

Universidade Federal de Goiás
INF - Instituto de Informática

Disciplina: Tópicos - Programação Funcional
Curso: Ciências da Computação

Professor: Daniel Ventura
Data: 12/09/2024

Laboratório 3: Conceitos Básicos

1. Usando uma lista em compreensão, escreva uma expressão para calcular a soma $1^2 + 2^2 + \dots + 100^2$ dos quadrados dos inteiros de 1 a 100.
2. A constante matemática π pode ser aproximada usando expansão em *séries* (i.e. somas infinitas), como por exemplo:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \dots + \frac{(-1)^n}{2n+1} + \dots$$

- (a) Escreva uma função `aprox :: Int → Double` para aproximar π somando n parcelas da série acima (onde n é o argumento da função).
- (b) Defina uma outra função `aprox2` usando a seguinte expansão para π^2 :

$$\frac{\pi^2}{12} = 1 - \frac{1}{4} + \frac{1}{9} - \dots + \frac{(-1)^k}{(k+1)^2} + \dots$$

Compare os resultados obtidos somando 10, 100 e 1000 termos com a aproximação π pré-definida no prelúdio-padrão.

3. Defina uma função `divprop :: Int → [Int]` usando uma lista em compreensão para calcular a lista de *divisores próprios* de um inteiro positivo (i.e. inferiores ao número dado). Exemplo: `divprop 10 = [1, 2, 5]`.
4. Um inteiro positivo n diz-se *perfeito* se for igual à soma dos seus divisores próprios. Defina uma função `perfeitos :: Int → [Int]` que calcula a lista de todos os números perfeitos até um limite dado como argumento. Exemplo: `perfeitos 500 = [6, 28, 496]`.
5. O Algoritmo de Luhn, utilizado para verificar a validade do número de um cartão de crédito, é definido da seguinte forma.

Considerando cada dígito como um número a parte:

- (a) Da direita para a esquerda, multiplique por 2 os números de forma alternada, começando da segunda posição;
- (b) Subtraia 9 de cada número com valor maior do que 9;
- (c) Some os números resultantes;
- (d) Se o resultado final for divisível por 10, então o número é válido.

Dessa forma,

- (a) Defina a função `dobroLuhn :: Int → Int` que calcula o dobro de um número, subtraindo 9 se o resultado for maior do que 9.
- (b) Utilizando a função `dobroLuhn` definida anteriormente, defina a função `luhn :: Int → Int → Int → Int → Bool`, decidindo se um número com 4 dígitos é um cartão de crédito válido.