

Professor Pietro Martins de Oliveira

# AL GO RIT MOS

do início ao fim

***PACOTE DE EXERCÍCIOS 5:  
ESTRUTURAS DE DADOS***

- 1) Desenvolva um algoritmo que preencha um vetor numérico de 10 posições. Ao final, o algoritmo deve mostrar o somatório de todos os elementos do vetor, bem como a média aritmética entre todos os termos.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

#### Exemplo de execução – Exercício 1 - Caso de teste

```
Insira o dado da posição 1:
1
Insira o dado da posição 2:
2
Insira o dado da posição 3:
3
Insira o dado da posição 4:
4
Insira o dado da posição 5:
5
Insira o dado da posição 6:
6
Insira o dado da posição 7:
7
Insira o dado da posição 8:
8
Insira o dado da posição 9:
9
Insira o dado da posição 10:
10
Somatório: 55
Média: 5.5
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### VETORES – Exercício 1 – Solução

```
01 Algoritmo "processaVetor"
02 Var vet: Vetor[1..10] de real
03     soma, media: real
04     i: inteiro
05 Inicio
06     soma <- 0
07     para i de 1 ate 10 passo 1 faca
08         escreval("Insira o dado da posição ", i, ":")
09         leia(vet[i])
10         soma <- soma + vet[i]
11     fimpara
12     media <- soma / 10
13     escreval("Somatório: ", soma)
14     escreval("Média: ", media)
```

## 15 Fimalgoritmo

- 2) Desenvolva um algoritmo que preencha um vetor de 5 posições com valores reais compreendidos entre 1 e 100 (suponha que o usuário irá respeitar o enunciado). Ao final, o algoritmo deve mostrar, na tela, o conteúdo de cada posição do vetor, dividido por 100.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

### Exemplo de execução – Exercício 2 - Caso de teste

```
Insira o dado da posição 1:
10
Insira o dado da posição 2:
20
Insira o dado da posição 3:
30
Insira o dado da posição 4:
40
Insira o dado da posição 5:
50
Conteúdo dividido por 100:
0.1
0.2
0.3
0.4
0.5
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

### VETORES – Exercício 2 – Solução

```
01 Algoritmo "cincoPosicoes"
02   Var vet: Vetor[1..5] de real
03   i: inteiro
04   Inicio
05     para i de 1 ate 5 passo 1 faca
06       escreval("Insira o dado da posição ", i, ":")
07       leia(vet[i])
08       vet[i] <- vet[i] / 100
09     fimpara
10     escreval("Conteúdo dividido por 100:")
11     para i de 1 ate 5 passo 1 faca
12       escreval(vet[i])
13     fimpara
```

14 Fimalgoritmo

- 3) Desenvolva um algoritmo que preencha um vetor numérico de 10 posições. Após preencher todo o vetor, o usuário deve inserir uma chave de busca X. Caso exista algum número igual a X, dentro do vetor, o algoritmo deve mostrar, na tela, o índice da primeira posição na qual X foi encontrado. Caso contrário, o algoritmo deve se encerrar com uma única mensagem, dizendo "Chave não encontrada".

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – *escreva* ou *escreval*). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – *leia*).

Exemplo de execução – Exercício 3 - Caso de teste 1

```
Insira o dado da posição 1:
11
Insira o dado da posição 2:
12
Insira o dado da posição 3:
13
Insira o dado da posição 4:
14
Insira o dado da posição 5:
15
Insira o dado da posição 6:
16
Insira o dado da posição 7:
17
Insira o dado da posição 8:
18
Insira o dado da posição 9:
19
Insira o dado da posição 10:
20
Insira a chave de busca:
15
Chave encontrada na posição: 5
```

Exemplo de execução – Exercício 4 - Caso de teste 2

```
Insira o dado da posição 1:
11
Insira o dado da posição 2:
12
Insira o dado da posição 3:
13
```

```
Insira o dado da posição 4:
14
Insira o dado da posição 5:
15
Insira o dado da posição 6:
16
Insira o dado da posição 7:
17
Insira o dado da posição 8:
18
Insira o dado da posição 9:
19
Insira o dado da posição 10:
20
Insira a chave de busca:
100
Chave não encontrada.
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### VETORES - Exercício 3 - Solução

