

Prova 1
Programação 2
13/08/2025

Aluno: _____

1. (2.0 pts) Escreva um programa que estime o valor da constante matemática e usando a fórmula abaixo. O número de termos da soma deve ser fornecido pelo usuário. Use o tipo `long` para o cálculo do fatorial.

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

2. (3.0 pts) Nesse exercício, você deve implementar o clássico jogo da velha bidimensional. Use um Array C++ 3x3 (matriz) para armazenar as jogadas. Cada jogador deve indicar suas jogadas inserindo um par de números inteiros representando os índices de linha e coluna do quadrado no qual deseja colocar sua marca, um 'X' ou um 'O'. Cada jogada deve ser para um quadrado vazio. Se a jogada for para um quadrado ocupado, o jogo deve solicitar outra posição para o jogador da vez, até que ele insira uma posição desocupada. Após cada movimento válido, o jogo deve verificar se alguém ganhou ou se houve empate.
3. (2.5 pts) Use um vector para armazenar uma lista de valores numéricos fornecidos pelo usuário. A lista pode ter qualquer tamanho. Implemente 3 funções, uma para calcular a média dos valores, uma para calcular a mediana dos valores (valor central após ordenação) e uma para calcular a moda dos valores (valor mais frequente). Lembre-se, vectors podem ser grandes, portanto faça a passagem da forma mais eficiente possível. Além disso, no caso da moda, lembre-se de que quando há empate entre valores, todos eles são considerados e, portanto, sua função deve retornar um vector com esses valores. Escreva o programa principal de teste.
4. (2.5 pts) Implemente um programa em C++ para o jogo da adivinhação. Faça o sorteio de um número randômico inteiro, que será o número secreto, na faixa de 1 a 100. Para isso, use as funcionalidades da biblioteca `<random>`. O número secreto deve ser uma variável global. Crie a função `int tentativa(int chute)` que compara o chute do usuário com o número secreto. Ela deve retornar 0 se o usuário acertou, 1 se o usuário chutou alto e -1 se o usuário chutou baixo. Use uma **variável estática** para contar quantas vezes a função já foi chamada. Informe se o chute foi correto ou não. Caso o jogador acerte, exiba também o total de tentativas usadas. Caso o jogador erre, informe se o chute foi alto ou baixo. O programa deve permitir que o usuário continue tentando até acertar.