Exercícios: Recursão

1. Crie uma função recursiva que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N.

2. Faça uma função recursiva que calcule e retorne o fatorial de um número inteiro N.

3. Escreva uma função recursiva que calcule a soma dos primeiros n cubos: $S(n) = 1^3 + 2^3 + ... + n^3$

4. Crie uma função recursiva que receba dois inteiros positivos k e n e calcule k^n.

5. Faça uma função recursiva que calcule e retorne o N-ésimo termo da sequência Fibonacci. Alguns números desta sequência são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...

6. Faça uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem crescente.

7. Faça uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem decrescente.

8. Faça uma função recursiva que receba um número inteiro positivo par N e imprima todos os números pares de 0 até N em ordem crescente.

9. Faça uma função recursiva que receba um número inteiro positivo par N e imprima todos os números pares de 0 até N em ordem decrescente.

10. Escreva uma função recursiva que exibe todos os elementos em uma lista de inteiros, separados por espaço.

11. Crie um programa que contenha uma função recursiva para encontrar o menor elemento em uma lista.

12. 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 mesma.

$$S = 2 + \frac{5}{2} + \frac{10}{3} + ... + \frac{1+n^2}{n}$$

13. Faça uma função recursiva que receba um número inteiro positivo impar N e retorne o fatorial duplo desse número. O fatorial duplo é definido como o produto de todos os números naturais ímpares de 1 até algum número natural ímpar N. Assim, o fatorial duplo de 5 é

$$5!! = 1 * 3 * 5 = 15$$

14. Faça uma função recursiva que receba um número inteiro positivo N e retorne o fatorial quádruplo desse número. O fatorial quádruplo de um número N é dado por:

$$\frac{(2n)!}{n!}$$

15. Faça uma função recursiva que receba um número inteiro positivo N e retorne o superfatorial desse número. O superfatorial de um número N é definida pelo produto dos N primeiros fatoriais de N. Assim, o superfatorial de 4 é

$$sf(4) = 1! * 2! * 3! * 4! = 288$$

16. Faça uma função recursiva que receba um número inteiro positivo N e retorne o hiperfatorial desse número. O hiperfatorial de um número N, escrito H(n), é definido por

$$H(n) = \prod_{k=1}^{n} k^k = 1^1 * 2^2 * 3^3 ... (n-1)^{n-1} * n^n$$

17. Faça uma função recursiva que receba um número inteiro positivo N e retorne o fatorial exponencial desse número. Um fatorial exponencial é um inteiro positivo N elevado à potência de N-1, que por sua vez é elevado à potência de N-2 e assim em diante. Ou seja,

$$n^{(n-1)^{(n-2)^{\cdots}}}$$

18. Escreva uma função recursiva que calcule a sequência dada por:

```
F(1) = 1
F(2) = 2
F(n) = 2 * F(n-1) + 3 * F(n-2).
```

19. Uma sequência de Fibonacci generalizada, de f0 a f1 é definida como fibg(f0, f1, 0), fibg(f0, f1, 1), fibg(f0, f1, 2), ..., onde:

```
fibg(f0, f1, 0) = f0
fibg(f0, f1, 1) = f1
fibg(f0, f1, n) = fibg(f0, f1, n-1) + fibg(f0, f1, n-2), se n > 1.
```

20. Faça uma função recursiva que permita somar os elementos de uma lista de inteiros.

21. Faça uma função recursiva que receba um número N e retorne o N-ésimo termo da sequência de tribonacci. Os números tribonacci são definidos pela seguinte recursão

```
f(n) = 0 se n = 0
f(n) = 0 se n = 1
f(n) = 1 se n = 2
f(n) = f(n-1)+f(n-2)+f(n-3) se n > 3
```

22. Faça uma função recursiva que receba um número N e retorne o N-ésimo termo da sequência de tetranacci. Os números tetranacci iniciam com quatro termos pré-determinados e a partir daí todos os demais números são obtidos pela soma dos quatro números anteriores. Os primeiros números tetranacci são: 0, 0, 0, 1, 1, 2, 4, 8, 15, 29, 56, 108, 208...

23. Faça uma função recursiva que receba um número N e retorne o N-ésimo termo da sequência de Padovan. A sequência de Padovan é uma sequência de naturais P(n) definida pelos valores iniciais
`P(0) = P(1) = p(2) = 1`
e a seguinte relação recursiva
`P(n) = P(n - 2) + P(n - 3) se n > 3`
Alguns valores da sequência são: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28...

24. Implemente a função h definida recursivamente por:

```
h(m,n) = m+1 ,se n = 1
h(m,n) = n+1 ,se m = 1
h(m,n) = h(m,n-1)+h(m-1,n) ,se m>1,n>1
```

25. Faça uma função recursiva para computar a função de Ackerman. A função de Acherman é definida recursivamente nos números não negativos como segue:

```
A(m,n) =  n+1 se m = 0
A(m,n) =  A(m-1,1) se m > 0 e n = 0
A(m,n) =  A(m-1,A(m,n-1)) se m > 0 e n > 0
```

26. Faça uma função recursiva para calcular os números de Pell. Os números de Pell são definidos pela seguinte recursão

```
p(n) = 0 se n = 0
p(n) = 1 se n = 1
p(n) = 2p(n-1)+ p(n-2) se n > 1
```

27. Faça uma função recursiva para calcular os números de Catalan. Os números de Catalan são definidos pela seguinte recursão

    $C(n) = 1$ se n = 0
    $C(n) = \frac{2(2n-1)}{n+1}C(n-1)$ se $n > 0$

    Alguns números desta sequência são: 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786...

28. Uma palavra de Fibonacci é definida por

```
f(n) = b se n = 0
f(n) = a se n = 1
f(n) = f(n-1)+f(n-2) se n > 1
```

    Aqui o símbolo “+” denota a concatenação de duas strings. Esta sequência inicia com as seguintes palavras:
    b, a, ab, aba, abaab, abaababa, abaababaabaab, ...
    Faça uma função recursiva que receba um número N e retorne a N-ésima palavra de Fibonacci.

29. Dado um número n na base decimal, escreva uma função recursiva que converte este número para binário.

30. Escreva uma função recursiva que determine quantas vezes um dígito K ocorre em um número natural N. Por exemplo, o dígito 2 ocorre 3 vezes em 762021192.

31. O máximo divisor comum dos inteiros x e y é o maior inteiro que é divisível por x e y. Escreva uma função recursiva **mdc** que retorna o máximo divisor comum de x e y. O mdc de x e y é definido como segue: se y é igual a 0, então mdc(x,y) é x; caso contrário, mdc(x,y) é mdc (y, x%y), onde % é o operador resto.