

- 9.1 Implemente um algoritmo que receba números inteiros digitados pelo usuário e os armazene em vetores paralelos, na ordem em que foram digitados. Inicialmente, o usuário deverá informar a quantidade de elementos da sequência, em valor limitado a 30. Ao final, o programa deverá imprimir a sequência, em ordem decrescente, sendo um valor em cada linha.

```

1  Algoritmo "Lista 09_01 – Ordenação Decrescente"
2  Var
3      numeros: vetor[1..30] de inteiro
4      qtd, i, j, aux: inteiro
5
6  Inicio
7      escreva("Informe a quantidade de elementos (máximo 30): ")
8      leia(qtd)
9
10     // Entrada de dados no vetor
11     para i de 1 ate qtd faca
12         escreva("Digite o ", i, "º número: ")
13         leia(numeros[i])
14     fimpara
15
16     // Algoritmo de ordenação (Bubble Sort) em ordem decrescente
17     para i de 1 ate qtd - 1 faca
18         para j de i + 1 ate qtd faca
19             se numeros[i] < numeros[j] entao
20                 aux <- numeros[i]
21                 numeros[i] <- numeros[j]
22                 numeros[j] <- aux
23         fimse
24     fimpara
25 fimpara
26
27 // Saída de dados: um valor em cada linha
28 escreval(" ")
29 escreval("Sequência em ordem decrescente:")
30 para i de 1 ate qtd faca
31     escreval(numeros[i])
32 fimpara
33 Fimalgoritmo

```

```

Informe a quantidade de elementos
Digite o 1º número: 4
Digite o 2º número: 8
Digite o 3º número: 2
Digite o 4º número: 11
Digite o 5º número: 8
Digite o 6º número: 7
Digite o 7º número: 12
Digite o 8º número: 9
Digite o 9º número: 3
Digite o 10º número: 10

Sequência em ordem decrescente:
12
11
10
9
8
8
7
4
3
2

>>> Fim da execução do programa !

```

9.2 Implemente um programa que receba, via teclado, os dados de até 30 alunos (número, nome, pelotão e telefone) e armazene-os em um vetor. Ao final do cadastro dos dados de cada aluno, o programa deve perguntar se deseja cadastrar outro aluno ou não. Ao responder sim, o programa deve permitir o acréscimo de mais um aluno. Se o usuário responder não, o programa deverá exibir os dados dos alunos cadastrados, conforme discriminado no exemplo:

```
--- Cadastro do Aluno 1
Digite o Número do aluno: 4550
Digite o Nome do aluno: João Pedro
Digite o Pelotão: 1º
Digite o Telefone: 21 988556677
Deseja cadastrar outro aluno? (S/N): S
-----
```

```
--- Cadastro do Aluno 2
Digite o Número do aluno: 3339
Digite o Nome do aluno: Ana Lúcia
Digite o Pelotão: 3º
Digite o Telefone: 11 234567666
-----
```

```
Deseja cadastrar outro aluno? (S/N): N
=====
```

RELATÓRIO DE ALUNOS

```
=====
NUM. | NOME      | PEL  | TELEFONE
-----
4550 | João Pedro | 1º   | 21 988556677
3339 | Ana Lúcia  | 3º   | 11 234567666
9999 | Carlos     | 8º   | 11 23456789
```

```
1 algoritmo "Lista 09_02 - Cadastro_Alunos_Vetor"
2 var
3   // Definição dos vetores para até 30 alunos
4   v_numero: vetor [1..30] de inteiro
5   v_nome: vetor [1..30] de caractere
6   v_pelotao: vetor [1..30] de caractere
7   v_telefone: vetor [1..30] de caractere
8
9   // Variáveis de controle
10  i, totalAlunos: inteiro
11  continuar: caractere
12
13  inicio
14    totalAlunos <- 0
15    continuar <- "S"
16
17    // Laço para cadastro (máximo 30 ou até usuário responder 'N')
18    enquanto (continuar = "S") e (totalAlunos < 30) faça
19      totalAlunos <- totalAlunos + 1
20
21    escreval("--- Cadastro do Aluno ", totalAlunos)
22    escreva("Digite o Número do aluno: ")
23    leia(v_numero[totalAlunos])
24    escreva("Digite o Nome do aluno: ")
25    leia(v_nome[totalAlunos])
26    escreva("Digite o Pelotão: ")
27    leia(v_pelotao[totalAlunos])
28    escreva("Digite o Telefone: ")
29    leia(v_telefone[totalAlunos])
```

```

30
31 escreval("-----")
32 escreva("Deseja cadastrar outro aluno? (S/N): ")
33 leia(continuar)
34 continuar <- maiusc(continuar)
35 fimenquanto
36
37 // Exibição do Relatório Final
38 escreval("")
39 escreval("=====")
40 escreval("          RELATÓRIO DE ALUNOS")
41 escreval("=====")
42 escreval("NUM. | NOME          | PEL | TELEFONE")
43 escreval("-----")
44
45 para i de 1 ate totalAlunos faca
46     escreval(v_numero[i], " | ", v_nome[i]:17, " | ", v_pelotao[i]:4, " | ", v_telefone[i])
47 fimpara
48 fimalgoritmo