```
01 Algoritmo "buscaSequencial"
02   Var X: real
03   vet: Vetor[1..10] de real
04   i, p: inteiro
05   achou: logico
06 Inicio
07   para i de 1 ate 10 passo 1 faca
08     escreval("Insira o dado da posição ", i, ":")
09     leia(vet[i])
10   fimpara
11   escreval("Insira a chave de busca:")
12   leia(X)
13   achou <- falso
14   i <- 1
15   enquanto (i <= 10) e (achou = falso) faca
16     se (vet[i] = X) entao
17       achou <- verdadeiro
18       p <- i
19     fimse
20     i <- i + 1
21   fimenquanto
22   se (achou <> falso) entao
23     escreval("Chave encontrada na posição: ", p)
24   senao
25     escreval("Chave não encontrada.")
26   fimse
27 Fimalgoritmo
```



- 4) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que carregue um vetor e nove elementos numéricos inteiros, calcule e mostre os números primos e suas respectivas posições.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

#### Exemplo de execução – Exercício 5 - Caso de teste

```
Insira o dado da posição 1:
1
Insira o dado da posição 2:
3
Insira o dado da posição 3:
7
Insira o dado da posição 4:
9
Insira o dado da posição 5:
11
Insira o dado da posição 6:
13
Insira o dado da posição 7:
17
Insira o dado da posição 8:
19
Insira o dado da posição 9:
21
3 é primo, posição: 2
7 é primo, posição: 3
11 é primo, posição: 5
13 é primo, posição: 6
17 é primo, posição: 7
19 é primo, posição: 8
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### VETORES – Exercício 4 – Solução

```
01 Algoritmo "vetorDePrimos"
02   Var N, i, j, cont: inteiro
03   vet: Vetor[1..9] de inteiro
04 Inicio
05   para i de 1 ate 9 passo 1 faca
06     escreval("Insira o dado da posição ", i, ":")
07     leia(vet[i])
08   fimpara
09   para i de 1 ate 9 passo 1 faca
```

```
10      cont <- 0
11      para j de 1 ate vet[i] passo 1 faca
12          se (vet[i] % j = 0) entao
13              cont <- cont + 1
14          fimse
15      fimpara
16      se (cont = 2) entao
17          escreval(vet[i], " é primo, posição: ", i)
18      fimse
19  fimpara
20 Fimalgoritmo
```

- 5) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que carregue dois vetores de dez elementos numéricos cada um e mostre um vetor resultante da intercalação desses dois vetores.

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – *escreva* ou *escreval*). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – *leia*).

#### Exemplo de execução – Exercício 5 – Caso de teste 1

```
Insira o dado vet1[ 1]:
10
Insira o dado vet1[ 2]:
20
Insira o dado vet1[ 3]:
30
Insira o dado vet1[ 4]:
40
Insira o dado vet1[ 5]:
50
Insira o dado vet1[ 6]:
60
Insira o dado vet1[ 7]:
70
Insira o dado vet1[ 8]:
80
Insira o dado vet1[ 9]:
90
Insira o dado vet1[ 10]:
100
Insira o dado vet2[ 1]:
11
Insira o dado vet2[ 2]:
12
Insira o dado vet2[ 3]:
13
Insira o dado vet2[ 4]:
14
```

```
Insira o dado vet2[ 5]:  
15  
Insira o dado vet2[ 6]:  
16  
Insira o dado vet2[ 7]:  
17  
Insira o dado vet2[ 8]:  
18  
Insira o dado vet2[ 9]:  
19  
Insira o dado vet2[ 10]:  
20  
10 11 20 12 30 13 40 14 50 15 60 16 70 17 80 18 90  
19 100 20
```

#### Exemplo de execução – Exercício 5 - Caso de teste 2

```
Insira o dado vet1[ 1]:  
11  
Insira o dado vet1[ 2]:  
12  
Insira o dado vet1[ 3]:  
13  
Insira o dado vet1[ 4]:  
14  
Insira o dado vet1[ 5]:  
15  
Insira o dado vet1[ 6]:  
16  
Insira o dado vet1[ 7]:  
17  
Insira o dado vet1[ 8]:  
18  
Insira o dado vet1[ 9]:  
19  
Insira o dado vet1[ 10]:  
20  
Insira o dado vet2[ 1]:  
21  
Insira o dado vet2[ 2]:  
22  
Insira o dado vet2[ 3]:  
23  
Insira o dado vet2[ 4]:  
24  
Insira o dado vet2[ 5]:  
25  
Insira o dado vet2[ 6]:  
26  
Insira o dado vet2[ 7]:  
27  
Insira o dado vet2[ 8]:  
28  
Insira o dado vet2[ 9]:
```



```
29
Insira o dado vet2[ 10]:
30
Vetor resultante:
 11  21  12  22  13  23  14  24  15  25  16  26  17  27  18  28  19
29  20  30
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### VETORES – Exercício 5 – Solução

