

Professor Pietro Martins de Oliveira

AL GO RIT MOS

do início ao fim

*PACOTE DE EXERCÍCIOS 4:
ESTRUTURAS DE REPETIÇÃO*

- 1) Desenvolva um algoritmo que some todos os números inteiros compreendidos entre 1 e 10 (inclusive).

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros 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

Somatório: 55

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 1 – Solução

```
01 Algoritmo "somaDe1a10"
02   Var soma, i: inteiro
03   Inicio
04     soma <- 0
05     i <- 1
06     enquanto (i <= 10) faca
07       soma <- soma + i
08       i <- i + 1
09     fimenquanto
10     escreval("Somatório: ", soma)
11 Fimalgoritmo
```

- 2) Desenvolva um algoritmo que receba dois números inteiros positivos A e B. Exiba na tela todos os números inteiros compreendidos entre A e B, excluindo os próprios A e B. Suponha que o usuário respeite o enunciado e insira valores válidos para A e B.

Exemplo de execução – Exercício 2 – Caso de teste

```
Insira o valor de A:
1
Insira o valor de B:
10
Série numérica:
2 3 4 5 6 7 8 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, segue aí uma solução que funciona:

ESTRUTURAS DE REPETIÇÃO – Exercício 2 – Solução

```
01 Algoritmo "inteirosEntreAeB"
02   Var A, B, i: inteiro
03   Inicio
04     escreval("Insira o valor de A:")
05     leia(A)
06     escreval("Insira o valor de B:")
07     leia(B)
08     i <- A+1
09     escreval("Série numérica:")
10     enquanto (i < B) faça
11       escreva(i, " ")
12       i <- i + 1
13     fimenquanto
14   Fimalgoritmo
```

- 3) Desenvolva um algoritmo que receba um número N e calcule o fatorial de N, sabendo que $N! = N * (N-1) * (N-2) * \dots * 3 * 2 * 1$.

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 3 – Caso de teste

```
Insira o valor de N:
5
O fatorial de 5 é 120
```

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 3 – Solução

```
01 Algoritmo "fatorial"
02   Var i, N, fat: inteiro
03   Inicio
04     escreval("Insira o valor de N:")
05     leia(N)
06     fat <- 1
07     para i de N ate 1 passo -1 faça
08       fat <- fat * i
09     fimpara
10     escreval("O fatorial de ", N, " é ", fat)
11   Fimalgoritmo
```

- 4) Desenvolva um algoritmo que receba um número N e imprima a tabuada de N, 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 4 - Caso de teste
Insira N:
5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

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:

```
ESTRUTURAS DE REPETIÇÃO - Exercício 4 - Solução
01 Algoritmo "tabuadaDeN"
02   Var N, i, res: inteiro
03   Inicio
04     escreval("Insira N:")
05     leia(N)
06     i <- 1
07     enquanto (i <= 10) faca
08       res <- N*i
09       escreval(N, " x ", i, " = ", res)
10       i <- i + 1
11     fimenquanto
12   Fimalgoritmo
```

- 5) Charlinho tem 11 anos, mede 1,40 metros de altura e cresce em média 2,1 centímetros ao ano. Seu irmão, Bossa, aos 14 anos, tem 1,45 metros de altura e cresce em média 1,1 centímetro por ano. Elabore um programa que conte quantos anos serão necessários para que a altura de Charlinho ultrapasse a de Bossa.

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
Serão necessários: 6 anos

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 5 – Solução

```
01 Algoritmo "alturaDosIrmãos"
02   Var char, boss: real
03   anos: inteiro
04 Inicio
05   anos <- 0
06   char <- 140
07   boss <- 145
08   enquanto (char <= boss) faça
09     anos <- anos + 1
10     char <- char + 2.1
11     boss <- boss + 1.1
12   fimenquanto
13   escreval("Serão necessários: ", anos, " anos")
14 Fimalgoritmo
```

- 6) (Adaptado de ASCENCIO e CAMPOS, 2008) Um funcionário de uma empresa recebe aumento salarial anualmente. Sabe-se que:
- a) esse funcionário foi contratado em 2015, com salário inicial de R\$ 1.000,00;
 - b) em 2016 recebeu aumento de 1,5% sobre seu salário inicial;
 - c) a partir de 2017 (inclusive), os aumentos salariais sempre corresponderam ao dobro da porcentagem do ano anterior.

Faça um programa que receba o ano atual determine o salário atual desse funcionário.

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
Qual é o ano atual?
2020
O salário atual é: 1539.0362176

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 6 – Solução

