

## 1ª Lista de Exercícios

### *Estruturas de seleção e repetição*

1. Escreva um aplicativo em Java que solicite ao usuário inserir o tamanho do lado de um quadrado e então exiba um quadrado vazio desse tamanho feito de asteriscos. Seu programa deve trabalhar com quadrados de todos os comprimentos de lado possíveis entre 1 e 20. Caso o usuário escolha um tamanho inválido, uma mensagem de erro deve ser exibida.

2. Escreva um aplicativo que aceite como entrada um inteiro contendo somente 0s e 1s (isto é, um inteiro binário) e imprime seu equivalente decimal. Caso o usuário digite um número com dígitos diferentes de 0s e 1s, uma mensagem de erro deve ser exibida.

**Dica:** Utilize os operadores de resto e divisão para pegar os dígitos do número binário, um de cada vez, da direita para a esquerda. No sistema de números decimais, o dígito mais à direita tem um valor posicional de 1 e o próximo dígito à esquerda tem um valor posicional de 10, depois 100, depois 1.000 e assim por diante. O número decimal 234 pode ser interpretado como  $4 * 1 + 3 * 10 + 2 * 100$ . No sistema de números binários, o dígito mais à direita tem um valor posicional de 1, o próximo dígito à esquerda tem um valor posicional de 2, depois 4, depois 8 e assim por diante. O equivalente decimal do binário 1101 é  $1 * 1 + 0 * 2 + 1 * 4 + 1 * 8$ , ou  $1 + 0 + 4 + 8$  ou, 13.

3. Calcule o valor de  $\pi$  através da serie infinita

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} \dots$$

Imprima uma tabela que mostre o valor aproximado de  $\pi$  computando usando um termo dessa série, dois termos, três termos e a sim por diante. Quantos termos de série você tem que utilizar para que a aproximação obtida tenha precisão de duas casas decimais? E precisão de três casas decimais? Quatro casas? Cinco casas?

4. Escreva um aplicativo que aceite como entrada um inteiro positivo e determine se este é um número perfeito.

Um número inteiro é perfeito se é igual à soma de seus divisores próprios. Divisores próprios de um número positivo  $N$  são todos os divisores inteiros positivos de  $N$  exceto o próprio  $N$ .

Por exemplo, o número 6, tem divisores próprios 1, 2 e 3, cuja soma é igual à 6, ou seja,  $1 + 2 + 3 = 6$ . Logo 6 é um número perfeito.

5. Escreva um aplicativo de calculadora que receba um valor em ponto flutuante, um operador e a seguir mais um valor em ponto flutuante. Utilize o comando switch para realizar a conta para as operações de soma, subtração, multiplicação e divisão e exiba o resultado. Caso o operador seja inválido ou ocorra uma divisão por zero, uma mensagem de erro deve ser mostrada.

6. Escreva um método estático que receba uma cadeia de caracteres, representada por uma String, que armazena o caminho do diretório raiz até um arquivo específico.

**Exemplo:** C:\musicas\somewhereintime\themeofpaganini.mp3

O método deve retornar uma nova String contendo apenas o nome do arquivo sem o caminho de diretórios.

7. Um número  $a$  é dito *permutação* de um número  $b$  se os dígitos de  $a$  formam uma permutação dos dígitos de  $b$ . Obs.: Considere que o dígito 0 (zero) não aparece nos números. Exemplo: 5412434 é uma permutação de 4321445, mas não é uma permutação de 4312455.

Faça um programa que lê dois inteiros positivos  $a$  e  $b$  e responda se  $a$  é permutação de  $b$ .

8. Dada uma sequência de  $n$  números inteiros determinar o comprimento de um segmento crescente de comprimento máximo.

**Exemplos:** Na sequência 5, 10, 3, 2, 4, 7, 9, 8, 5 o comprimento do segmento crescente máximo é 4. Na sequência 10, 8, 7, 5, 2 o comprimento do segmento crescente máximo é 1.

9. Dado um número inteiro positivo  $n$ , determinar todos os inteiros entre 1 e  $n$  que são comprimentos das hipotenusas de triângulos retângulos com catetos inteiros.

**10.** O dígito verificador é um mecanismo de autenticação utilizado para verificar a validade de um valor numérico, evitando fraudes ou erros de transmissão/digitação. Consiste em um ou mais dígitos acrescentados ao valor original e calculados a partir deste através de um algoritmo. Números de documentos de identificação, de matrícula, cartões de crédito e quaisquer outros códigos numéricos, que necessitem de maior segurança, utilizam dígitos verificadores (**Fonte:** *Wikipedia*).

Uma das rotinas mais tradicionais para cálculo do dígito verificador é denominada *Módulo 11*, e funciona da seguinte forma: cada dígito do número, começando da direita para a esquerda (menos significativo para o mais significativo) é multiplicado, na ordem, por 2, por 3, por 4 e assim sucessivamente, até o limite de multiplicação escolhido.

2	6	1	5	3	3	-	9
x7	x6	x5	x4	x3	x2		
14	36	5	20	9	6		
14 + 36 + 5 + 20 + 9 + 6						=	(90 x 10) / 11 = 81, resto 9 => DV = 9

Escreva um programa que receba um número inteiro, juntamente com um dígito verificador. Calcule o dígito verificador do número usando a técnica descrita acima, considerando que o limite de multiplicação é igual a 9 (após multiplicação por 9, a multiplicação retorna a 2). O algoritmo deve imprimir uma mensagem, indicando se o número é válido ou não segundo o código.