```
01 Algoritmo "intercalaVetores"
02   Var vet1, vet2: Vetor[1..10] de inteiro
03       inter: Vetor[1..20] de real
04       i, j: inteiro
05 Inicio
06   para i de 1 ate 10 passo 1 faca
07       escreval("Insira o dado vet1[", i, "]:")
08       leia(vet1[i])
09   fimpara
10   para i de 1 ate 10 passo 1 faca
11       escreval("Insira o dado vet2[", i, "]:")
12       leia(vet2[i])
13   fimpara
14   j <- 1
15   para i de 1 ate 10 passo 1 faca
16       inter[j] <- vet1[i]
17       inter[j+1] <- vet2[i]
18       j <- j + 2
19   fimpara
20   para i de 1 ate 20 passo 1 faca
21       escreva(inter[i], " ")
22   fimpara
23 Fimalgoritmo
```

- 6) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que carregue um vetor com oito números inteiros, calcule e mostre dois vetores resultantes. O primeiro vetor resultante deve conter os números positivos. O segundo vetor resultante deve conter os números negativos. Cada vetor resultante vai ter no máximo oito posições, sendo que nem todas devem obrigatoriamente ser utilizadas. Imprima o conteúdo dos vetores resultantes, sem que sejam impressos "lixos de memória".

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

#### Exemplo de execução – Exercício 6 - Caso de teste

```
Insira o dado vet1[ 1]:  
1  
Insira o dado vet1[ 2]:  
-1  
Insira o dado vet1[ 3]:  
2  
Insira o dado vet1[ 4]:  
-2  
Insira o dado vet1[ 5]:  
3  
Insira o dado vet1[ 6]:  
-3  
Insira o dado vet1[ 7]:  
4  
Insira o dado vet1[ 8]:  
-4  
Vetor de positivos:  
1 2 3 4  
Vetor de negativos:  
-1 -2 -3 -4
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### VETORES – Exercício 6 – Solução

```
01 Algoritmo "positivosNegativos"  
02   Var vet, po, ne: Vetor[1..8] de inteiro  
03   i, j, k: inteiro  
04 Inicio  
05   j <- 0  
06   k <- 0  
07   para i de 1 ate 8 passo 1 faca  
08     escreval("Insira o dado vet1[", i, "]:")  
09     leia(vet[i])  
10     se (vet[i] > 0) entao  
11       j <- j + 1  
12       po[j] <- vet[i]  
13     fimse  
14     se (vet[i] < 0) entao  
15       k <- k + 1  
16       ne[k] <- vet[i]  
17     fimse  
18   fimpara  
19   escreval("Vetor de positivos:")  
20   para i de 1 ate j passo 1 faca  
21     escreva(po[i], " ")  
22   fimpara  
23   escreval()  
24   escreval("Vetor de negativos:")  
25   para i de 1 ate k passo 1 faca  
26     escreva(ne[i], " ")
```

```
27     fimpara
28 Fimalgoritmo
```

- 7) Desenvolva um algoritmo que preencha cada elemento de uma matriz 3x3 com o quadrado do valor do índice da linha mais o valor do índice da coluna daquela posição. Ao final, o algoritmo deve mostrar a matriz, na tela.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

#### Exemplo de execução – Exercício 7 – Caso de teste

```
2  3  4
5  6  7
10 11 1
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### MATRIZES – Exercício 7 – Solução

```
01 Algoritmo "populaMatriz"
02     Var i, j: inteiro
03     mat: Vetor[1..3,1..3] de inteiro
04 Inicio
05     para i de 1 ate 3 passo 1 faca
06         para j de 1 ate 3 passo 1 faca
07             mat[i,j] <- quad(i) + j
08             escreva(mat[i,j], " ")
09         fimpara
10     escreval()
11 fimpara
12 Fimalgoritmo
```

- 8) Desenvolva um algoritmo que preencha uma matriz numérica de dimensões 3x3. Depois de a matriz ter sido populada, o algoritmo deverá imprimir a matriz da seguinte forma: os dados da diagonal principal devem ser impressos normalmente e os dados fora da diagonal principal devem substituídos por zero.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens

geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia)

#### Exemplo de execução – Exercício 8 - Caso de teste

