

Professor Pietro Martins de Oliveira

AL GO RIT MOS

do início ao fim

*PACOTE DE EXERCÍCIOS 2:
SUB-ROTINAS*

- 1) (Adaptado de ASCENCIO e CAMPOS, 2008) O custo de um carro novo ao consumidor final é o preço de fábrica somado ao percentual de lucro do distribuidor, acrescido dos impostos aplicados ao preço de fábrica. Faça um programa que receba o preço de fábrica de um veículo, o percentual de lucro do distribuidor e o percentual de impostos. Em cada item, crie uma função distinta para calcular e retornar:

- a) o valor correspondente ao lucro do distribuidor;
- b) o valor correspondente aos impostos;
- c) o preço final do veículo.

Após criar cada uma das funções, desenvolva um algoritmo que declare e invoque cada uma das funções, mostrando o lucro do distribuidor, os impostos e o valor final do veículo.

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 preço de fábrica:
20000
Insira a porcentagem de lucro do distribuidor:
30
Insira a porcentagem de impostos:
30
Lucro do distribuidor: R$ 6000
Impostos: R$ 6000
Valor final: R$ 32000
```

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:

SUB-ROTINAS – Exercício 1 – Solução

```
01 Algoritmo "carroNovo"
02   Funcao calcLucroDist(p_fab: real; por_dist:
    real):real
03   Inicio
04       retorne p_fab * por_dist / 100
05   Fimfuncao
06
07   Funcao calcImpostos(p_fab: real; por_imp: real):real
08   Inicio
09       retorne p_fab * por_imp / 100
10   Fimfuncao
11
12   Funcao calcPrecoFinal(p_fab: real; v_dist: real;
    v_imp:real):real
13   Inicio
14       retorne p_fab + v_dist + v_imp
```

```
15      Fimfuncao
16
17      Var preco_f, porce_dist, porc_i, lucr_dst, imp,
vlr_f: real
18  Início
19      escreval("Insira o preço de fábrica:")
20      leia(preco_f)
21      escreval("Insira a porcentagem de lucro do
distribuidor:")
22      leia(porce_dist)
23      escreval("Insira a porcentagem de impostos:")
24      leia(porc_i)
25      lucr_dst <- calcLucroDist(preco_f, porce_dist)
26      imp <- calcImpostos(preco_f, porc_i)
27      vlr_f <- calcPrecoFinal(preco_f, lucr_dst, imp)
28      escreval("Lucro do distribuidor: R$", lucr_dst)
29      escreval("Impostos: R$", imp)
30      escreval("Valor final: R$", vlr_f)
31  Fimalgoritmo
```

- 2) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o número de horas trabalhadas por um gestor e o valor do salário mínimo vigente. Crie uma função que calcule o salário a receber do gestor, seguindo as regras abaixo:

- I - a hora trabalhada vale a metade do salário mínimo;
 - II - o salário bruto equivale ao número de horas trabalhadas multiplicado pelo valor da hora trabalhada;
 - III - o imposto equivale a 3% do salário bruto;
 - IV - o salário a receber equivale ao salário bruto menos o imposto.
- Crie um algoritmo que invoque a respectiva função e mostre o salário a receber.

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 número de horas trabalhadas:
180
Insira valor do salário mínimo:
1000
Salário a receber: 87300
```

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:

SUB-ROTINAS – Exercício 2 – Solução

```
01 Algoritmo "salarioGestor"
02   Funcao calcLucroDist(n_hr: real; s_min: real):real
03     Var hr_t, s_brt, imp, s_rec: real
04     Inicio
05       hr_t <- s_min/2
06       s_brt <- n_hr * hr_t
07       imp <- s_brt * 0.03
08       s_rec <- s_brt - imp
09       retorne s_rec
10   Fimfuncao
11   Var horas_trab, sal_min, sal_rec: real
12   Inicio
13     escreval("Insira número de horas trabalhadas:")
14     leia(horas_trab)
15     escreval("Insira valor do salário mínimo:")
16     leia(sal_min)
17     sal_rec <- calcLucroDist(horas_trab, sal_min)
18     escreval("Salário a receber: ", sal_rec)
19   Fimalgoritmo
```

