# Quarta Lista de Exercícios - Computational Thinking

1. Dada uma sequência de números inteiros onde o último elemento é o 0, escreva um algoritmo que calcula a soma dos números pares da sequência.

2. Dados o número $n$ de alunos de uma turma de Algoritmos e suas notas da primeira prova, determinar a média das notas dessa turma. Considere que o usuário digite as informações corretamente.

3. Altere o algoritmo anterior para, além da média, contar os alunos que tiraram entre 0 e $5,0$ ($0 \leq nota < 5,0$) e acima de $5,0$ ($nota \geq 5,0$).

4. Dados $n$ um inteiro positivo e uma sequência de $n$ números reais, escreva um algoritmo que conta e imprime a quantidade de números positivos e a quantidade de números negativos.

5. Escreva um algoritmo que, dados um número inteiro positivo $n$, imprime na tela a contagem de todos os divisores positivos de $n$.

6. Em uma prova de concurso com 70 questões haviam 20 pessoas concorrendo. Sabendo que cada questão vale 1 ponto, escreva um algoritmo que lê a pontuação da prova obtida de cada um dos candidatos e calcula:

   a) a maior e a menor nota

   b) o percentual de candidatos que acertaram até 20 questões, o percentual que acertaram de 21 a 50 e o percentual que acertou acima de 50 questões

7. A conversão de graus Fahrenheit para centígrados é obtida pela fórmula $C = \frac{5}{9}(F - 32)$. Escreva um algoritmo que calcule e escreva uma tabela de graus centígrados em função de graus Fahrenheit que variem de 50 a 150 Fahrenheit de 1 em 1.

8. Um número inteiro positivo $n$ é denominado primo se existirem apenas dois divisores inteiros positivos dele: o 1 e o próprio $n$. Escreva um algoritmo que recebe um inteiro $n$ e verifica se $n$ é primo ou não.

9. Dados um montante em dinheiro inicial $d$, uma taxa de juros mensal $j$ e um período de tempo em meses $t$, escreva um algoritmo que calcula o valor final em dinheiro se $d$ ficar aplicado a taxa de juros $j$ durante $t$ meses.

10. Escreva um algoritmo que recebe um inteiro positivo $n$ e calcula $n! = 1 \cdot 2 \cdot 3 \ldots \cdot (n-1) \cdot n$. Por exemplo, se $n = 6$, então $6! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 = 720$.

11. Se $F_n$ é o $n$-ésimo número da sequência de Fibonacci, podemos calculá-la através da seguinte fórmula de recorrência:

$$F_n = \begin{cases} 1 & \text{se } n = 1 \text{ ou } n = 2; \\ F_{n-1} + F_{n-2} & \text{se } n > 2 \end{cases}$$

   Vamos mostrar os 10 primeiros números da sequência de Fibonacci:

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55$$

   Escreva um algoritmo que dado $n$, calcula o $n$-ésimo número da sequência de Fibonacci.

12. Dizemos que um número natural n é palíndromo se o $1^{o}$ algarismo de n é igual ao seu último algarismo, o $2^{o}$ algarismo de n é igual ao penúltimo algarismo, e assim sucessivamente. Exemplos: 567765 e 32423 são palíndromos. 567675 não é palíndromo.

13. Vamos escrever um programa que consiste em um Jogo de Adivinhação. O jogo funciona do seguinte modo: sorteia-se um número inteiro aleatório entre 1 e 1000. Sua tarefa é tentar adivinhar o número sorteado através de "chutes". A cada chute o programa deverá informar se o número "sorteado"é maior, menor ou igual ao número "chutado". Quando o usuário acertar o número deverá ser impresso uma mensagem dizendo que ele acertou e a quantidade de chutes dados. Para gerar números aleatórios entre 1 e 1000 use as seguintes instruções dentro do seu programa Python.

```
import random

sorteado = random.randint(1,1001)
```

14. Dizemos que um inteiro positivo n é perfeito se for igual à soma de seus divisores positivos diferentes de n.

    Exemplo:

    6 é perfeito, pois $1 + 2 + 3 = 6$.

    Sua tarefa será a de escrever um algoritmo em Python que, dado um inteiro positivo $n$, verificar se $n$ é perfeito.

15. Dados um inteiro $n$ e uma seqüência de $n$ números inteiros, determinar o comprimento de um segmento crescente de comprimento máximo.

    Exemplos:

    Na seqüência 5, 10, 3, 2, 4, 7, 9, 8, 5 o comprimento do segmento crescente máximo é 4.

    Na seqüência 10, 8, 7, 5, 2 o comprimento de um segmento crescente máximo é 1.

16. Uma das maneiras de evitar erros na digitação de números como conta corrente, CPF, boleto bancário é a utilização de um ou mais dígitos de controle. Um dos métodos de cálculo é a utilização do método módulo 10. Segue a descrição do algoritmo: Dado um número inteiro $n$ devemos pegar cada dígito desse número começando pela casa das unidades e multiplicar, alternadamente, por 2 e por 1. Caso o resultado da multiplicação seja um número maior ou igual a 10 devemos simplificar esse valor somando os dois dígitos. Após feitas as multiplicações e as simplificações devemos somar todos os valores e calcular o resto da divisão dessa soma por 10. Se o resto for 0 o dígito de controle é zero, caso contrário o dígito de controle será 10 menos o resto.

    A Figura 1 pode servir de exemplo para o algoritmo móduto 10.

    Por exemplo vamos pegar o número do meio do boleto, o número corresponde a $n =$

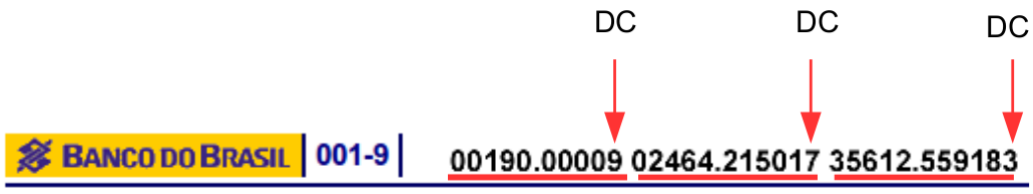


Figura 1: Trecho de boleto bancário do Banco do Brasil

246421501 e o dígito de controle será 7. Vamos efetuar os seguintes cálculos:

$$\begin{aligned} 1*2 &= 2 \\ 0*1 &= 0 \\ 5*2 &= 10 \\ 1*1 &= 1 \\ 2*2 &= 4 \\ 4*1 &= 4 \\ 6*2 &= 12 \\ 4*1 &= 4 \\ 2*2 &= 4 \end{aligned}$$

Daí somamos $2+0+1(1+0)+1+4+4+3(1+2)+4+4=23$. O resto da divisão de 23 por 10 é 3 e como ele é diferente de zero o dígito de controle de 246421501 será 7 $(10-3)$.

Escreva um algoritmo que lê um número inteiro positivo e calcula o seu dígito de controle usando o método do módulo 10.

17. Em quase todas as linguagens existe um método/função que calcula a raiz quadrada de um número. Contudo, vamos supor que na linguagem Python não exista tal método. Sua tarefa é tentar encontrar a raiz quadrada de um número por tentativa e erro, ou seja, seu algoritmo recebe um número real $n > 0$ e tenta encontrar sua raiz quadrada. Dica: pegue um número e eleve ao quadrado, caso ele fique "próximo"ao número $n$, significa que você encontrou a raiz quadrada.

Boa sorte!