```
Insira o dado da posição 1, 1:
1
Insira o dado da posição 1, 2:
2
Insira o dado da posição 1, 3:
3
Insira o dado da posição 2, 1:
4
Insira o dado da posição 2, 2:
5
Insira o dado da posição 2, 3:
6
Insira o dado da posição 3, 1:
7
Insira o dado da posição 3, 2:
8
Insira o dado da posição 3, 3:
9
1 0 0
0 5 0
0 0 9
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segure aí uma solução que funciona:

#### MATRIZES – Exercício 8 – Solução

```
01 Algoritmo "diagonalPrincipal"
02   Var mat: Vetor[1..3,1..3] de real
03   i, j: inteiro
04   Inicio
05     para i de 1 ate 3 passo 1 faca
06       para j de 1 ate 3 passo 1 faca
07         escreval("Insira o dado da posição ", i,
08         ", ", j, ":")
09         leia(mat[i,j])
10       fimpara
11     fimpara
12     para i de 1 ate 3 passo 1 faca
13       para j de 1 ate 3 passo 1 faca
14         se (i = j) entao
15           escreva(mat[i,j], " ")
16         senao
17           escreva("0 ")
18         fimse
19       fimpara
20     fimpara
21   FimAlgoritmo
```

21 Fimalgoritmo

- 9) Desenvolva um algoritmo que preencha uma matriz numérica de dimensões 3x3. Ao final, o algoritmo deve mostrar o somatório de todos os elementos da matriz, bem como a média aritmética entre todos os termos.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

Exemplo de execução – Exercício 9 - Caso de teste

```
Insira o dado da posição 1, 1:
1
Insira o dado da posição 1, 2:
2
Insira o dado da posição 1, 3:
3
Insira o dado da posição 2, 1:
4
Insira o dado da posição 2, 2:
5
Insira o dado da posição 2, 3:
6
Insira o dado da posição 3, 1:
7
Insira o dado da posição 3, 2:
8
Insira o dado da posição 3, 3:
9
Somatório: 45
Média: 5
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

MATRIZES – Exercício 9 – Solução

```
01 Algoritmo "processaMatriz"
02   Var mat: Vetor[1..3,1..3] de real
03       soma, media: real
04       i, j: inteiro
05 Inicio
06   soma <- 0
```

```
07     para i de 1 ate 3 passo 1 faca
08         para j de 1 ate 3 passo 1 faca
09             escreval("Insira o dado da posição ", i,
10                 ", ", j, ":")
11             leia(mat[i,j])
12             soma <- soma + mat[i,j]
13         fimpara
14     fimpara
15     media <- soma / 9
16     escreval("Somatório: ", soma)
17     escreval("Média: ", media)
18 Fimalgoritmo
```

- 10)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que carregue uma matriz 2x2, calcule e mostre uma matriz resultante que será a própria matriz digitada multiplicada pelo maior elemento da matriz.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

#### Exemplo de execução – Exercício 10 – Caso de teste

```
Insira o dado da posição 1, 1:
1
Insira o dado da posição 1, 2:
2
Insira o dado da posição 2, 1:
3
Insira o dado da posição 2, 2:
4
4 8
12 16
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### MATRIZES – Exercício 10 – Solução

```
01 Algoritmo "matrizMultMaior"
02     Var mat: Vetor[1..2,1..2] de real
03     maior: real
04     i, j: inteiro
05 Inicio
06     para i de 1 ate 2 passo 1 faca
07         para j de 1 ate 2 passo 1 faca
08             escreval("Insira o dado da posição ", i,
09                 ", ", j, ":")
10             leia(mat[i,j])
```



```
11      fimpara
12  fimpara
13  maior <- mat[1,1]
14  para i de 1 ate 2 passo 1 faca
15      para j de 1 ate 2 passo 1 faca
16          se (mat[i,j] > maior) entao
17              maior <- mat[i,j]
18          fimse
19      fimpara
20  fimpara
21  para i de 1 ate 2 passo 1 faca
22      para j de 1 ate 2 passo 1 faca
23          mat[i,j] <- mat[i,j]*maior
24          escreva(mat[i,j], " ")
25      fimpara
26      escreval()
27  fimpara
28  Fimalgoritmo
```

- 11)** Desenvolva um algoritmo que preencha uma matriz numérico de dimensões 3x3. Após preencher toda a matriz, o usuário deve inserir uma chave de busca X. Caso exista algum número igual a X, dentro da matriz, o algoritmo deve mostrar, na tela, os índices da linha e da coluna da posição na qual na qual X foi encontrado pela primeira vez. Caso contrário, o algoritmo deve se encerrar com uma única mensagem, dizendo "Chave não encontrada.".

