

## **Lista de Exercícios 06**

### *Estrutura de Repetição em Java*

Cada aluno deverá, individualmente, resolver os exercícios abaixo.

#### **1. Impressão de 1 a 10**

**Descrição::** Escreva um programa que exiba na tela todos os números de 1 até 10.

#### **2. Soma de 1 a N**

**Descrição::** Leia um número inteiro N e calcule a soma dos números de 1 até N. Certifique-se de que o programa trate o caso de N negativo ou zero, neste caso, exiba a soma como 0 ( $N=0$ ) ou mostre uma mensagem apropriada ( $N < 0$ ).

#### **3. Tabuada de um número**

**Descrição::** Leia um número inteiro e exiba a tabuada desse número (do 1 ao 10).

#### **4. Média de valores lidos**

**Descrição::** Leia 5 números inteiros (um de cada vez) e calcule a média desses valores. Ao final, exiba a média na tela.

#### **5. Soma de ímpares entre 1 e 100**

**Descrição::** Calcule e exiba a soma de todos os números ímpares entre 1 e 100 (inclusive).

#### **6. Número invertido**

**Descrição::** Leia um número inteiro positivo e exiba o mesmo número invertido.

- **Exemplo:** Se o usuário digitar 1234, o programa deve exibir 4321.

## 7. Fatorial

**Descrição::** Leia um inteiro  $N$  e calcule o fatorial de  $N$ . Trate o caso de  $N = 0$  (fatorial de 0 é 1) e  $N$  negativo com uma mensagem apropriada.

## 8. Jogo de adivinhação simples

**Descrição::** O programa deve “sortear” mentalmente (definindo no próprio código) um número entre 1 e 10. Em seguida, peça para o usuário adivinhar o número. Enquanto o usuário não acertar, continue pedindo um palpite. Quando o usuário acertar, exiba uma mensagem de “Acertou!”.

## 9. Verificação de número primo

**Descrição::** Peça ao usuário para digitar um número inteiro positivo e verifique se ele é primo. Exiba uma mensagem indicando “É primo” ou “Não é primo”.

- **Dica:** Para verificar se um número  $N$  é primo, basta checar se há algum divisor entre 2 e  $N-1$  (ou mais eficientemente entre 2 e a raiz quadrada de  $N$ ). Se encontrar divisor, não é primo.

## 10. Contagem regressiva

**Descrição::** Leia um número inteiro  $N$  e faça uma contagem regressiva a partir de  $N$  até 0 na tela.

## 11. Menu de opções com repetição

**Descrição::** Implemente um pequeno menu textual que ofereça opções como:

1. Exibir “Olá!”
2. Exibir data (fixa, não precisa ser automática)
3. Sair

O programa deve continuar exibindo o menu até que a opção “3” seja escolhida.

## 12. Leitura até número negativo

**Descrição::** Leia números inteiros em um loop e mostre a soma de todos eles quando um número negativo for digitado. O número negativo não deve entrar na soma.

## 13. Soma dos dígitos

**Descrição::** Leia um número inteiro positivo e calcule a soma de seus dígitos.

- **Exemplo:** Se o usuário digitar 321, a soma dos dígitos é  $3 + 2 + 1 = 6$ .

## 14. Média de valores até zero

**Descrição::** Faça um programa que leia valores (um por vez) e calcule a média desses valores. A leitura deve continuar até que o valor 0 seja inserido, momento em que o programa deve parar e exibir a média dos valores positivos lidos.

## 15. Intervalo entre dois números

**Descrição::** Leia dois números inteiros **a** e **b** e exiba todos os números do intervalo entre **a** e **b** (inclusive). Considere também o caso em que **a** > **b**.

- **Dica:** Verifique qual é o menor e maior entre **a** e **b**.

