

Gabarito — Segunda Lista de Exercícios - Computational Thinking

1. Faça um algoritmo que receba um número e mostre uma mensagem caso este número seja maior que 10.

```
1 numero = float(input("Digite um numero: "))
2
3 if numero > 10:
4     print("O numero digitado e maior que 10")
```

2. Escrever um algoritmo que leia dois valores inteiro distintos e informe qual é o maior ou se houve um empate.

```
1 num1 = int(input("Digite primeiro numero: "))
2
3 num2 = int(input("Digite segundo numero: "))
4
5 if num1 > num2:
6     print(num1, " e maior que ", num2)
7 elif num1 < num2:
8     print(num1, " e maior que ", num2)
9 else:
10    print("EMPATE")
```

4. Escreva um algoritmo para ler o nome de 2 times e o número de gols marcados em uma partida. Escrever o nome do vencedor. Caso não haja vencedor deverá ser impresso a palavra EMPATE.

```
1 timeCasa = input("Time da casa:")
2 timeVisitante = input("Time visitante:")
3 golCasa = int(input("Gols da casa:"))
4 golVisitante = int(input("Gols do visitante:"))
5
6 if golCasa > golVisitante:
7     print("Time da casa venceu!")
8 elif golCasa < golVisitante:
9     print("Time visitante venceu!")
10 else:
11     print("Houve empate")
12
13 print(timeCasa, " ", golCasa, " X ", golVisitante, " ", timeVisitante)
```

5. A jornada de trabalho semanal é de 40 horas. O funcionário que trabalhar mais de 40 horas receberá hora extra, cujo valor é o valor da hora regular com um acréscimo de 50%. Escreva um algoritmo que leia o número de horas trabalhadas em um mês, o salário por hora e escreva o salário total do funcionário (considere que o mês possua 4 semanas exatas).

```
1 dias = int(input("Informe dias uteis: "))
2 horas_trab = int(input("Informe horas trabalhadas"))
3 salario_hora = float(input("Salario-hora: "))
4
5 salario = horas_trab * salario_hora
6 jornada = dias * 8
7 hora_extra = 0
8
9 if horas_trab > jornada:
10     #tem hora-extra para calcular
11     hora_extra = (horas_trab - jornada) * salario_hora / 2
12
13 print("Salario: {:.2f}".format(salario))
14
15 if hora_extra > 0:
16     print("Hora-extra: {:.2f}".format(hora_extra))
17     print("Salario Total: {:.2f}".format(salario + hora_extra))
```

6. Faça um programa para ler dois números inteiros A e B e informar se A é divisível por B.

```

1  a = int(input("Digite dividendo: "))
2  b = int(input("Digite divisor: "))
3
4  resto = a % b
5  if resto == 0:
6      print(a, " é divisível por ", b)

```

7. A raiz quadrada é uma operação que apenas aceita números positivos. Escreva um algoritmo que lê um número qualquer e retorna a raiz quadrada desse número se possível. Use a função `math.sqrt(<numero>)` para calcular a raiz quadrada em Python. Note que, para usar essa função, você terá que importar o módulo **math** antes.

```

1  import math
2
3  #coloque aqui o resto do seu código
4  #tudo na frente do sustenido é
5  #considerado um comentário em Python
6  numero = float(input("Digite um número: "))
7  if numero < 0:
8      print("Número negativo. Impossível calcular raiz quadrada")
9  else:
10     resultado = math.sqrt(numero)
11     print("Raiz quadrada de ", numero, " é ", resultado)

```

8. Escreva um algoritmo que recebe a idade de um nadador e mostra sua categoria conforme a tabela a seguir:

Categoria	Idade
Infantil	5 a 7
Juvenil	8 a 10
Adolescente	11 a 15
Adulto	16 a 30
Senior	acima de 30

```

1  idade = int(input("Digite a idade do nadador: "))
2  categoria = "Sem categoria"
3
4  if idade >= 5 and idade <= 7:
5      categoria = "Infantil"
6  elif idade >= 8 and idade <= 10:
7      categoria = "Juvenil"
8  elif idade >= 11 and idade <= 15:
9      categoria = "Adolescente"
10 elif idade >= 16 and idade <= 30:
11     categoria = "Adulto"
12 elif idade > 30:
13     categoria = "Senior"
14
15 print(categoria)