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – *escreva* ou *escreval*). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – *leia*).

Exemplo de execução – Exercício 11 - Caso de teste 1

```
Insira o dado da posição 1, 1:
1
Insira o dado da posição 1, 2:
2
Insira o dado da posição 1, 3:
3
Insira o dado da posição 2, 1:
4
Insira o dado da posição 2, 2:
5
Insira o dado da posição 2, 3:
6
Insira o dado da posição 3, 1:
7
Insira o dado da posição 3, 2:
8
Insira o dado da posição 3, 3:
9
```

```
Insira a chave de busca:
5
Chave encontrada na linha: 2 coluna: 2
```

#### Exemplo de execução – Exercício 11 - Caso de teste 2

```
Insira o dado da posição 1, 1:
1
Insira o dado da posição 1, 2:
2
Insira o dado da posição 1, 3:
3
Insira o dado da posição 2, 1:
4
Insira o dado da posição 2, 2:
5
Insira o dado da posição 2, 3:
6
Insira o dado da posição 3, 1:
7
Insira o dado da posição 3, 2:
8
Insira o dado da posição 3, 3:
9
Insira a chave de busca:
10
Chave não encontrada.
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### MATRIZES – Exercício 11 – Solução

```
01 Algoritmo "buscaMatricial"
02     Var X: real
03     mat: Vetor[1..3,1..3] de real
04     i, j, lin, col: inteiro
05     achou: logico
06 Inicio
07     para i de 1 ate 3 passo 1 faca
08         para j de 1 ate 3 passo 1 faca
09             escreval("Insira o dado da posição ", i,
10                 ", ", j, ":")
11                 leia(mat[i,j])
12         fimpara
13     fimpara
14     escreval("Insira a chave de busca:")
15     leia(X)
16     achou <- falso
17     i <- 1
18     enquanto (i <= 3) e (achou = falso) faca
19         j <- 1
20         enquanto (j <= 3) e (achou = falso) faca
```

```
20         se (mat[i,j] = X) entao
21             achou <- verdadeiro
22             lin <- i
23             col <- j
24             fimse
25             j <- j + 1
26         fimenquanto
27         i <- i + 1
28     fimenquanto
29     se (achou <> falso) entao
30         escreval("Chave encontrada na linha: ", lin, "
coluna: ", col, " ")
31     senao
32         escreval("Chave não encontrada.")
33     fimse
34 Fimalgoritmo
```

- 12)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que carregue uma matriz 3 x 5 com números inteiros e some cada uma das linhas, armazenando o resultado das somas em um vetor. A seguir, multiplique cada elemento da matriz pela soma da respectiva linha daquele elemento e mostre a matriz resultante.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

Exemplo de execução – Exercício 12 - Caso de teste

```
Insira o dado da posição 1, 1:
1
Insira o dado da posição 1, 2:
2
Insira o dado da posição 1, 3:
3
Insira o dado da posição 1, 4:
4
Insira o dado da posição 1, 5:
5
Insira o dado da posição 2, 1:
10
Insira o dado da posição 2, 2:
20
Insira o dado da posição 2, 3:
30
Insira o dado da posição 2, 4:
40
Insira o dado da posição 2, 5:
50
Insira o dado da posição 3, 1:
11
```

```
Insira o dado da posição 3, 2:
12
Insira o dado da posição 3, 3:
13
Insira o dado da posição 3, 4:
14
Insira o dado da posição 3, 5:
15
15    30    45    60    75
1500  3000  4500  6000  7500
715   780   845   910   975
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### MATRIZES – Exercício 12 – Solução

```
01 Algoritmo "processaMatriz2"
02   Var mat: Vetor[1..3,1..5] de real
03   vet: Vetor[1..3] de real
04   i, j: inteiro
05 Inicio
06   para i de 1 ate 3 passo 1 faca
07     vet[i] <- 0
08     para j de 1 ate 5 passo 1 faca
09       escreval("Insira o dado da posição ", i, ",
10               ", j, ":")
11       leia(mat[i,j])
12       vet[i] <- vet[i] + mat[i,j]
13     fimpara
14   fimpara
15   para i de 1 ate 3 passo 1 faca
16     para j de 1 ate 5 passo 1 faca
17       mat[i,j] <- vet[i]*mat[i,j]
18       escreva(mat[i,j], " ")
19     fimpara
20   fimpara
21 Fimalgoritmo
```