## 16. Múltiplos de 3 até 100

**Descrição::** Escreva um programa que mostre todos os múltiplos de 3 de 1 até 100.

## 17. Divisores de um número

**Descrição::** Leia um número inteiro e exiba todos os seus divisores positivos.

- **Exemplo:** Se o número é 12, então seus divisores são 1, 2, 3, 4, 6, 12.

## 18. Contagem de dígitos

**Descrição::** Leia um número inteiro e conte quantos dígitos ele possui.

- **Exemplo:** Se for digitado 2023, o programa deve exibir 4.

## 19. Verificação de perfeito

**Descrição::** Leia um número inteiro e verifique se ele é um número perfeito. Um número perfeito é aquele cuja soma de seus divisores (excluindo ele mesmo) é igual ao próprio número (por exemplo, 6 e 28).

## 20. Sequência de Fibonacci (limitada)

**Descrição::** Leia um valor inteiro N e gere os primeiros N termos da sequência de Fibonacci. A sequência de Fibonacci é definida como:

1.  $F(0) = 0$ ,
2.  $F(1) = 1$ ,
3.  $F(n) = F(n-1) + F(n-2)$  para  $n \geq 2$

- **Exemplo:** Se  $N = 6$ , a saída deve ser 0 1 1 2 3 5.

## 21. Algoritmo de Collatz

**Enunciado:** Implemente o *Algoritmo de Collatz*. Dado um número inteiro positivo  $n$ , faça o seguinte:

- Se  $n$  for par, substitua-o por  $n / 2$ .
- Se  $n$  for ímpar, substitua-o por  $3 * n + 1$ . Repita esse processo até que  $n$  se torne 1, exibindo cada valor intermediário no caminho.

**Exemplo:** Se  $n = 6$ , a sequência gerada é:  $6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ .

- **Desafio:** Certifique-se de que o programa lida corretamente com  $n = 1$  (já inicia em 1, então não precisa iterar).

---

## 22. Conversão Decimal–Binário (sem arrays)

**Enunciado:** Leia um número inteiro positivo e converta-o para a representação binária, exibindo o resultado na tela **sem** usar arrays ou coleções.

- **Passos sugeridos:**
  - Use um laço para extrair os restos da divisão por 2 (usando  $\%$  2).
  - Construa uma string (por concatenação) que represente o binário **ao contrário** porque o primeiro resto corresponde ao bit menos significativo.
  - Ao final, inverta essa string para exibir o binário correto.
- **Dica:** Para inverter a string sem arrays, você pode:
  - Criar uma segunda string e concatenar caracter a caracter do final para o início da primeira string. ir construindo a string final na ordem certa desde o início, mas terá de pensar numa lógica de “empilhar” os bits, por exemplo, concatenando o bit mais recente no início da string em vez de no final.

**Exemplo:**

- Se o usuário digitar 13, o binário é 1101.

## 23. Palíndromo Numérico

**Enunciado:** Verifique se um número inteiro positivo é palíndromo **sem converter o número diretamente em uma string** (se quiser, pode usar a manipulação de string apenas para exibir mensagens, mas não para resolver a lógica principal).

- **Como resolver:**
  - Leia o número original em uma variável **original**.
  - Use um laço para inverter o número (pegar dígitos com % 10 e formar outro número).
  - Compare o número invertido com o original. Se forem iguais, é palíndromo; caso contrário, não é.
- **Exemplo:**
  - Se o número for 121, ao inverter obtemos 121, logo é palíndromo.
  - Se for 123, ao inverter obtemos 321, não é palíndromo.

## 24. Número de Armstrong (Geral)

**Descrição:** Um número é dito de *Armstrong* (ou *Narcisista*) se a soma de cada um dos seus dígitos elevados à quantidade de dígitos do número for igual a ele próprio. Por exemplo:

- 153 tem 3 dígitos, e  $1^3 + 5^3 + 3^3 = 153 = 153$ .
- 370, 371 e 407 também são exemplos de 3 dígitos.

Bom trabalho.