array:
 - ✓ `var = a[0];`



Arrays

- Podem ter mais de **1 dimensão**;
- Arrays com uma dimensão correspondem à arranjos lineares de dados (**Listas**);



Arrays

- Arrays com **duas** dimensões correspondem à tabelas (aplicações em **matrizes** de dados).

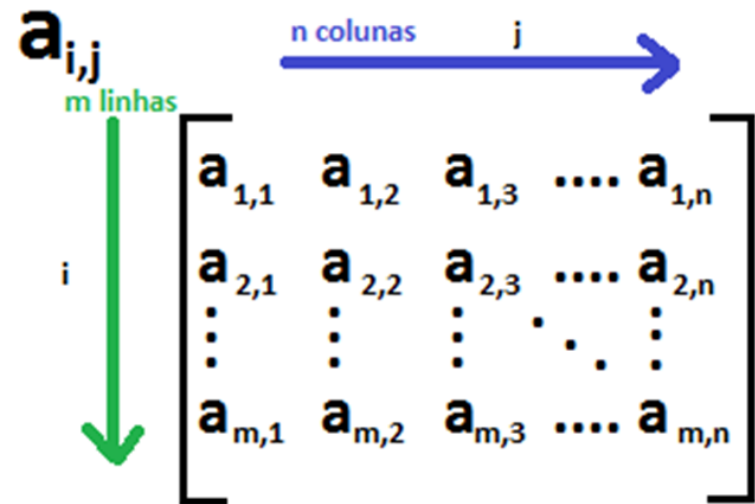
NOME	SEXO	IDADE	ALTURA	PESO
João Miguel	Masculino	25	1,90	78
Susana Nogueira	Feminino	23	1,81	104
Rita Susana	Feminino	22	1,72	62
Carlos Piedade	Masculino	29	1,83	92