- 13)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que carregue uma matriz 10x3 com as três notas de dez alunos. Mostre um relatório com o número do aluno (número da linha) e a prova em que cada aluno obteve menor nota. Ao final do relatório, mostre quantos alunos tiveram menor nota na prova 1, quantos alunos tiveram menor nota na prova 2 e quantos alunos tiveram menor nota na prova 3.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens

geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

#### Exemplo de execução – Exercício 13 - Caso de teste

```
Insira a nota 1 do aluno 1:
10
Insira a nota 2 do aluno 1:
10
Insira a nota 3 do aluno 1:
10
Insira a nota 1 do aluno 2:
9
Insira a nota 2 do aluno 2:
9
Insira a nota 3 do aluno 2:
9
Insira a nota 1 do aluno 3:
8
Insira a nota 2 do aluno 3:
8
Insira a nota 3 do aluno 3:
8
Insira a nota 1 do aluno 4:
7
Insira a nota 2 do aluno 4:
8
Insira a nota 3 do aluno 4:
9
Insira a nota 1 do aluno 5:
9
Insira a nota 2 do aluno 5:
8
Insira a nota 3 do aluno 5:
7
Insira a nota 1 do aluno 6:
8
Insira a nota 2 do aluno 6:
7
Insira a nota 3 do aluno 6:
9
Insira a nota 1 do aluno 7:
9
Insira a nota 2 do aluno 7:
7
Insira a nota 3 do aluno 7:
8
Insira a nota 1 do aluno 8:
4
Insira a nota 2 do aluno 8:
5
Insira a nota 3 do aluno 8:
6
Insira a nota 1 do aluno 9:
```

```
6
Insira a nota 2 do aluno 9:
5
Insira a nota 3 do aluno 9:
4
Insira a nota 1 do aluno 10:
6
Insira a nota 2 do aluno 10:
4
Insira a nota 3 do aluno 10:
5
Aluno 1
Prova de menor nota: 1
Aluno 2
Prova de menor nota: 1
Aluno 3
Prova de menor nota: 1
Aluno 4
Prova de menor nota: 1
Aluno 5
Prova de menor nota: 3
Aluno 6
Prova de menor nota: 2
Aluno 7
Prova de menor nota: 2
Aluno 8
Prova de menor nota: 1
Aluno 9
Prova de menor nota: 3
Aluno 10
Prova de menor nota: 2
Menores notas na primeira prova: 5
Menores notas na segunda prova: 3
Menores notas na terceira prova: 2
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### MATRIZES – Exercício 13 – Solução

```
01 Algoritmo "tresNotasDezAlunos"
02   Var notas: Vetor[1..10,1..3] de real
03       q1, q2, q3, menor, p_menor: real
04       i, j: inteiro
05 Inicio
06   para i de 1 ate 10 passo 1 faca
07       para j de 1 ate 3 passo 1 faca
08           escreval("Insira a nota ", j, " do aluno
09               ", i, ":")
10               leia(notas[i,j])
11           fimpara
12       fimpara
```



```
12     q1 <- 0
13     q2 <- 0
14     q3 <- 0
15     para i de 1 ate 10 passo 1 faca
16         escreval("Aluno ", i)
17         menor <- notas[i,1]
18         p_menor <- 1
19         para j de 1 ate 3 passo 1 faca
20             se (notas[i,j] < menor) entao
21                 menor <- notas[i,j]
22                 p_menor <- j
23             fimse
24         fimpara
25         escreval("Prova de menor nota: ", p_menor)
26         escolha (p_menor)
27             caso 1
28                 q1 <- q1 + 1
29             caso 2
30                 q2 <- q2 + 1
31             caso 3
32                 q3 <- q3 + 1
33         fimescolha
34     fimpara
35     escreval("Menores notas na primeira prova: ", q1)
36     escreval("Menores notas na segunda prova: ", q2)
37     escreval("Menores notas na terceira prova: ", q3)
38 Fimalgoritmo
```

- 14)** Faça um programa que seja capaz de armazenar os dados de três pessoas: nome, idade, peso e altura. Ao final, o algoritmo deve mostrar, na tela, o nome e a idade da primeira pessoa e o peso e altura da última pessoa.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

Exemplo de execução – Exercício 14 - Caso de teste

