



## Lista 2 - Estruturas Homogêneas

### Vetores numéricos, funções e arquivos de cabeçalho

#### Exercícios para sala de aula

##### Observações:

- ✓ Vetores precisam ter tamanho definido quando são declarados.
- ✓ Cuidado para não ultrapassar o tamanho do vetor, ou seja, percorrer índices (ler) ou armazenar valores além do tamanho definido para o vetor.
- ✓ Para gerar números aleatório utilizar a função **rand()** que está na biblioteca **stdlib.h** e necessário incluir também **time.h** para usar a função **time(NULL)**. Inicialmente declarar **srand(time(NULL))**; para que seja gerado um início (semente) aleatória para a função **rand()**; depois utilizar **rand()**, que pode ser **rand()** / dividido por alguma constante ou variável ou **rand()** % resto de alguma constante ou variável para obter valores em uma determinada faixa.
- ✓ As funções **srand()** e **rand()** requerem a biblioteca **stdlib.h** e a função **time()** requer a biblioteca **time.h**.

**ATENÇÃO:** Usando a biblioteca **Vetores.h** (criada no Exercício 2 da Lista1), apenas adapte a função **GerarVetorInteiros()** para que gere elementos aleatórios no intervalo de 1 até o limite. Use essa função e a função **MostrarVetorInteiros()** para resolver os exercícios 1 a 6 a seguir. Em todos os exercícios, implemente a repetição de programa.

1) Gerar um vetor de inteiros com 15 elementos aleatórios entre 1 e 25. Gerar outros dois vetores, um contendo os valores pares e outro os ímpares. Mostrar os três vetores.

Exemplo:

```

=== VETOR COMPLETO ===
7 17 1 9 25 21 16 12 17 4 14 23 19 18 21

=== VETOR DE PARES ===
16 12 4 14 18

=== VETOR DE IMPARES ===
7 17 1 9 25 21 17 23 19 21

```

Deseja repetir o programa (S ou N)?

2) Gerar um vetor A de inteiros com 8 elementos aleatórios entre 1 e 10. Construir um vetor B de mesma dimensão com os elementos de A multiplicados por 2.

Exemplo:

```

=== VETOR A ===
9 1 3 5 4 5 8 10

=== VETOR B ===
18 2 6 10 8 10 16 20

```

Deseja repetir o programa (S ou N)?

3) Gerar um vetor A de inteiros com 7 elementos aleatórios entre 1 e 10. Construir um vetor B de mesmo tamanho, sendo que cada elemento de B deverá ser o elemento de A correspondente multiplicado por sua posição (ou índice), ou seja,  $B[i] = A[i] * i$ .

Exemplo:

```

=== VETOR A ===
4 1 10 4 1 4 8

=== VETOR B ===
0 1 20 12 4 20 48

```

Deseja repetir o programa (S ou N)?

4) Gerar um vetor A de inteiros com 10 elementos aleatórios entre 1 e 10 e construir um vetor B de mesma dimensão com os mesmos elementos de A, mas em ordem invertida, ou seja, o primeiro elemento de A será o último de B, o segundo elemento de A será o penúltimo de B e assim sucessivamente.

Exemplo:

```

=== VETOR A ===
5 9 8 2 3 2 10 1 7 3

=== VETOR B ===
3 7 1 10 2 3 2 8 9 5

```

Deseja repetir o programa (S ou N)?

5) Gerar vetores A e B, sendo A com  $n$  elementos aleatórios entre 1 e 20 e B com  $n$  elementos aleatórios entre 1 e 15. O usuário informa a quantidade de elementos dos vetores, ou seja, o seu tamanho. Validar para que seja informado um valor positivo para a quantidade de elementos. Mostrar os elementos dos vetores A e

B. Construir um vetor C, em que cada elemento de C será 1 quando o elemento de A for maior que o respectivo elemento em B; será 0 quando o elemento de A for igual ao respectivo elemento em B; e será -1 quando o elemento de A for menor que o respectivo elemento em B. Mostrar os elementos do vetor C.

