



## Lista 2 - Estruturas Homogêneas

### Vetores de caracteres, funções e arquivos de cabeçalho

### Exercícios de sala de aula

#### **Observações:**

- ✓ Vetores precisam ter tamanho definido quando são declarados.
- ✓ Cuidado para não ultrapassar o tamanho de um 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 a biblioteca **time.h** para usar **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 números em uma determinada faixa.
- ✓ A constante **RAND\_MAX** tem valor 32767. As funções **srand()** e **rand()** e a constante **RAND\_MAX** requerem a biblioteca **stdlib.h** e a função **time()** requer a biblioteca **time.h**.

1) Criar uma função para gerar valores aleatórios em um vetor com o seguinte protótipo:

```
void gerarVetorInteiroComFaixa(int vet[], int tam, int limInf, int limSup);
```

Sendo que **limInf** e **limSup** são os limites da faixa de valores sorteados.

**Dica:** Use `vetor[i] = (rand() % ((limSup - limInf) + 1)) + limInf;` para gerar valores aleatórios entre os limites informados.

Usando a função **gerarVetorInteiroComFaixa()**, gerar aleatoriamente um vetor com 20 elementos entre 5 e 20. Em seguida ordenar o vetor. Percorrer o vetor ordenado e mostrar os divisores e a quantidade de divisores de cada um dos valores armazenados. Caso existam elementos repetidos, considerar apenas um deles.

**Atenção:** Neste exercício, além de utilizar a função **gerarVetorInteiroComFaixa()**, usar também a função **mostrarVetorInteiro()**, para exibir os elementos de um vetor, a função **ordenarVetorInteiro()**, para ordenar os

elementos de um vetor, e as funções `mostrarDivisores()` e `qtdeDivisores()`, que criadas em listas de exercícios anteriores e estão armazenadas em uma biblioteca chamada "divisores.h".

Exemplo:

```
==== VETOR ORIGINAL ====
17 11 15 8 13 13 19 19 12 11 14 18 16 8 8 11 13 12 12 16

==== VETOR ORDENADO ====
8 8 8 11 11 11 12 12 12 13 13 13 14 15 16 16 17 18 19 19

==== DIVISORES ====
8 => 1 2 4 8 - 4 divisores
11 => 1 11 - 2 divisores
12 => 1 2 3 4 6 12 - 6 divisores
13 => 1 13 - 2 divisores
14 => 1 2 7 14 - 4 divisores
15 => 1 3 5 15 - 4 divisores
16 => 1 2 4 8 16 - 5 divisores
17 => 1 17 - 2 divisores
18 => 1 2 3 6 9 18 - 6 divisores
19 => 1 19 - 2 divisores

Deseja repetir o programa (S/s)?
```

2) Gerar um vetor com 20 elementos aleatórios entre 5 e 20. Ordenar o vetor. Percorrer o vetor ordenado e mostrar os pares e a soma dos pares de cada um dos valores armazenados. Caso existam elementos repetidos, considerar apenas um deles.

Neste exercício, além de utilizar a função `gerarVetorInteiroComFaixa()`, a função `mostrarVetorInteiro()` e a função `ordenarVetorInteiro()`, criar também função `mostrarPares()`, que recebe um número e mostra todos os números pares positivos até esse número, e função `somarPares()`, que recebe um número e retorna a soma de todos os números pares positivos até esse número. As funções `mostrarPares()` e `somarPares()` devem ser criadas em uma biblioteca chamada "pares.h".

Exemplo:

```
==== VETOR ORIGINAL ====
15 6 16 20 12 5 9 18 11 7 12 15 19 16 11 5 10 19 13 13

==== VETOR ORDENADO ====
5 5 6 7 9 10 11 11 12 12 13 13 15 15 16 16 18 19 19 20

==== PARES ====
5 => 2 4 Soma = 6
6 => 2 4 6 Soma = 12
7 => 2 4 6 Soma = 12
9 => 2 4 6 8 Soma = 20
10 => 2 4 6 8 10 Soma = 30
11 => 2 4 6 8 10 Soma = 30
12 => 2 4 6 8 10 12 Soma = 42
13 => 2 4 6 8 10 12 Soma = 42
15 => 2 4 6 8 10 12 14 Soma = 56
16 => 2 4 6 8 10 12 14 16 Soma = 72
18 => 2 4 6 8 10 12 14 16 18 Soma = 90
19 => 2 4 6 8 10 12 14 16 18 Soma = 90
20 => 2 4 6 8 10 12 14 16 18 20 Soma = 110

Deseja repetir o programa (S/s)?
```

3) Na biblioteca "vetores.h" criar função:

- Com o protótipo `void gerarVetorFloat0a1(float vetor[], int tam)` para gerar valores float aleatórios entre 0 e 1 em um vetor. Use `vetor[i] = (float)(rand()) / RAND_MAX;` para gerar valores aleatórios entre 0 e 1.
- Com o protótipo `void gerarVetorFloat0a100(float vetor[], int tam)` para gerar valores float aleatórios entre 0 e 100 em um vetor. Use `vetor[i] = ((float)(rand()) / RAND_MAX) * 100;` para gerar valores aleatórios entre 0 e 100.
- Com o protótipo `void mostrarVetorFloat(float vetor[], int tam)` para mostrar os elementos de um vetor float.
- Com o protótipo `float somarVetorFloat(float vetor[], int tam)` para somar os elementos de um vetor float.