```
01 Algoritmo "atualizaSalario"
02   Var sal, sal_novo, porc: real
03   i, ano_atual: inteiro
04 Inicio
05   escreval("Qual é o ano atual?")
06   leia(ano_atual)
07   sal <- 1000
08   porc <- 1.5/100
09   sal_novo <- sal + sal*porc
10   para i de 2017 ate ano_atual passo 1 faca
11     porc <- 2 * porc
12     sal_novo <- sal_novo + sal_novo*porc
13   fimpara
14   escreval("O salário atual é: ", sal_novo)
15 Fimalgoritmo
```

- 7) Desenvolva um algoritmo que peça para o usuário inserir vários números inteiros. O algoritmo deverá contabilizar a quantidade de números positivos informados. Caso o usuário digite 0, o algoritmo deve mostrar quantidade contabilizada e encerrar.

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

```
Insira um número:
5
Insira um número:
5
Insira um número:
-1
Insira um número:
0
Quantidade de positivos: 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:

ESTRUTURAS DE REPETIÇÃO – Exercício 7 – Solução

```
01 Algoritmo "contaPositivos"
02   Var num, cont: inteiro
03 Inicio
04   cont <- 0
05   repita
```

```
06     escreval("Insira um número:")
07     leia(num)
08     se (num > 0) entao
09         cont <- cont + 1
10     fimse
11     ate (num = 0)
12     escreval("Quantidade de positivos: ", cont)
13 Fimalgoritmo
```

- 8) Desenvolva um algoritmo que peça para o usuário informar dois números. Após isso, o algoritmo deve mostrar cálculo o primeiro número elevado ao segundo. Ao final, o algoritmo deve perguntar se o usuário deseja repetir a operação. Caso o usuário insira o caractere "s", o algoritmo solicita novamente dois números e mostra novamente a potência do primeiro pelo segundo. Caso contrário, o algoritmo é encerrado.

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 primeiro número:

2

Insira o segundo número:

3

A elevando a B: 8

Deseja repetir a operação? (s/n)

s

Insira o primeiro número:

2

Insira o segundo número:

2

A elevando a B: 4

Deseja repetir a operação? (s/n)

n

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 8 – Solução

```
01 Algoritmo "repetePotenciacao"
02   Var A, B, res: real
03   op: caractere
04 Inicio
05   repita
06     escreval("Insira o primeiro número:")
07     leia(A)
08     escreval("Insira o segundo número:")
09     leia(B)
10     res <- exp(A, B)
11     escreval("A elevado a B: ", res)
12     escreval("Deseja repetir a operação? (s/n)")
13     leia(op)
14   ate (op <> "s")
15 Fimalgoritmo
```

- 9) Desenvolva um algoritmo que peça ao usuário que insira dois números inteiros positivos A e B, no qual A deve ser menor que B (supõe-se que o usuário irá respeitar esse enunciado). O algoritmo deve mostrar, na tela, todos os números ímpares compreendidos entre A e B (inclusive).

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 valor de A:

1

Insira o valor de B:

10

É impar: 1

É impar: 3

É impar: 5

É impar: 7

É impar: 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, segue aí uma solução que funciona:

ESTRUTURAS DE REPETIÇÃO – Exercício 9 – Solução

```
01 Algoritmo "imparesEntreAeB"
02   Var A, B, i: inteiro
03 Inicio
04   escreval("Insira o valor de A:")
05   leia(A)
06   escreval("Insira o valor de B:")
07   leia(B)
```



```
08     para i de A ate B passo 1 faca
09         se (i % 2 <> 0) entao
10             escreval("É ímpar:", i)
11         fimse
12     fimpara
13 Fimalgoritmo
```

- 10)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que leia dez conjuntos de dois valores, o primeiro representando o número do aluno e o segundo representando sua altura em centímetros. Encontre o aluno mais alto e o mais baixo. Mostre o número do aluno mais alto e o número do mais baixo, junto com suas alturas.