Exemplos:

```
Informe o tamanho do vetor A: 7
Informe o tamanho do vetor B: 12

=== VETOR A ===
20  18  2   2  20  19   3

=== VETOR B ===
5   3  12  2  15  14  3  13  8  11  6  12

=== VETOR C ===
1   1  -1  0  1  1  0  -1  -1  -1  -1  -1

Deseja repetir o programa (S ou N)?
```

6) Gerar um vetor de inteiros com 10 elementos aleatórios entre 1 e 200. Desse vetor, calcular e mostrar:

- a) A soma de elementos armazenados nesse vetor que são inferiores a 100;
- b) A quantidade de elementos armazenados nesse vetor que tem valor igual a 100;
- c) A média dos elementos armazenados nesse vetor que são superiores a 100. Validar para não realizar uma divisão por zero.

Exemplo:

```
=== VETOR ===
126 100  2 121  68  97  9  86 134  71

Soma dos valores menores do que 100: 333
Quantidade de valores igual a 100: 1
Média dos valores maiores do que 100: 127.00
```

Deseja repetir o programa (S ou N)? s

7) Em uma determinada cidade, sabe-se que, de janeiro a abril de um determinado ano (121 dias), a temperatura ficou entre a 5° e 45°. Gerar um vetor com valores inteiros randômicos nesse intervalo para esse período de tempo. Para isso, crie uma função na biblioteca **Vetores.h** chamada **GerarVetorInteiroComLimite()** que receba como parâmetros: o vetor, o tamanho do vetor, o valor inicial e o valor final do limite da aleatoriedade. A partir do vetor gerado, obtenha e mostre:

- a) A menor temperatura ocorrida.
- b) A maior temperatura ocorrida.
- c) A temperatura média.
- d) O número de dias em que a temperatura foi inferior à temperatura média.

Exemplo:

```

=== VETOR ===
18  7  44  36  17  16  18  41  27  39  41  20  44  17  41  7
28  15  42  41  36  37  45  13  30  45  29  36  39  20  18  35
27  30  30  39  33  38  27  10  5  38  29  43  28  45  20  33
12  23  42  10  14  18  10  13  7  34  8  37  35  34  10  32
45  18  45  11  11  34  19  14  39  5  20  37  30  11  32  19
21  35  22  41  16  36  35  18  10  25  27  5  37  32  11  17
43  45  45  40  6  28  32  45  33  26  39  44  11  40  36  18
37  13  7  42  21  23  23  12  20

```

Menor temperatura ocorrida: 5

Maior temperatura ocorrida: 45

Temperatura media: 26.97

Quantidade de dias em que a temperatura foi menor que a media: 54

Deseja repetir o programa (S ou N)?

8) Gerar aleatoriamente um vetor com 100 elementos que conterá somente valores 0 ou 1. Mostrar esse vetor. Em seguida contar quantos 0s sucessivos o vetor contém, armazenar essa quantidade em outro vetor e mostrá-lo. Qual seria o tamanho mínimo e máximo desse vetor que armazena as quantidades? Criá-lo para o tamanho máximo possível.

Exemplo:

```

=== VETOR A ===
1  1  1  1  1  1  0  1  0  1  1  0  0  0  1  0
0  1  0  1  0  1  1  0  1  0  0  0  0  0  1  1
0  0  1  0  1  1  1  1  0  0  0  0  1  1  0  1
0  1  0  1  0  0  1  0  0  1  1  1  0  0  1  0
1  0  0  0  0  0  1  0  1  1  1  0  1  1  1  1
1  1  1  0  1  1  0  0  1  1  1  0  0  0  1  0
1  0  0  1

```

```

=== VETOR B ===
1  1  3  2  1  1  1  5  2  1  4  1  1  1  2  2  2  1  5  1
1  1  2  3  1  2

```

Deseja repetir o programa (S ou N)? n

9) O que faz o algoritmo a seguir:

```

declare a[50], i, soma como inteiro
declare media como real
soma ← 0