```
Digite o nome 1:Jairo
Insira a idade 1:20
Digite o peso 1:78
Insira a altura 1:1,78
Digite o nome 2:Camila
Insira a idade 2:25
Digite o peso 2:65
Insira a altura 2:1,67
Digite o nome 3:Lúcia
Insira a idade 3:78
Digite o peso 3:69
Insira a altura 3:1,67
```

```
Nome da primeira pessoa: Jairo
Idade da primeira pessoa: 20
Peso da última pessoa: 69
Altura da última pessoa: 1.67
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### REGISTROS – Exercício 14 – Solução

```
01 Algoritmo "dadosPessoas"
02   Tipo
03     pessoa = registro
04       nome: caractere
05       idade: inteiro
06       peso: real
07       altura: real
08   Fimregistro
09
10   Var pessoas: Vetor[1..3] de pessoa
11       i: inteiro
12 Início
13   para i de 1 ate 3 passo 1 faca
14     escreva("Digite o nome ", i, ":")
15     leia(pessoas[i].nome)
16     escreva("Insira a idade ", i, ":")
17     leia(pessoas[i].idade)
18     escreva("Digite o peso ", i, ":")
19     leia(pessoas[i].peso)
20     escreva("Insira a altura ", i, ":")
21     leia(pessoas[i].altura)
22   fimpara
23   escreval("Nome da primeira pessoa: ", pessoas[1].nome)
24   escreval("Idade da primeira pessoa: ",
25 pessoas[1].idade)
26   escreval("Peso da última pessoa: ", pessoas[3].peso)
27   escreval("Altura da última pessoa: ",
28 pessoas[3].altura)
29 Fimalgoritmo
```

- 15)** Faça um programa que seja capaz de armazenar os dados de uma pessoa: nome, idade, peso e altura. Seu programa deve ser capaz de armazenar 5 pessoas. Ao final dos cadastros, o seu programa deve imprimir, na tela, todas as informações de todas as pessoas. Seu programa deve mostrar, também, o nome da pessoa mais magra, nome da pessoa mais baixa e a média das idades de todas as pessoas.

Quer dicas de como seu algoritmo deveria funcionar? Observe o quadro abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens

geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

#### Exemplo de execução – Exercício 15 - Caso de teste

```
Digite o nome 1:José da Silva
Insira a idade 1:30
Digite o peso 1:89
Insira a altura 1:1,89
Digite o nome 2:Carlos Almeida
Insira a idade 2:40
Digite o peso 2:112
Insira a altura 2:1,80
Digite o nome 3:Camila Oliveira
Insira a idade 3:21
Digite o peso 3:56
Insira a altura 3:1,60
Digite o nome 4:Maria da Silva
Insira a idade 4:24
Digite o peso 4:58
Insira a altura 4:1,56
Digite o nome 5:Mário Lacerda
Insira a idade 5:56
Digite o peso 5:78
Insira a altura 5:1,55
Pessoa mais magra: Camila Oliveira
Pessoa mais alta: Mário Lacerda
Média das idades: 34.2
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### REGISTROS – Exercício 15 – Solução

```
01 Algoritmo "dadosPessoas"
02     Tipo
03         pessoa = registro
04             nome: caractere
05             idade: inteiro
06             peso: real
07             altura: real
08     Fimregistro
09     Var lista_prod: Vetor[1..5] de pessoa
10         p_magra, a_baixa, m_idade: real
11         n_magra, n_baixa: caractere
12         i: inteiro
13 Inicio
14     para i de 1 ate 5 passo 1 faca
15         escreva("Digite o nome ", i, ":")
16         leia(lista_prod[i].nome)
17         escreva("Insira a idade ", i, ":")
18         leia(lista_prod[i].idade)
19         escreva("Digite o peso ", i, ":")
```