```

```

--- Cadastro do Aluno 1
Digite o Número do aluno: 1000
Digite o Nome do aluno: Pedro César
Digite o Pelotão: 7
Digite o Telefone: 11 22334455
-----
Deseja cadastrar outro aluno? (S/N): S
--- Cadastro do Aluno 2
Digite o Número do aluno: 2000
Digite o Nome do aluno: Ana
Digite o Pelotão: 3
Digite o Telefone: 12 343434344
-----
Deseja cadastrar outro aluno? (S/N): S
--- Cadastro do Aluno 3
Digite o Número do aluno: 3333
Digite o Nome do aluno: Carlos
Digite o Pelotão: 8
Digite o Telefone: 11 3434343434
-----
Deseja cadastrar outro aluno? (S/N): N
-----

```

RELATÓRIO DE ALUNOS			
NUM.	NOME	PEL	TELEFONE
1000	Pedro César	7	11 22334455
2000	Ana	3	12 343434344
3333	Carlos	8	11 3434343434

```

>>> Fim da execução.

```

- 9.3 Escreva um algoritmo que crie e imprima uma matriz de inteiros, de 4 x 4 elementos. Permitir a entrada dos valores numéricos desta matriz e exibir a impressão da matriz de duas maneiras: 1ª) de forma contínua, na ordem em que foi informada; 2ª) no formato organizado (modelo convencional de matrizes).

```
1  algoritmo "Lista 09_03 – Matriz 4x4"
2  var
3    m: vetor [1..4, 1..4] de inteiro
4    l, c: inteiro // l: linha; c: coluna
5
6  inicio
7    // Passo 1: Entrada de dados
8    escreval("Preencha a matriz 4x4:")
9    para l de 1 ate 4 faca
10     para c de 1 ate 4 faca
11      escreva("Digite o valor para [", l, ",", c, "]: ")
12      leia(m[l,c])
13    fimpara
14  fimpara
15
16  escreval("") // Pula linha para organizar a saída
17
18  // Passo 2: Impressão Contínua
19  escreval("1a) Impressão Contínua (Ordem de digitação):")
20  para l de 1 ate 4 faca
21    para c de 1 ate 4 faca
22      escreva(m[l,c], " ")
23    fimpara
24  fimpara
25
26  escreval("")
27  escreval("")
28
29  // Passo 3: Impressão no Formato Organizado (Matriz)
30  escreval("2a) Impressão no Formato Convencional:")
31  para l de 1 ate 4 faca
32    para c de 1 ate 4 faca
33      // O ":4" reserva 4 espaços para alinhar os números
34      escreva(m[l,c]:4)
35    fimpara
36    escreval("") // Após terminar as colunas de uma linha, pula para a linha de baixo
37  fimpara
38  fimAlgoritmo
```