Usando essas funções:

- Gerar um vetor float com 10 elementos aleatórios entre 0 e 1, exibir o vetor e mostrar a soma de todos os elementos do vetor.
- Gerar um vetor float com 50 elementos aleatórios entre 0 e 100 e exibir o vetor. Criar outros dois vetores numéricos do tipo float, em um deles armazenar os valores menores que 50 e em outro os maiores que 50. Mostrar os vetores gerados.

Exemplo:

```
==== VETOR 1 ====
0.8    0.5    0.5    0.4    0.1    0.1    0.3    0.7    0.0    0.2
Soma: 3.67

==== VETOR 2 ====
76.6    53.3    53.0    44.2    11.0    6.6    31.4    72.5    0.8    17.1    45.2    39.4    43.1    57.2    37.6    83.9    65.6    57.3
25.4    80.9    18.4    94.7    45.6    63.7    16.8    26.5    68.0    50.4    76.6    46.0    38.3    74.4    32.9    90.4    69.6    75.4
81.9    76.3    45.3    51.1    37.1    94.2    51.1    23.9    40.8    36.8    0.9    27.0    20.6    91.4

==== VETOR 3 ====
44.2    11.0    6.6    31.4    0.8    17.1    45.2    39.4    43.1    37.6    25.4    18.4    45.6    16.8    26.5    46.0    38.3    32.9
45.3    37.1    23.9    40.8    36.8    0.9    27.0    20.6

==== VETOR 4 ====
76.6    53.3    53.0    72.5    57.2    83.9    65.6    57.3    80.9    94.7    63.7    68.0    50.4    76.6    74.4    90.4    69.6    75.4
81.9    76.3    51.1    94.2    51.1    91.4

Deseja repetir o programa (S/s)?
```

4) Na biblioteca "vetores.h" criar função:

- Com o protótipo `void gerarVetorCharMinuscula(char vetor[], int tam)` para gerar caracteres alfabéticos minúsculos em um vetor. Use `vetor[i] = rand() % 26 + 97;`

**Explicando:** A função `rand() % 26` gera um número aleatório entre 0 e 25, que somado com 97 equivale a um valor entre 97 a 122, que se refere aos caracteres alfabéticos minúsculos da tabela ASCII.

b) Com o protótipo `void gerarVetorCharMaiuscula(char vetor[], int tam)` para caracteres alfabéticos maiúsculos. Use `vetor[i] = rand() % 26 + 65;`

**Explicando:** A função `rand() % 26` gera um número aleatório entre 0 e 25, que somado com 65 equivale a um valor entre 65 a 90, que se refere aos caracteres alfabéticos maiúsculos da tabela ASCII.

c) Com o protótipo `void mostrarVetorChar(char vetor[], int tam, int n)` para mostrar o vetor gerado em colunas com *n* caracteres por linha, separados por um espaço.

Usando essas funções:

a) Gerar um vetor com 100 caracteres alfabéticos minúsculos.

b) Gerar um vetor com 200 caracteres alfabéticos maiúsculos.

c) Mostrar ambos os vetores gerados em colunas com 10 caracteres por linha, sendo cada caractere deve ser separado por um espaço.

Exemplo:

Quantos caracteres deseja mostrar por linha? 20

=== VETOR DE MINUSCULAS ===

```
k q u v c n c k d q d x g t u y c u h o
m x i o z q y e h i s l y q z j z c d a
p b e k d d p w n c p b x l a h z d c f
r p t h l p b p i m x e h k b i a h l o
h a t g i u b s f q t q p s s v o n l k
```

=== VETOR DE MAIUSCULAS ===

```
K Q U V C N C K D Q D X G T U Y C U H O
M X I O Z Q Y E H I S L Y Q Z J Z C D A
P B E K D D P W N C P B X L A H Z D C F
R P T H L P B P I M X E H K B I A H L O
H A T G I U B S F Q T Q P S S V O N L K
F H I Q F Y E P D R Y K W T V X T R K H
H D F F J H W Z J D G M Y Y C W K J O C
U B D F C Z O A I S P W Z Y H V R F N K
D A Z C N R O B W A J T L N B S B B X D
J D D G G Y U X J S Q I U C N H W I W I
```

Deseja repetir o programa (S/s)?