Matrizes

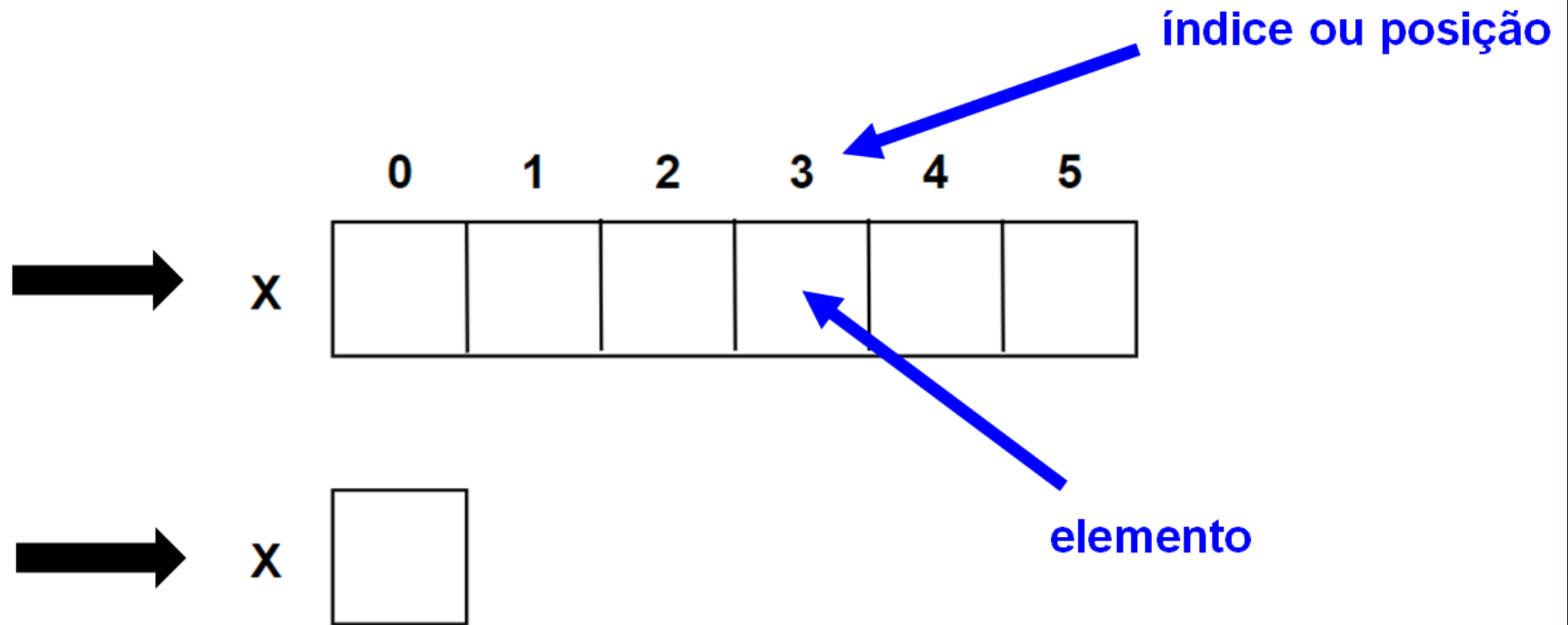


$$A = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 2 & 1 & 1 & 1 \\ 2 & 3 & 0 & 1 \\ -1 & 1 & 2 & 2 \end{bmatrix}$$

Matriz **m** por **n**



Arrays com uma dimensão



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {

    printf("\nInicio do Programa 01");

    int  vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");

}
```



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>


int main() {

    printf("\nInicio do Programa 01");

    int vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```



0	1	2	3	4	5	6	7	8	9



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>


int main() {

    printf("\nInicio do Programa 01");

    int  vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```



0	1	2	3	4	5	6	7	8	9
76									



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>


int main() {

    printf("\nInicio do Programa 01");

    int vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```



0	1	2	3	4	5	6	7	8	9
76	22								



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {


    printf("\nInicio do Programa 01");

    int  vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");

}
```



0	1	2	3	4	5	6	7	8	9
76	22	10							



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {


    printf("\nInicio do Programa 01");

    int  vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");

}
```



0	1	2	3	4	5	6	7	8	9
76	22	10	5						



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {

    printf("\nInicio do Programa 01");

    int vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```

0	1	2	3	4	5	6	7	8	9
76	22	10	5	36					



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {

    printf("\nInicio do Programa 01");

    int vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```

0	1	2	3	4	5	6	7	8	9
76	22	10	5	36	67				



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {

    printf("\nInicio do Programa 01");

    int  vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```

0	1	2	3	4	5	6	7	8	9
76	22	10	5	36	67	89			



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {

    printf("\nInicio do Programa 01");

    int  vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```

0	1	2	3	4	5	6	7	8	9
76	22	10	5	36	67	89	92		



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {

    printf("\nInicio do Programa 01");

    int  vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```

0	1	2	3	4	5	6	7	8	9
76	22	10	5	36	67	89	92	15	



Arrays com uma dimensão



```
//Programa 01 - Unidade 7
#include <stdio.h>

int main() {

    printf("\nInicio do Programa 01");

    int  vetor[10];

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    printf("\nFim do Programa 01");
}
```

0	1	2	3	4	5	6	7	8	9
76	22	10	5	36	67	89	92	15	28



Arrays com uma dimensão

```
//Programa 02 - Unidade 7
#include <stdio.h>

int main() {
    printf("\n*****");
    printf("\nInicio do Programa 02");

    int  vetor[10],i;

    vetor[0]= 76;
    vetor[1]= 22;
    vetor[2]= 10;
    vetor[3]= 5;
    vetor[4]= 36;
    vetor[5]= 67;
    vetor[6]= 89;
    vetor[7]= 92;
    vetor[8]= 15;
    vetor[9]= 28;

    for(i=0; i < 10; i++)
        printf("\n %d", vetor[i]);

    printf("\nFim do Programa 02");
    printf("\n*****");
}
```

Arrays com uma dimensão

```
*****
Inicio do Programa 02
76
22
10
5
36
67
89
92
15
28
Fim do Programa 02
*****
```

Inicializando arrays

```
//Programa 03 - Unidade 7
#include <stdio.h>

int main() {
    printf("\n*****");
    printf("\nInicio do Programa 03");

    int  vetor[5]= { 2,4,6,8,10},i;

    for(i=0; i < 5; i++)
        printf("\n %d", vetor[i]);

    printf("\nFim do Programa 03");
    printf("\n*****");
}
```

Inicializando arrays

E:\USCS\DISCIPLINAS_USCS\Disciplinas_1S_2021\Alg_Est_Dados_I\Fontes_C\Unidade_7\Pgm_03.exe

Início do Programa 03

2

4

6

8

10

Fim do Programa 03

Process exited after 0.01946 se
Press any key to continue . . .

Inicializando arrays

**NÃO
TEM**