- 3) (Adaptado de ASCENCIO e CAMPOS, 2008) Pedro comprou um saco de ração para seus gatos, com o peso em quilos. Faça uma função que receba o peso do saco de ração, em quilos, o número de gatos e a quantidade de ração fornecida para cada gato por dia, em gramas. A função deve retornar o total de quilos de ração restante no saco, após um dia de consumo. Assim sendo, considerando que Pedro possui dois gatos, crie um algoritmo que invoque a função recém criada para calcular e mostrar quanto restará de ração no saco após cinco dias.

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

```
Qual o peso do saco (quilos)?
20
Qual o número de gatos?
2
Qual o peso da porção diária (gramas)?
250
Após 5 dias, sobrarão: 17.5 Kg de ração
```

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:

SUB-ROTINAS – Exercício 3 – Solução


```
01 Algoritmo "gatosPedro"
02   Funcao calcRacaoSaco(s: real; n: real; q: real):real
03     Var r: real
04     Inicio
05       r <- s - (n*q/1000)
06       retorne r
07   Fimfuncao
08   Var sacco, n_gatos, qtde_gr, sobra: real
09 Inicio
10   escreval("Qual o peso do sacco (quilos)?")
11   leia(sacco)
12   escreval("Qual o número de gatos?")
13   leia(n_gatos)
14   escreval("Qual o peso da porção diária (gramas)?")
15   leia(qtde_gr)
16   sobra <- calcRacaoSaco(sacco, n_gatos, qtde_gr)
17   sobra <- calcRacaoSaco(sobra, n_gatos, qtde_gr)
18   sobra <- calcRacaoSaco(sobra, n_gatos, qtde_gr)
19   sobra <- calcRacaoSaco(sobra, n_gatos, qtde_gr)
20   sobra <- calcRacaoSaco(sobra, n_gatos, qtde_gr)
21   escreval("Após 5 dias, sobrarão: ", sobra, " Kg de
22   ração")
23 Fimalgoritmo
```

- 4) (Adaptado de ASCENCIO e CAMPOS, 2008) Cada degrau de uma escada tem X cm de altura. Faça uma função que receba essa altura, em centímetros, e a altura que o usuário deseja alcançar ao subir a escada, em metros. A função deve retornar o número de degraus necessários para se atingir a altura desejada (desprezando a altura do próprio usuário). Em seguida, crie um algoritmo para que o usuário possa informar os dados de entrada da função e, ao final, calcule e mostre o número de degraus.

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 – 4 - Caso de teste
Insira a altura de cada degrau (cm):
25
Insira a altura da escada (m):
2
Número de degraus: 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:

SUB-ROTINAS – Exercício 4 – Solução

```
01 Algoritmo "degrausEscada"
02   Funcao calcDegraus(a_deg: inteiro; a_esc:
03   real):inteiro
04   Inicio
05       retorne int(a_esc*100) DIV a_deg
06   Fimfuncao
07   Var escada: real
08       degrau, n_degraus: inteiro
09 Inicio
10     escreval("Insira a altura de cada degrau (cm):")
11     leia(degrau)
12     escreval("Insira a altura da escada (m):")
13     leia(escada)
14     n_degraus <- calcDegraus(degrau, escada)
15     escreval("Número de degraus: ", n_degraus)
16 Fimalgoritmo
```

5) (Adaptado de ASCENCIO e CAMPOS, 2008) Sabe-se que o quilowatt de energia custa um milésimo do salário mínimo. Faça um procedimento que receba o valor do salário mínimo e quantidade de quilowatts consumida por uma residência. O procedimento deve calcular e retornar através de passagem de parâmetros por referência:

- a) o valor, em reais, de cada quilowatt;
- b) o valor, em reais, a ser pago por essa residência;
- c) o valor, em reais, a ser pago com desconto de 15%.