5) Na biblioteca "vetores.h" criar uma função com o protótipo `void gerarVetorPositivoNegativo(int vetor[], int tam, int limNegativo, int limPositivo)` para gerar números positivos e negativos em um intervalo de valores. Use `vetor[i] = rand() % (limPositivo + limNegativo + 1) - limNegativo;`

Usando essa função:

Gerar um vetor A de inteiros com 100 elementos, com valores entre -50 e +50. Armazenar em um vetor B somente os valores positivos do vetor A.

Exemplo:

```

===== VETOR A =====
30      -4      4      -41     -11     -43     -44     -32     -21     -25     -23
40      10     -10      5      46     -12     -20     -5      48     -3     -41
35     -31     -33     -26     -30     -8     -20      5      34      23     -20
-36     -20      3     -15     -50     22     -26     -44     28     -15     -40
25      35      25      40     -40     -14     42      19      18      48     -20
37      24     -4      49     -7     -50     -4      38      23     -36

===== VETOR B =====
30      4      49      26      40      10      5      46      48      28      48
9       31      13      43      32      3      22      28      16      38      17
42      19      18      48      4      44      37      24      49      38      23

Deseja repetir o programa (S/s)?

```

6) Uma locadora de vídeos armazena em um vetor A de 300 posições a quantidade de filmes retirados por seus clientes durante o ano. A locadora está fazendo uma promoção e para cada 10 filmes retirados, o cliente tem direito a uma locação grátis. Faça um programa que crie um vetor B contendo a quantidade de locações gratuitas a que cada cliente tem direito. Declare ambos os vetores com valores inteiros.

Exemplo:

```

===== VETOR A =====
52      50      4      94      13      81      7      78      1      36      21      35      91      51      62      31      96      96
87      66      48      94      6      66      90      9      5      55      52      26      5      84      95      47      44      74
34      18      40      90      36      31      5      12      70      68      89      76      77      96      31      86      89      94
14      66      75      71      58      42      93      2      59      30      88      36      49      52      1      100      48      84
18      27      46      78      90      91      21      63      23      52      17      60      64      15      65      76      84      87
8       40      30      29      15      46      26      17      49      3      70      13      64      45      56      60      58      67
63      72      98      97      79      59      76      43      6      96      98      18      53      72      29      47      96      31
60      54      33      84      19      97      10      88      58      56      49      27      46      40      76      39      60      89
77      77      30      77      15      25      79      10      77      70      47      81      98      74      49      15      86      27
83      80      25      30      95      79      18      19      91      37      25      65      76      14      40      27      74      20
66      99      43      45      73      22      36      43      21      40      36      93      57      29      80      57      94      97
12      36      42      24      92      22      11      69      84      87      84      20      91      89      43      25      54      50
80      95      54      24      53      84      76      20      70      35      68      20      38      48      3      24      94      23
60      47      46      47      54      61      46      100      79      84      71      91      12      91      66      17      35      34
89      75      85      63      75      20      86      45      84      63      1      11      98      59      29      43      45      40
52      88      58      39      17      56      46      93      45      34      53      45      37      35      33      28      83      32
83      48      86      95      33      27      71      97      73      36      10      3

===== VETOR B =====
5       5       0       9       1       8       0       7       0       3       2       3       9       5       6       3       9       9
8       6       4       9       0       6       9       0       0       5       5       2       0       8       9       4       4       7
3       1       4       9       3       3       0       1       7       6       8       7       7       9       3       8       8       9
1       6       7       7       5       4       9       0       5       3       8       3       4       5       0       10      4       8
1       2       4       7       9       9       2       6       2       5       1       6       6       1       6       7       8       8
0       4       3       2       1       4       2       1       4       0       7       1       6       4       5       6       5       6
6       7       9       9       7       5       7       4       0       9       9       1       5       7       2       4       9       3
6       5       3       8       1       9       1       8       5       5       4       2       4       4       7       3       6       8
7       7       3       7       1       2       7       1       7       7       4       8       9       7       4       1       8       2
8       8       2       3       9       7       1       1       9       3       2       6       7       1       4       2       7       2
6       9       4       4       7       2       3       4       2       4       3       9       5       2       8       5       9       9
1       3       4       2       9       2       1       6       8       8       8       2       9       8       4       2       5       5
8       9       5       2       5       8       7       2       7       3       6       2       3       4       0       2       9       2
6       4       4       4       5       6       4       10      7       8       7       9       1       9       6       1       3       3
8       7       8       6       7       2       8       4       8       6       0       1       9       5       2       4       4       4
5       8       5       3       1       5       4       9       4       3       5       4       3       3       3       2       8       3
8       4       8       9       3       2       7       9       7       3       1       0

```

Deseja repetir o programa (S/s)?