

Exercícios de Recursividade

- 1) computar o fatorial de um número inteiro positivo: def fat(n)
- 2) computar x^y : def potencia(x,y)
- 3) exibir uma sequência crescente de 1 até N: def sequencia_crescente(x):
- 4) exibir uma sequência decrescente de N até 1: def sequencia_decrescente (x):
- 5) determinar o maior elemento de um vetor(array): def maior_elem_vetor(vet, max):
- 6) determinar o soma dos elementos de um vetor: def soma_ate_n(vet, n):
- 7) Computar fibonacci de um número dado: def fib(n)
Fib(0) = 0, fib(1)=1, fib(n) para $n > 1 = \text{fib}(n-1) + \text{fib}(n-2)$
- 8) Calcular o mdc de x e y

$$\text{gcd}(x, y) = \begin{cases} x & \text{if } y = 0 \\ \text{gcd}(y, \text{remainder}(x, y)) & \text{if } x \geq y \text{ and } y > 0 \end{cases}$$

- 9) Computar o somatório dos elementos de 1 até N
- 10) Determinar o menor elemento de um vetor(array): def menor_elem_vetor(vet, max):
- 11)

Faça uma função recursiva que calcule o valor da série S descrita a seguir para um valor $n > 0$ a ser fornecido como parâmetro para a

$$\bullet S = 1 + 1/1! + 1/2! + 1/3! + 1/n!$$

12) Considere agora a função que calcula o n-ésimo número harmônico:

$$H(n) = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n \quad (n \geq 1)$$

Uma outra definição recursiva:

$$H(n) = 1 \text{ se } n \leq 1 \\ 1/n + H(n-1) \text{ se } n > 1$$

Usando a definição recursiva acima:

$$H(4) = 1/4 + H(3) = 1/4 + 1/3 + H(2) = 1/4 + 1/3 + 1/2 + H(1) = 1/4 + 1/3 + 1/2 + 1$$

Análogo ao fatorial, a função acima também tem o caso de parada (n igual a 1), onde o valor da função não é recorrente.