```

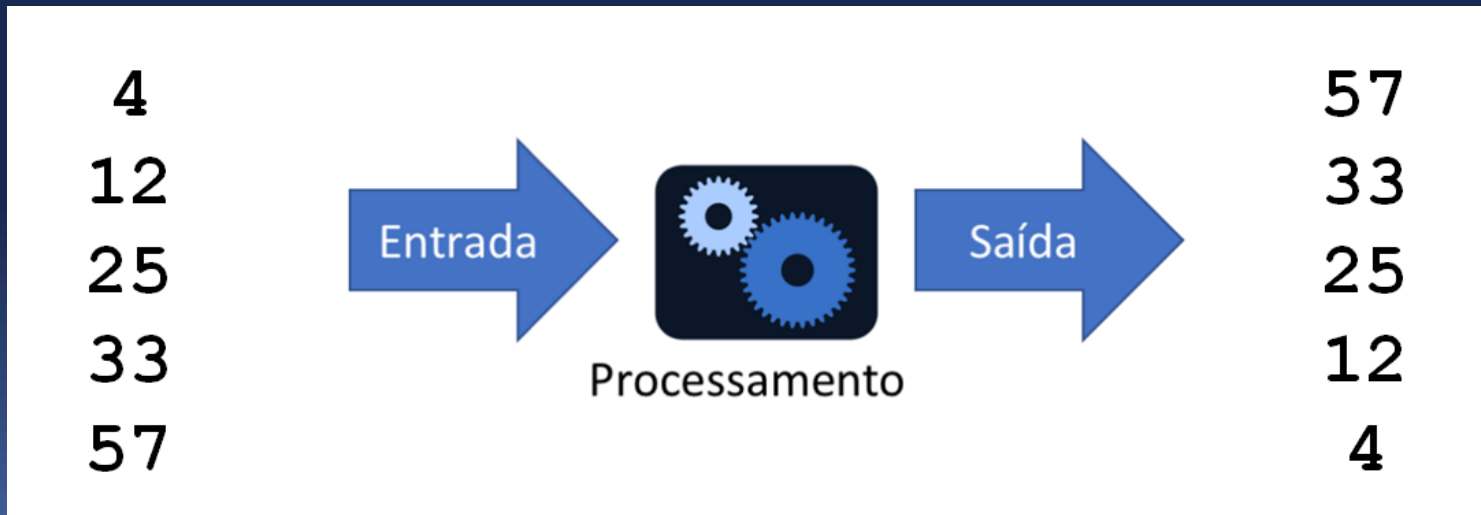
E:\USCS\DISCIPLINAS_USCS\Disciplinas_1S_2021\Alg_1st_Dados_1\Fontes_C\Unidade_7\Pgm_03.exe

*****
Inicio do Programa 03
2
4
6
8
10
Fim do Programa 03
*****

-----
Process exited after 0.01946 se
Press any key to continue . . .
  
```

Programa 04

- Escrever um programa na **Linguagem C** no qual o usuário digita **5 valores numéricos inteiros**. O programa deverá exibir os números na **ordem inversa**.
- Entrada e saída conforme Figura abaixo:



//Programa 04 - Unidade 7

```
#include <stdio.h>
```

```
#include <locale.h>
```

```
int main() {
```

```
    setlocale(LC_ALL, "Portuguese");
```

```
    printf("\n*****");
```

```
    printf("\nInicio do Programa 04");
```

```
    int  vetor[5], i, trab;
```

```
    for(i=0; i < 5; i++) {
```

```
        printf("\nEntre com um valor numérico inteiro: ");
```

```
        scanf("%d", &trab);
```

```
        vetor[i] = trab;
```

```
    }
```

```
    for(i=4; i >= 0; i--)
```

```
        printf("\n%d", vetor[i]);
```

```
    printf("\nFim do Programa 04");
```

```
    printf("\n*****");
```

```
}
```

Início do Programa 04

Entre com um valor numérico inteiro: 4

Entre com um valor numérico inteiro: 12

Entre com um valor numérico inteiro: 25

Entre com um valor numérico inteiro: 33

Entre com um valor numérico inteiro: 57

57

33

25

12

4

Fim do Programa 04

Programa 05

- Escrever um programa na **Linguagem C** no qual o usuário digita **5 valores numéricos inteiros**. O programa deverá exibir o total de valores entrados que estão **acima** da **média**.

Programa 05 – Sem Arrays