Sabendo disso, desenvolva um algoritmo que peça para o usuário inserir o valor do salário mínimo e a quantidade de quilowatts consumida. Invoque o respectivo procedimento e mostre, na tela, as informações dos itens a), b) e c).

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 salário mínimo:
1000
Insira a quantidade de KW gastos:
200
Valor de 1 KW (em R$): 1
Valor a ser pago (em R$): 200
Valor com desconto de 15% (em R$): 170
```

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:

```
SUB-ROTINAS - Exercício 5 - Solução
01 Algoritmo "calculaEnergético"
02   Procedimento calcKW(v_sal: real; q_kw: real; Var
03   v_kw: real; Var v_rs: real; Var v_dsc: real)
04     Inicio
05       v_kw <- v_sal / 1000
06       v_rs <- v_kw * q_kw
07       v_dsc <- v_rs - v_rs*0.15
08     Fimprocedimento
09     Var val_sal, qtde_kw, val_kw, val_reais, val_desc:
10     real
11     Inicio
12       escreval("Insira o salário mínimo:")
13       leia(val_sal)
14       escreval("Insira a quantidade de KW gastos:")
15       leia(qtde_kw)
16       calcKW(val_sal, qtde_kw, val_kw, val_reais, val_desc)
17       escreval("Valor de 1 KW (em R$): ", val_kw)
18       escreval("Valor a ser pago (em R$): ", val_reais)
19       escreval("Valor com desconto de 15% (em R$): ",
20       val_desc)
21     Fimalgoritmo
```

- 6) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um procedimento que receba um número real, calcule e retorne:

- a) a parte inteira desse número;
- b) a parte fracionária desse número.

Crie um algoritmo que peça para o usuário inserir o número real e, em seguida, calcule e mostre o que se pede nos itens a) e b).

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 um número real:
3,14
Parte inteira: 3
Parte fracionária: 0.14
```

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:

SUB-ROTINAS - Exercício 6 - Solução

```
01 Algoritmo "processaNumero"
02   Procedimento numReal(n: real; Var i: inteiro; Var fr:
    real)
03     Inicio
04       i <- int(n)
05       fr <- n - i
06     Fimprocedimento
07     Var num, parte_frac: real
08       parte_int: inteiro
09   Inicio
10     escreval("Insira um número real:")
11     leia(num)
12     numReal(num, parte_int, parte_frac)
13     escreval("Parte inteira: ", parte_int)
14     escreval("Parte fracionária: ", parte_frac)
15 Fimalgoritmo
```

- 7) (Adaptado de ASCENCIO e CAMPOS, 2008) Crie um procedimento que receba a quantidade de dinheiro em reais que uma pessoa que vai viajar possui. Essa pessoa vai passar por vários países e precisa converter seu dinheiro em dólares, euro e libra esterlina. Sabe-se que a cotação do dólar é de R\$ 4,00, do euro é R\$ 4,25 e do iene é R\$ 0,10. O procedimento deverá fazer a leitura dos dados do usuário e exibir o resultado das conversões diretamente, sem passagem de parâmetros. Desenvolva um algoritmo que invoque o procedimento para realizar os cálculos.

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 o valor em reais:
100
Em dólares:  25
Em euros:    23.5294117647059
Em ienes:    1000
```

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:

SUB-ROTINAS – Exercício 7 – Solução

```
01 Algoritmo "converteReais"
02   Procedimento cotacoes()
03     Var reais: real
04   Inicio
05     escreval("Insira o valor em reais:")
06     leia(reais)
```



```
07     escreval("Em dólares: ", reais/4)
08     escreval("Em euros: ", reais/4.25)
09     escreval("Em ienes: ", reais/0.1)
10     Fimprocedimento
11 Inicio
12     cotacoes()
13 Fimalgoritmo
```