Exemplo de saída do programa:

```
Preencha a matriz 4x4:
Digite o valor para [ 1, 1]: 1
Digite o valor para [ 1, 2]: 2
Digite o valor para [ 1, 3]: 3
Digite o valor para [ 1, 4]: 4
Digite o valor para [ 2, 1]: 10
Digite o valor para [ 2, 2]: 11
Digite o valor para [ 2, 3]: 12
Digite o valor para [ 2, 4]: 13
Digite o valor para [ 3, 1]: 21
Digite o valor para [ 3, 2]: 22
Digite o valor para [ 3, 3]: 23
Digite o valor para [ 3, 4]: 24
Digite o valor para [ 4, 1]: 55
Digite o valor para [ 4, 2]: 56
Digite o valor para [ 4, 3]: 57
Digite o valor para [ 4, 4]: 58

1a) Impressão Contínua (Ordem de digitação):
 1  2  3  4 10 11 12 13 21 22 23 24 55 56 57 58

2a) Impressão no Formato Convencional:
 1   2   3   4
10  11  12  13
21  22  23  24
55  56  57  58

>>> Fim da execução do programa !
```

- 9.4 Faça um pseudocódigo que recebe a idade de uma população (n=12), e calcula e apresenta o valor da média e do desvio padrão dessa população de elementos. *Como estamos no capítulo que trata de vetores e matrizes, um destes deverá constar da solução.*

A fórmula para o cálculo do desvio padrão de uma população é:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}}$$

onde,

x_i = elemento i da população

μ = média

N = quantidade de elementos

△ Para resolver este problema, precisamos dividir o cálculo em duas etapas principais após a coleta dos dados: primeiro calculamos a média (μ) e, em seguida, utilizamos essa média para calcular o desvio padrão (σ) conforme a fórmula fornecida. No Visualg, utilizaremos a função `raizq` para a raiz quadrada e o operador `^` para a potenciação. Este último também pode ser utilizado para cálculo da raiz, como exemplo a raiz quadrada de x seria $x^{(1/2)}$.

```
1 algoritmo "Lista 09_04 – Cálculo da Média e Desvio Padrão"
2 var
3   idades: vetor [1..12] de inteiro
4   i: inteiro
5   somaldades, media, somaDesvios, desvioPadrao: real
6
7 inicio
8   somaldades <- 0
9   somaDesvios <- 0
10
```

```

11 // Passo 1: Receber as 12 idades e calcular a soma total
12 escreval("--- Entrada de Dados (12 idades) ---")
13 para i de 1 ate 12 faca
14     escreva("Digite a idade da pessoa ", i, ": ")
15     leia(idades[i])
16     somaldades <- somaldades + idades[i]
17 fimpara
18
19 // Passo 2: Calcular a Média (mu)
20 media <- somaldades / 12
21
22 // Passo 3: Calcular o Somatório de (xi - media)^2
23 para i de 1 ate 12 faca
24     somaDesvios <- somaDesvios + (idades[i] - media)^2
25 fimpara
26
27 // Passo 4: Calcular o Desvio Padrão Populacional (sigma)
28 desvioPadrao <- raizq(somaDesvios / 12)
29
30 // Passo 5: Exibir resultados
31 escreval("-----")
32 escreval("Média da população: ", media:4:2)
33 escreval("Desvio Padrão: ", desvioPadrao:4:4)
34 escreval("-----")
35 fimalgoritmo

```

```

--- Entrada de Dados (12 idades) ---
Digite a idade da pessoa 1: 33
Digite a idade da pessoa 2: 22
Digite a idade da pessoa 3: 11
Digite a idade da pessoa 4: 88
Digite a idade da pessoa 5: 45
Digite a idade da pessoa 6: 53
Digite a idade da pessoa 7: 22
Digite a idade da pessoa 8: 21
Digite a idade da pessoa 9: 22
Digite a idade da pessoa 10: 26
Digite a idade da pessoa 11: 15
Digite a idade da pessoa 12: 18
-----
Média da população: 31.33
Desvio Padrão: 20.6653
-----
>>> Fim da execução do programa !