```
#include <stdio.h>
int main()
{
    int contador=0;
    float  media, valor1, valor2, valor3, valor4, valor5;

    //scanf ("%f %f %f %f %f",
    //          &n1, &n2, &n3, &n4, &n5);

    printf ("\nEntre com o primeiro valor: ");
    scanf ("%f", &valor1);

    printf ("\nEntre com o segundo valor: ");
    scanf ("%f", &valor2);

    printf ("\nEntre com o terceiro valor: ");
    scanf ("%f", &valor3);

    printf ("\nEntre com o quarto valor: ");
    scanf ("%f", &valor4);

    printf ("\nEntre com o quinto valor: ");
    scanf ("%f", &valor5);

    media = (valor1+ valor2 + valor3 + valor4 + valor5)/5;

    if (valor1>media) contador++;
    if (valor2>media) contador++;
    if (valor3>media) contador++;
    if (valor4>media) contador++;
    if (valor5>media) contador++;

    printf ("\n\nMedia = %f  Total de valores acima da media: %d", media, contador);
    return 0;
}
```



Programa 05 – Com Arrays



```
#include <stdio.h>
int main() {

    printf("\n\n==== Inicio do Programa ==== \n\n");

    int i, contador=0;
    float soma=0, media;
    float v[5];

    for (i=0;i<5;i++) {
        printf("\nEntre com o valor: ");
        scanf ("%f", &v[i]);
        soma = soma + v[i];
    }

    media = soma/i;

    for (i=0;i<5;i++) {

        if (v[i]>media) contador++;

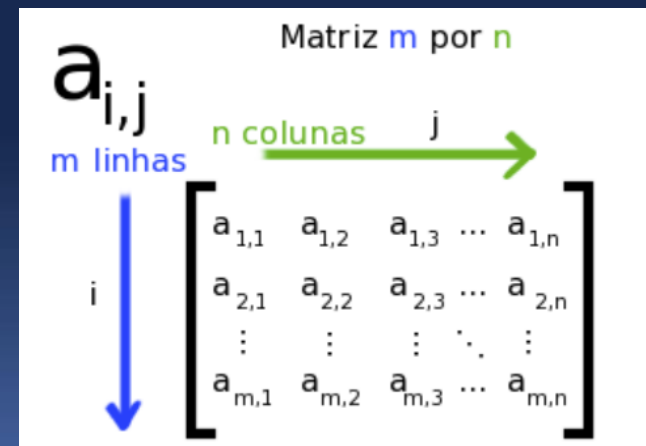
    }
    printf ("\n\nMedia: %f \n\nValores acima da media:  %d\n", media, contador);

    printf("\n\n===== Fim de Programa ===== \n\n");
}
```

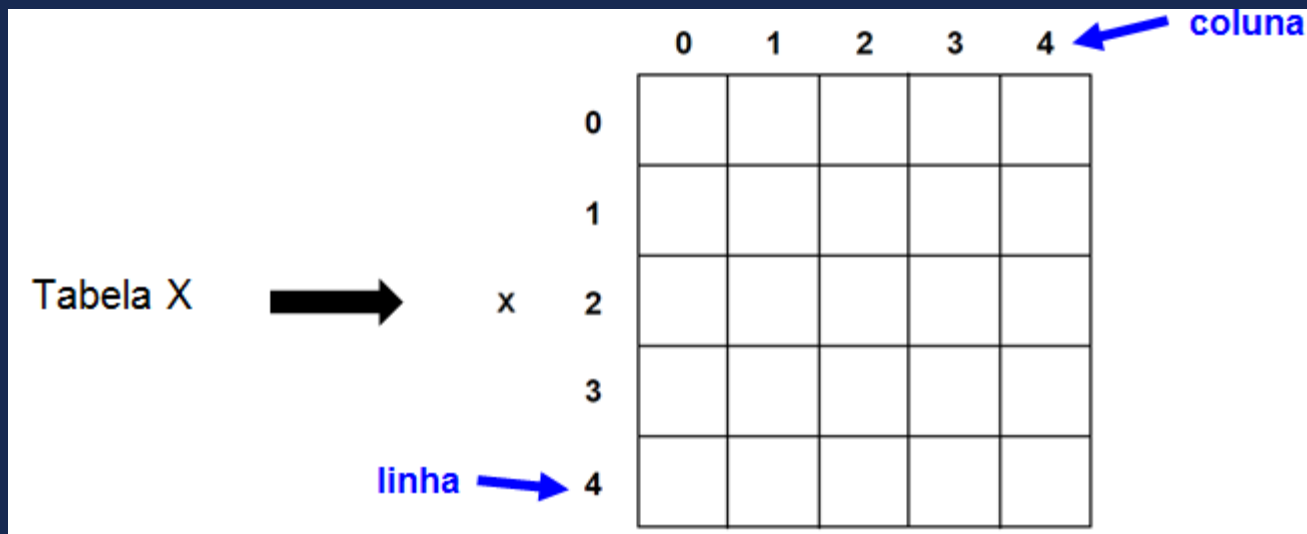


Arrays de 2 dimensões (Tabelas)

- Pode-se organizar os dados na forma de **tabelas** com matrizes de duas dimensões.
- Os elementos podem também são manipulados individualmente, com a referência feita sempre por meio de **dois índices**: o primeiro para controlarmos a **linha**; o segundo, para controlarmos a **coluna**.



Arrays de 2 dimensões (Tabelas)



Programa 06

**Não
Tem**

Entrada



Processamento

Saída

35	26	12
52	74	27
93	28	49
44	60	98

Programa 06