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 número do aluno:
1
Insira a altura do aluno:
1,90
Insira o número do aluno:
2
Insira a altura do aluno:
1,91
Insira o número do aluno:
3
Insira a altura do aluno:
1,92
Insira o número do aluno:
4
Insira a altura do aluno:
1,99
Insira o número do aluno:
5
Insira a altura do aluno:
1,98
Insira o número do aluno:
6
Insira a altura do aluno:
1,64
Insira o número do aluno:
7
Insira a altura do aluno:
1,69
Insira o número do aluno:
8
```

```
Insira a altura do aluno:
1,56
Insira o número do aluno:
9
Insira a altura do aluno:
1,66
Insira o número do aluno:
10
Insira a altura do aluno:
1,89
Número do maior aluno: 4
Altura do maior aluno: 1.99
Número do menor aluno: 8
Altura do menor aluno: 1.56
```

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 10 – Solução

```
01 Algoritmo "alunosAltura"
02   Var i, num, n_maior, n_menor: inteiro
03   alt, maior, menor: real
04 Inicio
05   para i de 1 ate 10 passo 1 faca
06     escreval("Insira o número do aluno:")
07     leia(num)
08     escreval("Insira a altura do aluno:")
09     leia(alt)
10     se (i = 1) entao
11       maior <- alt
12       n_maior <- num
13       menor <- alt
14       n_menor <- num
15     senao
16       se (alt > maior) entao
17         maior <- alt
18         n_maior <- num
19       fimse
20       se (alt < menor) entao
21         menor <- alt
22         n_menor <- num
23       fimse
24     fimse
25   fimpara
26   escreval("Número do maior aluno: ", n_maior)
27   escreval("Altura do maior aluno: ", maior)
28   escreval("Número do menor aluno: ", n_menor)
29   escreval("Altura do menor aluno: ", menor)
30 Fimalgoritmo
```

- 11)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que mostre os oito primeiros termos da sequência de Fibonacci.

0-1-1-2-3-5-8-13-21-34-55-...

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 11 - Caso de teste
Série de Fibonacci:
0 1 1 2 3 5 8 13
```

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:

```
ESTRUTURAS DE REPETIÇÃO – Exercício 11 – Solução
01 Algoritmo "oitoTermosFibonacci"
02   Var i, ant1, ant2, atual: inteiro
03   Inicio
04     ant1 <- 1
05     ant2 <- 0
06     escreval("Série de Fibonacci:")
07     escreva(ant2, " ")
08     escreva(ant1, " ")
09     para i <- 3 ate 8 faca
10       atual <- ant1 + ant2
11       escreva(atual, " ")
12       ant2 <- ant1
13       ant1 <- atual
14     fimpara
15   Fimalgoritmo
```

- 12)** (Adaptado de ASCENCIO e CAMPOS, 2008) Desenvolva um algoritmo que receba um número N e informe se N é um número primo, ou não. A saber: um número primo é um inteiro positivo que só pode ser dividido por ele mesmo e por um, apenas.

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 12 - Caso de teste XXX
Insira um número:
```

```
7
7 é primo.
```

Exemplo de execução – Exercício 12 – Caso de teste XXX

```
Insira um número:
10
10 não é primo.
```

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 12 – Solução

```
01 Algoritmo "numeroPrimo"
02   Var N, i, cont: inteiro
03   Inicio
04     escreval("Insira um número:")
05     leia(N)
06     cont <- 0
07     para i de 1 ate N passo 1 faca
08       se (N % i = 0) entao
09         cont <- cont + 1
10       fimse
11     fimpara
12     se (cont = 2) entao
13       escreval(N, " é primo.")
14     senao
15       escreval(N, " não é primo.")
16     fimse
17 Fimalgoritmo
```

- 13)** Desenvolva um algoritmo que mostre a tabuada de todos os números inteiros compreendidos entre 1 e 10 (inclusive).

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

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
```