```

- 9.5 Utilizando os conceitos de procedimentos, funções e vetores, construa um algoritmo que determine, para uma série de 10 valores inteiros, digitados pelo usuário, quantos são maiores que a média do conjunto. O programa deverá fornecer como saída: a série digitada, a média da série e a quantidade de valores acima da média.

```
1  algoritmo "Lista 09_05 - Operação sobre vetores"
2  var
3      vetorA: vetor [1..10] de inteiro
4      media: real
5      acimaMedia, indice: inteiro // Variável para armazenar o retorno da função de contagem
6
7  funcao calcularMedia(): real // Função 1: Retorna a Média
8  var
9      soma, j: inteiro
10 inicio
11     soma <- 0
12     para j de 1 ate 10 faca
13         soma <- soma + vetorA[j]
14     fimpara
15     retorne soma / 10
16 fimfuncao
17
18 funcao contarAcima(): inteiro // Função 2: Retorna a Quantidade Acima da Média
19 var
20     j, contador: inteiro
21 inicio
22     contador <- 0
23     para j de 1 ate 10 faca
24         se (vetorA[j] > media) entao
25             contador <- contador + 1
26         fimse
27     fimpara
28     retorne contador
29 fimfuncao
30
31 // Bloco Principal
32 inicio
33     para indice de 1 ate 10 faca
34         escreva("Digite o ", indice, "º valor: ")
35         leia(vetorA[indice])
36     fimpara
37
38     media <- calcularMedia()
39     // Chamamos a função e atribuímos o resultado à variável 'res'
40     acimaMedia <- contarAcima()
41
42     escreval("-----")
43     escreval("Média do conjunto: ", media)
44     escreval("Quantidade acima da média: ", acimaMedia)
45     escreval("-----")
46 fimAlgoritmo
```

```
Digite o 1º valor: 1
Digite o 2º valor: 2
Digite o 3º valor: 3
Digite o 4º valor: 4
Digite o 5º valor: 5
Digite o 6º valor: 6
Digite o 7º valor: 7
Digite o 8º valor: 8
Digite o 9º valor: 9
Digite o 10º valor: 10
-----
Média do conjunto: 5.5
Quantidade acima da média: 5
-----
>>> Fim da execução do programa !
```

9.6 Na álgebra linear, a multiplicação de um escalar k por uma matriz A resulta em uma nova matriz B , onde cada elemento $b_{ij} = k \times a_{ij}$. Desenvolva um pseudocódigo que:

- Leia uma matriz $A[3][3]$.
- Leia um valor multiplicador k .
- Multiplique a matriz A pelo valor de k e armazene o resultado na matriz B .
- Apresente a matriz original e a matriz multiplicada.

```
1  algoritmo "Lista 09_06 - Multiplicação Escalar de Matriz"
2  var
3      matA: vetor [1..3, 1..3] de real
4      matB: vetor [1..3, 1..3] de real
5      k: real
6      l, c: inteiro
7
8  inicio
9      // 1. Leitura da Matriz A
10     escreval("Preencha a matriz A [3x3]:")
11     para l de 1 ate 3 faca
12         para c de 1 ate 3 faca
13             escreva("Elemento [", l, ", ", c, "]: ")
14             leia(matA[l,c])
15         fimpara
16     fimpara
17
18     // 2. Leitura do Escalar k
19     escreval("")
20     escreva("Digite o valor do multiplicador (k): ")
21     leia(k)
22
23     // 3. Processamento: B = k * A
24     para l de 1 ate 3 faca
25         para c de 1 ate 3 faca
26             matB[l,c] <- k * matA[l,c]
27         fimpara
28     fimpara
29
30     // 4. Apresentação dos Resultados
31     escreval("")
32     escreval("--- MATRIZ ORIGINAL (A) ---")
33     para l de 1 ate 3 faca
34         para c de 1 ate 3 faca
35             escreva(matA[l,c]:6:1, " ")
36         fimpara
37     escreval("")
38     fimpara
39
40     escreval("")
41     escreval("--- MATRIZ RESULTANTE ---")
42     para l de 1 ate 3 faca
43         para c de 1 ate 3 faca
```

```
44     escreva(matB[l,c]:6:1, " ")
45     fimpara
46     escreval("")
47     fimpara
48     fimalgoritmo
```

```
Preencha a matriz A [3x3]:
Elemento [ 1, 1]: 1
Elemento [ 1, 2]: 2
Elemento [ 1, 3]: 3
Elemento [ 2, 1]: 4
Elemento [ 2, 2]: 5
Elemento [ 2, 3]: 6
Elemento [ 3, 1]: 7
Elemento [ 3, 2]: 8
Elemento [ 3, 3]: 9

Digite o valor do multiplicador (k): 10

--- MATRIZ ORIGINAL (A) ---
  1.0   2.0   3.0
  4.0   5.0   6.0
  7.0   8.0   9.0

--- MATRIZ RESULTANTE (B = 10 * A) ---
 10.0  20.0  30.0
 40.0  50.0  60.0
 70.0  80.0  90.0

>>> Fim da execução do programa !
```

```
1  Algoritmo "Extra 09_01 Busca número versão 1"
2  Var
3      numeros: vetor[1..10] de inteiro
4      numero_buscado, i: inteiro
5      encontrado: logico
6  Inicio
7      // Entrada dos 10 números iniciais
8      escreva("--- Entrada de Dados ---")
9      para i de 1 ate 10 faca
10         escreva("Digite o ", i, "º número: ")
11         leia(numeros[i])
12     fimpara
13
14     // Entrada do número que será verificado
15     escreva("Digite o número que deseja buscar: ")
16     leia(numero_buscado)
17
18     // Verificação (Busca Linear)
19     encontrado <- falso
20     para i de 1 ate 10 faca
21         se numeros[i] = numero_buscado entao
22             encontrado <- verdadeiro
23             interrompa // Interrompe o laço assim que encontrar, otimizando o código
24         fimse
25     fimpara
26
27     // Saída do resultado
28     se encontrado = verdadeiro entao
29         escreva("Resultado: O número ", numero_buscado, " ESTÁ contido na lista.")
30     senao
31         escreva("Resultado: O número ", numero_buscado, " NÃO está contido na lista.")
32     fimse
33
34  Fimalgoritmo
```

```
--- Entrada de Dados ---
Digite o 1º número: 1
Digite o 2º número: 2
Digite o 3º número: 3
Digite o 4º número: 4
Digite o 5º número: 5
Digite o 6º número: 6
Digite o 7º número: 7
Digite o 8º número: 8
Digite o 9º número: 9
Digite o 10º número: 10
Digite o número que deseja buscar: 6
Resultado: O número 6 ESTÁ contido na lista.

>>> Fim da execução do programa !
```

```

1  Algoritmo "Extra 09_02 Busca número versão 2 "
2  Var
3      numeros: vetor[1..10] de inteiro
4      numero_buscado, i, posicao: inteiro
5      encontrado: logico
6  Inicio
7      // Entrada dos 10 números iniciais
8      escreval("--- Entrada de Dados ---")
9      para i de 1 ate 10 faca
10         escreva("Digite o ", i, "º número: ")
11         leia(numeros[i])
12     fimpara
13
14     // Entrada do número que será verificado
15     escreva("Digite o número que deseja buscar: ")
16     leia(numero_buscado)
17
18     // Verificação (Busca Linear)
19     encontrado <- falso
20     para i de 1 ate 10 faca
21         se numeros[i] = numero_buscado entao
22             encontrado <- verdadeiro
23             posicao <- i
24             interrompa // Interrompe o laço assim que encontrar, otimizando o código
25         fimse
26     fimpara
27
28     // Saída do resultado
29     se encontrado = verdadeiro entao
30         escreval("O número ", numero_buscado, " ESTÁ contido na lista (posição:", posicao)
31     senao
32         escreval("O número ", numero_buscado, " NÃO está contido na lista.")
33     fimse
34  Fimalgoritmo

```

```

--- Entrada de Dados ---
Digite o 1º número: 1
Digite o 2º número: 2
Digite o 3º número: 3
Digite o 4º número: 4
Digite o 5º número: 5
Digite o 6º número: 6
Digite o 7º número: 7
Digite o 8º número: 8
Digite o 9º número: 9
Digite o 10º número: 10
Digite o número que deseja buscar: 6
Resultado: O número 6 ESTÁ contido na lista.

>>> Fim da execução do programa !

```