CRISTINÁRIO II - HASKELL
06 DE MAIO DE 2025

1. Crie em Haskell uma função que receba dois números inteiros como parâmetros e retorne uma tupla contendo a soma, a diferença, o produto e o quociente (divisão inteira) desses dois números. Em seguida, imprima os resultados no main.
2. Crie uma expressão em Haskell que receba uma lista de pares de números inteiros, e calcule para cada par a soma e a diferença. Use map para aplicar essa operação a todos os pares e imprima os resultados como uma lista de tuplas.
3. Escreva uma função em Haskell que receba um número e retorne uma tupla com o seu quadrado e o seu cubo. Utilize essa função para calcular esses valores para o número 3.
4. Escreva uma função em Haskell que receba uma lista de números e retorne uma nova lista com o quadrado de cada elemento. Utilize essa função para calcular os quadrados dos números de 1 a 5.
5. Escreva uma função em Haskell que receba uma lista de raios e retorne uma lista com as áreas dos respectivos círculos, utilizando map. Teste a função com os raios [3, 5, 8].
6. Escreva uma função em Haskell que receba o raio de um círculo e retorne uma tupla com a área e a circunferência do círculo. Use as fórmulas:
    - Área = π * r²
    - Circunferência = 2 * π * r
    - Teste a função com um raio de 7.

7. Crie uma tupla que contenha o nome de um aluno, sua nota e um status ("Aprovado" ou "Reprovado"). Em seguida, escreva uma função que receba a tupla e retorne apenas o status do aluno.
8. Crie uma lista de tuplas, onde cada tupla contém o nome de um aluno e sua nota. Em seguida, escreva uma função que filtre apenas os nomes dos alunos que tiraram nota maior ou igual a 7 e os imprima.
9. **maiorDeTres :: Int -> Int -> Int -> Int.** Escreva uma função que receba três números inteiros e retorne o maior entre eles.
10. **somaQuadradosPares :: [Int] -> Int.** Crie uma função que receba uma lista de inteiros, filtre os pares, calcule o quadrado de cada um e retorne a soma desses quadrados.
11. **media :: [Float] -> Float.** Implemente uma função que calcule a média de uma lista de números reais (não vazia).
12. **potenciasDeDois :: Int -> [Int].** Escreva uma função que receba um número n e retorne uma lista com as potências de 2 de 0 até n.
13. **segundo :: [a] -> a.** Implemente uma função que retorne o segundo elemento de uma lista (assuma que ela tem pelo menos dois elementos).
14. **elementosIguais :: Eq a => [a] -> Bool.** Escreva uma função que verifica se todos os elementos de uma lista são iguais.
15. **divideMeio :: [a] -> ([a], [a]).** Crie uma função que divida uma lista ao meio (ou seja, retorne duas listas com tamanhos iguais ou quase iguais).
16. **zipComSoma :: [Int] -> [Int] -> [Int].** Escreva uma função que receba duas listas e some os elementos correspondentes (como o zipWith (+)).
17. **contaMaiorQue :: Int -> [Int] -> Int.** Crie uma função recursiva que conte quantos elementos da lista são maiores que um valor dado.
18. **replica :: Int -> a -> [a].** Implemente uma função que, dado um número n e

um elemento, retorne uma lista com n cópias desse elemento.

19. **mapeiaQuadrado :: [Int] -> [Int].** Crie uma função que aplique o quadrado a todos os elementos da lista (sem usar map).
20. **reverteComRecursao :: [a] -> [a].** Reescreva a função de inversão de lista usando recursão pura (sem reverse).
21. **fibonacci :: Int -> Int.** Escreva uma função recursiva que receba um número n e retorne o n-ésimo número da sequência de Fibonacci.
22. **fibonacciLista :: Int -> [Int].** Crie uma função que receba um número n e retorne uma lista com os n primeiros números da sequência de Fibonacci.
23. **fibonacciMenoresQue :: Int -> [Int].** Escreva uma função que gere todos os números da sequência de Fibonacci menores que um valor n dado como parâmetro.
24. **ehImpar :: Int -> Bool.** Escreva uma função que receba um número inteiro e retorne True se ele for ímpar, ou False caso contrário.
25. **filtraImpares :: [Int] -> [Int].** Crie uma função que receba uma lista de inteiros e retorne apenas os números ímpares da lista.
26. **somaImpares :: Int -> Int.** Escreva uma função que calcule a soma dos n primeiros números ímpares.

Obrigada por participar do Cristinário!! Infelizmente, você não ganhará um prêmio, mas talvez ganhe um trauma.