1 x 9 = 9
1 x 10 = 10

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36

6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60

7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80

9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90

10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 13 – Solução

```
01 Algoritmo "tabuadaDo1ao10"
02   Var i, j, res: inteiro
03 Inicio
04   i <- 1
05   enquanto (i <= 10) faca
06     j <- 1
07     enquanto (j <= 10) faca
08       res <- i*j
09       escreval(i, " x ", j, " = ", res)
10       j <- j + 1
11     fimenquanto
12     escreval()
13     i <- i + 1
14   fimenquanto
15 Fimalgoritmo
```

- 14)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que leia um número N e que indique quantos valores inteiros e positivos devem ser lidos a seguir. Para cada número lido, mostre o fatorial desse valor.

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

Quantos números serão informados:

3

Insira o numero 1

5

O fatorial de 5 é 120

Insira o numero 2

4

O fatorial de 4 é 24

Insira o numero 3

3

O fatorial de 3 é 6

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 14 – Solução

```
01 Algoritmo "variosFatoriais"
02   Var N, fat, i, j, num: inteiro
03 Inicio
04   escreval("Quantos números serão informados:")
05   leia(N)
06   escreval()
07   para i de 1 ate N passo 1 faca
08     escreval("Insira o numero ", i)
09     leia(num)
10     fat <- 1
11     para j de 1 ate num passo 1 faca
12       fat <- fat * j
13     fimpara
14     escreval("O fatorial de ", num, " é ", fat)
15   escreval()
16   fimpara
17 Fimalgoritmo
```

- 15)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que leia um valor N inteiro e positivo, calcule e mostre o valor de E, conforme a fórmula a seguir:

$$E = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/N!$$

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

Insira o valor de N:

5

Resultado: 2.716666666666667

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 15 – Solução

```
01 Algoritmo "valorDeE"
02   Var n, i, j: inteiro
03   res, fat: real
04 Inicio
05   escreval("Insira o valor de N:")
06   leia(N)
07   res <- 1
08   para i de 1 ate N passo 1 faca
09     fat <- 1
10     para j <- 1 ate i passo 1 faca
11       fat <- fat * j
```

```
12         fimpara
13         res <- res + 1/fat
14     fimpara
15     escreval("Resultado: ", res)
16 Finalgoritmo
```

- 16)** (Adaptado de ASCENCIO e CAMPOS, 2008) Desenvolva um algoritmo que peça ao usuário que insira cinco conjuntos de dois números inteiros positivos (A, B), no qual A deve ser menor que B (supõe-se que o usuário irá respeitar esse enunciado). Para cada dupla (A, B), informada pelo usuário, o algoritmo deve mostrar, na tela, todos os números pares compreendidos entre A e B (inclusive).

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 16 - Caso de teste

Insira um valor de A:

1

Insira um valor de B:

5

É par: 2

É par: 4

Insira um valor de A:

6

Insira um valor de B:

10

É par: 6

É par: 8

É par: 10

Insira um valor de A:

11

Insira um valor de B:

15

É par: 12

É par: 14

Insira um valor de A:

16

Insira um valor de B:

19

É par: 16

É par: 18

Insira um valor de A:

20

```
Insira um valor de B:  
21  
É par: 20
```

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:

ESTRUTURAS DE REPETIÇÃO – Exercício 16 – Solução

```
01 Algoritmo "paresDeAaB"  
02   Var A, B, i, j: inteiro  
03 Inicio  
04   para i de 1 ate 5 passo 1 faca  
05     escreval("Insira um valor de A:")  
06     leia(A)  
07     escreval("Insira um valor de B:")  
08     leia(B)  
09     para j de A ate B passo 1 faca  
10       se (j % 2 = 0) entao  
11         escreval("É par:", j)  
12       fimse  
13     fimpara  
14   fimpara  
15 Fimalgoritmo
```