```
//Programa 06 - Unidade 7
```

```
#include <stdio.h>
```

```
#include <locale.h>
```

```
int main() {  
    setlocale(LC_ALL, "Portuguese");  
    printf("\n*****");  
    printf("\nInício do Programa 06\n");
```

```
    int tab[4][3], i,j;
```

```
    tab[0][0] = 35;
```

```
    tab[0][1] = 26;
```

```
    tab[0][2] = 12;
```

```
    tab[1][0] = 52;
```

```
    tab[1][1] = 74;
```

```
    tab[1][2] = 27;
```

```
    tab[2][0] = 93;
```

```
    tab[2][1] = 28;
```

```
    tab[2][2] = 49;
```

```
    tab[3][0] = 44;
```

```
    tab[3][1] = 60;
```

```
    tab[3][2] = 98;
```

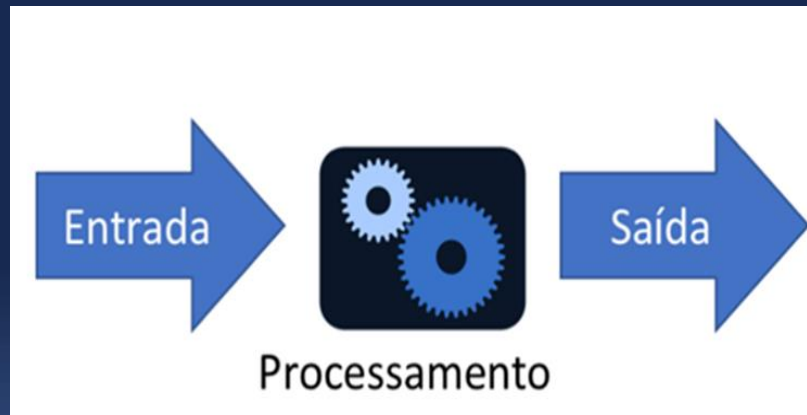
```
    for (i=0; i<4; i++) {  
        printf("\nLinha %d ---> ", i);  
        for (j=0; j<3; j++)  
            printf("%d ", tab[i][j]);  
        printf("\n");  
    }  
    printf("\nFim do Programa 06");  
    printf("\n*****");  
  
    return 0;  
}
```



Programa 07

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

Escreva um programa na **Linguagem C** que receba **16** números inteiros e os armazene em uma matriz **4x4**. Em seguida o programa deverá **exibir** o conteúdo dessa matriz.



10	11	12	13
14	15	16	17
18	19	20	21
22	23	24	25

Programa 07



```
//Programa 07 - Unidade 7
#include <stdio.h>
#include <locale.h>

int main() {
    setlocale(LC_ALL, "Portuguese");
    printf("\n*****");
    printf("\nInício do Programa 07\n");

    int tab[4][4], i, j, trab;

    for (i=0; i<4; i++) {
        for (j=0; j<4; j++) {
            printf("\nEntre com um valor numérico: ");
            scanf("%d", &trab);
            tab[i][j] = trab;
        }
    }
}
```

```
for (i=0; i<4; i++) {
    printf("\nLinha %d ---> ", i);
    for (j=0; j<4; j++)
        printf("%d ", tab[i][j]);
}

printf("\nFim do Programa 07");
printf("\n*****");

return 0;
}
```

