Tecnologia em Análise e Desenvolvimento de Sistemas
Lógica de Programação
Professor Vinícius Fritzen Machado

INSTITUTO FEDERAL
Rio Grande do Sul
Campus Rio Grande

Avaliação de Aprendizagem II

Desenvolva 5 dos exercícios abaixo utilizando somente o que foi visto em sala de aula. Novas soluções são encorajadas, no entanto, é necessário que os alunos demonstrem domínio sobre as técnicas apresentadas. Os códigos fontes serão avaliados quanto a funcionalidade, legibilidade, estrutura e organização. Enviar os códigos fontes para o email

vinicius.machado+logica2025@riogrande.ifrs.edu.br

No Assunto, incluir seu Nome + sobrenome. Compacte os arquivos .java em .zip e renomeie o arquivo com seu nome. Entregue esta folha assinada e espera a confirmação de que o email chegou.

Boa avaliação!!!

1. Escreva um algoritmo para ler um valor entre 1 (inclusive) e 10 (inclusive). Se o valor lido não estiver entre 1 (inclusive) e 10 (inclusive), deve ser lido um novo valor. Após a leitura do valor, escrever o valor lido na tela

2. Faça um programa que, para um número indeterminado de pessoas: leia a idade de cada uma, sendo que a idade 0 (zero) indica o fim da leitura e não deve ser considerada. A seguir calcule e mostre
   • o número de pessoas;
   • a idade média do grupo;
   • a menor idade e a maior idade

3. Um número perfeito é aquele cuja soma de seus divisores é igual ao próprio número. Por exemplo, o número 6 que possui como divisores 1, 2, 3, e como 1+2+3=6. 6 é um número perfeito. Desenvolva um programa que calcule os números perfeitos no intervalo de 0 a 1000.

4. Crie um programa que calcule uma aproximação para o logaritmo natural de (1 + x), para valores de `x` entre `-1 < x ≤ 1`, usando `n` termos da série de Taylor:

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots + (-1)^{n+1} \cdot \frac{x^n}{n}$$

   Restrições:

- Solicite ao usuário os valores de `x` e `n`.
- Use somente operadores e laços
- Use `double` para variáveis de resultado.

- **Exemplo de entrada:**
  `x = 0.5`
  `n = 5`
  Saída esperada: `ln(1 + 0.5) ≈ 0.407…`

5. Implemente um programa que receba um número real positivo `x` e calcule sua raiz quadrada com aproximação usando o método de Newton-Raphson.

$$r_{n+1} = \frac{1}{2}\left(r_n + \frac{x}{r_n}\right)$$

Critério de parada:

- O cálculo deve parar quando a diferença entre r_n e r_{n+1} for menor que `0.0001`.
  Exemplo de entrada:
  `x = 25`
  Saída esperada: `Raiz aproximada: 5.0`

6. Implemente um programa que receba um número inteiro `N` (N > 0) e calcule uma aproximação para o número π usando a Série de Leibniz:.

$$\pi \approx 4 \cdot \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots \pm \frac{1}{(2N-1)}\right)$$

Regras:

- Use apenas operações aritméticas e laços
- Não utilize bibliotecas prontas como `Math.PI`
- O valor de N indica a quantidade de termos somados

Exemplo de entrada: `N = 10`
Exemplo de saída esperada: `Aproximação de PI: 3.0418396189`

7. Escreva um programa que leia um valor inteiro n > 0 e desenhe na tela o losango com asteriscos (*) que possua a quantidade de linhas n. Ex: n = 7

```
   *
  ***
 *****
*******
 *****
  ***
   *
```

8. Escreva um programa que leia um valor inteiro n > 0 e desenhe na tela o contorno de triângulo com asteriscos (*) que possua a quantidade de linhas n. Ex: n = 7

```
*
**
* *
*  *
*   *
*    *
*******
```