```

9. Uma equação de 2º grau é da forma: $ax^2 + bx + c = 0$, onde $a \neq 0$. Escreva um algoritmo que recebe os três coeficientes da equação, calcula e imprime as raízes reais se for possível. Use a seguinte fórmula para resolver a equação:

$$\Delta = b^2 - 4ac$$

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a}$$

$$x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

```

1  import math
2
3  aux = input("Digite a: ")
4
5  a = float(aux)
6  b = float(input("Digite b: "))
7  c = float(input("Digite c: "))
8
9  delta = b * b - 4 * a * c
10
11 if delta < 0:
12     print("A equacao nao possui raizes reais")
13 else:
14     raiz1 = -b + math.sqrt(delta)
15     raiz1 = raiz1 / (2 * a)
16
17     raiz2 = (-b - math.sqrt(delta)) / (2 * a)
18
19     print("Raiz 1", raiz1)
20     print("Raiz 2", raiz2)

```

10. Escreva um algoritmo que calcule o que deve ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento. Utilize os códigos da tabela a seguir para ler qual a condição de pagamento escolhida e efetuar o cálculo adequado.

código	condição de pagamento
1	A vista em dinheiro ou cheque, recebe 10% de desconto
2	A vista no cartão de crédito, recebe 5% de desconto
3	Em duas vezes, preço normal de etiqueta sem juros
4	Em quatro vezes, preço normal de etiqueta mais juros de 7%

```

1  preco = float(input("Informe o preco do produto: "))
2
3  opcao = int(input("Informe opcao de pagto: "))
4
5  novoPreco = 0.0
6  if opcao == 1:
7      print("Recebeu desconto de 10%")
8      novoPreco = preco * 0.9
9  elif opcao == 2:
10     print("Recebeu desconto de 5%")
11     novoPreco = preco * 0.95
12  elif opcao == 3:
13     novoPreco = preco
14  else:
15     print("Recebeu aumento de 7%")
16     novoPreco = preco * 1.07
17
18  print("Voce vai pagar {}", novoPreco)

```

11. Faça um algoritmo que leia as médias semestrais obtidas por um aluno na disciplina de Computational Thinking, o número de aulas ministradas e o número de aulas assistidas por

este aluno nesta disciplina. Calcule e mostre a média final deste aluno e diga se ele foi aprovado ou reprovado ou está de exame. Lembrando que a média do primeiro semestre tem peso 4 e a do segundo peso 6, além disso, o aluno tem que ter frequentado mais de 70% das aulas.

```

1  mediaPrimSem = float(input("Primeiro semestre: "))
2  mediaSegSem = float(input("Segundo semestre: "))
3
4  mediaFinal = (4 * mediaPrimSem + 6 * mediaSegSem) / 10
5
6  aulasDadas = int(input("numero de aulas dadas: "))
7  aulasAssistidas = int(input("numero de aulas assistidas: "))
8
9  frequencia = aulasAssistidas / aulasDadas * 100
10
11 print("A media final foi de {}".format(mediaFinal))
12 situacao = "Aprovado"
13
14 if mediaFinal < 4 or frequencia < 70:
15     situacao = "Reprovado"
16
17 if mediaFinal >= 4 and mediaFinal < 6:
18     situacao = "Exame"
19
20 print("O aluno foi ", situacao)

```

12. Desenvolva um algoritmo que informe se uma data é válida ou não. O algoritmo deverá ler 2 números inteiros, que representem o dia e o mês e informar se é um dia do mês válido. Desconsidere os casos de ano bissexto, ou seja, fevereiro têm 28 dias.
13. Agora, vamos acrescentar na verificação de data os casos de ano bissexto, ou seja, o ano que fevereiro tem 29 dias. Um ano é bissexto se:
 - a) o ano for divisível por 4
 - b) anos múltiplos de 100, não são bissextos
 - c) quando o ano for divisível por 400 ele é bissexto
 - d) as últimas regras prevalecem sobre as primeiras

Para exemplificar um pouco essas regras, observe que 1900 não foi bissexto mas 2000 foi.

```

1  dia = int(input("Digite o dia: "))
2  mes = int(input("Digite o mes: "))
3  ano = int(input("Digite o ano: "))
4
5  dataValida = True
6
7  if dia < 1 or dia > 31 or mes < 1 or mes > 12: #DATAS INEXISTENTES
8      dataValida = False
9
10 if dia == 31 and (mes == 4 or mes == 6 or mes == 9 or mes == 11):
11     print("Entrou aqui")
12     dataValida = False
13
14 if mes == 2 and dia > 28:
15     if dia == 29:
16         #aqui vamos verificar se o ano e bissexto
17         if ano % 4 != 0:
18             dataValida = False
19         elif ano % 100 == 0 and ano % 400 != 0:
20             dataValida = False
21
22     else: #situacao para os dias 30 e 31 de fevereiro
23         dataValida = False

```

```
24
25
26 #if dataValida == True:
27 if dataValida:
28     print("Data valida")
29 else:
30     print("Data invalida")
```

Boa sorte!