```

```

repetir i = 0, até i < 50, passo 1
    leia a[i]
fim-repetir

```

```

repetir i = 0, até i < 50, passo 1
    soma ← soma + a[i]
fim-repetir

```

```

repetir i = 0, até i < 50, passo 1
    escreva a[i]
fim-repetir

```

```
media ← soma / 50
escreva media
```

Implemente a solução na linguagem C para esse algoritmo. Deve ser utilizada a função para gerar os valores do vetor aleatoriamente e a função para mostrar os elementos do vetor, já implementadas na biblioteca **Vetores.h** e também criar uma função chamada **SomarVetorInteiro()**, que deve ser incluída na biblioteca **Vetores.h** e receberá como parâmetros o vetor e o tamanho do vetor e retornará o valor da soma dos elementos do vetor. Considere que os valores do vetor devem estar compreendidos em um intervalo de 0 a 100, dessa forma, a função **GerarVetorInteiro()** deve ser modificada para aceitar essa faixa de valores.

Exemplo:

```
=== VETOR ===
77  55  52  92   2  88   7  98  32  91  85  53  82  96  60  16
84  91  53  27  65   5 100  43  68  95  76  32  25  39  72  36
23   8  35   2  93  15  49  49  90  22  44   0  32  38  15  77
61  79
```

Media: 52.58

Deseja repetir o programa (S ou N)?

10) O que faz o algoritmo a seguir:

```
declare vet[80], i como inteiro
repetir i = 0, até i < 80, passo 1
    leia vet[i]
fim-repetir
menor ← vet[0]
posicaomenor ← 0

repetir i = 0, até i < 80, passo 1
    se ( vet[i] < menor ) then
        menor ← vet(i)
        posicaomenor ← i
    fim-se
fim-repetir
escreva 'menor = ', menor, 'na posicao =', posicaomenor
```

Implemente a solução na linguagem C para esse algoritmo. Devem ser utilizadas as funções para gerar aleatoriamente os valores do vetor e mostrar os elementos do vetor, que se encontram em **Vetores.h**. Considere que os valores do vetor devem estar compreendidos em um intervalo de 0 a 100.

Exemplo:

```
=== VETOR ===
52  20  60  42  89  50  84  80  13  80  10  71  90  65  64  99
48  41  25  68   2  94  11  92   0  73  55 100  45  73  47  14
97  56  30  12  22  46  82  47  98  29  92  48  21  87  67  17
46 100   3  69  14  46   5   1  95  97 100  21  67  51  64  45
78  83  92  93  14  97  66  42   2  42  12  15  73  59  22  25
```

Menor valor eh 0 e esta na posicao 24

Deseja repetir o programa (S ou N)?

11) O que faz o algoritmo a seguir:

```
declare A[30], B[30], i, k como inteiro
repetir i = 0, até i < 30, passo 1
    leia A[i]
    escrevaA[i]
fim-repetir

k ← 0
repetir i = 0, i < 30, passo 1
    se ( A[i] <> 0 ) then
        b[k] ← a[i]
        k ← k + 1
    fim-se
fim-repetir
repetir i = 0, até i < k, passo 1
    escreva B[i]
fim-repetir
```

Implemente uma solução na linguagem C para esse algoritmo. Para gerar os elementos do vetor A, deve ser utilizada a função para gerar valores aleatórios e a função para mostrar os valores do vetor. Considere que os valores do vetor devem estar compreendidos em um intervalo de 0 a 100.

Exemplo:

```
=== VETOR A ===
49  79  92  82  45  26  30  82  15  83  74  53  77  35  61  86
73  92  0  44  74  29  0  43  76  84  22  59  7  46

=== VETOR B ===
49  79  92  82  45  26  30  82  15  83  74  53  77  35  61  86
73  92  44  74  29  43  76  84  22  59  7  46

Deseja repetir o programa (S ou N)? n
```