```
20     leia(lista_prod[i].peso)
21     escreva("Insira a altura ", i, ":")
22     leia(lista_prod[i].altura)
23     fimpara
24     p_magra <- lista_prod[1].peso
25     a_baixa <- lista_prod[1].altura
26     m_idade <- lista_prod[1].idade
27     para i de 2 ate 5 passo 1 faca
28         se (lista_prod[i].peso < p_magra) entao
29             p_magra <- lista_prod[i].peso
30             n_magra <- lista_prod[i].nome
31         fimse
32         se (lista_prod[i].altura < a_baixa) entao
33             a_baixa <- lista_prod[i].altura
34             n_baixa <- lista_prod[i].nome
35         fimse
36         m_idade <- m_idade + lista_prod[i].idade
37     fimpara
38     m_idade <- m_idade / 5
39     escreval("Pessoa mais magra: ", n_magra)
40     escreval("Pessoa mais alta: ", n_baixa)
41     escreval("Média das idades: ", m_idade)
42 Fimalgoritmo
```

- 16) Faça um programa que seja capaz de armazenar os dados de um produto: código, nome, valor e quantidade. Seu programa deve ser capaz de armazenar 5 produtos. Ao final dos cadastros, o usuário deve inserir o código de um produto e o seu programa deve imprimir, na tela, as informações daquele produto. Caso o produto não se encontre cadastrado, deve-se informar ao usuário a seguinte mensagem: "código não encontrado".

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – escreva ou escreval). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – leia).

#### Exemplo de execução – Exercício 16 - Caso de teste 1

```
Insira o código do produto  1:
10
Insira o nome do produto  1:
Arroz
Insira o valor do produto  1:
2,50
Insira a quantidade do produto  1:
1
Insira o código do produto  2:
20
Insira o nome do produto  2:
Feijão
Insira o valor do produto  2:
```

```
5,60
Insira a quantidade do produto 2:
4
Insira o código do produto 3:
30
Insira o nome do produto 3:
Farofa
Insira o valor do produto 3:
1,20
Insira a quantidade do produto 3:
2
Insira o código do produto 4:
40
Insira o nome do produto 4:
Sabonete
Insira o valor do produto 4:
0,98
Insira a quantidade do produto 4:
10
Insira o código do produto 5:
50
Insira o nome do produto 5:
Picanha
Insira o valor do produto 5:
60,00
Insira a quantidade do produto 5:
3
Insira a chave de busca:
30
Nome: Sabonete
Valor: 0.98
Quantidade: 10
```

#### Exemplo de execução - Exercício 16 - Caso de teste 2

```
Insira o código do produto 1:
10
Insira o nome do produto 1:
Arroz
Insira o valor do produto 1:
2,50
Insira a quantidade do produto 1:
1
Insira o código do produto 2:
20
Insira o nome do produto 2:
Feijão
Insira o valor do produto 2:
5,60
Insira a quantidade do produto 2:
4
Insira o código do produto 3:
30
Insira o nome do produto 3:
```

```
Farofa
Insira o valor do produto 3:
1,20
Insira a quantidade do produto 3:
2
Insira o código do produto 4:
40
Insira o nome do produto 4:
Sabonete
Insira o valor do produto 4:
0,98
Insira a quantidade do produto 4:
10
Insira o código do produto 5:
50
Insira o nome do produto 5:
Picanha
Insira o valor do produto 5:
60,00
Insira a quantidade do produto 5:
3
Insira a chave de busca:
99
Código não encontrado
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### REGISTROS - Exercício 16 - Solução

```
01 Algoritmo "buscaSequencial"
02     Tipo
03         produto = registro
04         codigo: inteiro
05         nome: caractere
06         valor: real
07         qtde: real
08     Fimregistro
09     Var lista_prod: Vetor[1..5] de produto
10         i, X, p: inteiro
11         achou: logico
12 Inicio
13     para i de 1 ate 5 passo 1 faca
14         escreval("Insira o código do produto ", i, ":")
15         leia(lista_prod[i].codigo)
16         escreval("Insira o nome do produto ", i, ":")
17         leia(lista_prod[i].nome)
18         escreval("Insira o valor do produto ", i, ":")
19         leia(lista_prod[i].valor)
20         escreval("Insira a quantidade do produto ", i,
21         ":")
21         leia(lista_prod[i].qtde)
```



```
22     fimpara
23     escreval("Insira a chave de busca:")
24     leia(X)
25     achou <- falso
26     i <- 1
27     enquanto (i <= 5) e (achou = falso) faca
28         se (lista_prod[i].codigo = X) entao
29             achou <- verdadeiro
30             p <- i
31         fimse
32         i <- i + 1
33     fimenquanto
34     se (achou <> falso) entao
35         escreval("Nome: ", lista_prod[i].nome)
36         escreval("Valor: ", lista_prod[i].valor)
37         escreval("Quantidade: ", lista_prod[i].qtde)
38     senao
39         escreval("Código não encontrado.")
40     fimse
41 Fimalgoritmo
```