

Lista de Exercícios – Recursão<sup>1</sup>  
Prof. André Luiz Moura

**Observações:**

- Resolver, pelos menos, 19 (dezenove) das 37 (trinta e sete) questões.
- A implementação de cada função resolvida deve seguir o formato abaixo:

```
#include <stdio.h>
...
// Protótipo da função
<tipo de retorno> <nome da função>(lista de argumentos formais);

int main()
{
    ...
    return 0;
}

// Implementação da função
<tipo de retorno> <nome da função>(lista de argumentos formais)
{
    ...
}
```

- O arquivo contendo a implementação de função e a chamada a ela deve obedecer ao seguinte formato: **q<número>.c**. Exemplo: **q09.c**.
- Compactar todos os arquivos com como segue: **ResListaRecursao.zip**

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:  $S(n) = 1^3 + 2^3 + \dots + n^3$
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 dessa sequência são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...
6. A multiplicação de dois números inteiros pode ser feita através de somas sucessivas. Proponha um algoritmo recursivo `Multip_Rec(n1,n2)` que calcule a multiplicação de dois inteiros.
7. Faça uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem crescente.

---

<sup>1</sup> Extraída de lista elaborada pelo Prof. André Backes (UFU).

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

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 crescente.

10. 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.

11. Escreva uma função recursiva que exibe todos os elementos em um array de inteiros, separados por espaço.

12. Crie um programa que contenha uma função recursiva para encontrar o menor elemento em um vetor.

13. Escreva uma função recursiva SomaSerie(i,j,k). Esta função devolve a soma da série de valores do intervalo [i,j], com incremento k.

14. Escreva uma função recursiva ImprimeSerie(i,j,k). Esta função imprime na tela a série de valores do intervalo [i,j], com incremento k.

15. 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.

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

16. Faça uma função recursiva que receba um número inteiro positivo ímpar 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

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

17. 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:

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

18. 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

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

19. 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

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

20. 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,

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

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

$$F(1) = 1$$

$$F(2) = 2$$

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

22. Uma sequência de Fibonacci generalizada, de  $f_0$  a  $f_1$  é definida como  $\text{fibg}(f_0, f_1, 0)$ ,

$$\text{fibg}(f_0, f_1, 1), \text{fibg}(f_0, f_1, 2), \dots, \text{onde:}$$

$$\text{fibg}(f_0, f_1, 0) = f_0$$

$$\text{fibg}(f_0, f_1, 1) = f_1$$

$$\text{fibg}(f_0, f_1, n) = \text{fibg}(f_0, f_1, n-1) + \text{fibg}(f_0, f_1, n-2), \text{ se } n > 1.$$

23. Faça uma função recursiva que permita somar os elementos de um vetor de inteiros.

24. 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

$$f(n) = 0 \text{ se } n = 0$$

$$f(n) = 0 \text{ se } n = 1$$

$$f(n) = 1 \text{ se } n = 2$$

$$f(n) = f(n-1) + f(n-2) + f(n-3) \text{ se } n > 3$$

25. 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...

26. 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

$$P(n) = P(n-2) + P(n-3) \text{ se } n > 3$$

Alguns valores da sequência são: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28...

27. Implemente a função  $h$  definida recursivamente por:

$$h(m, n) = m+1, \text{ se } n = 1$$

$$h(m, n) = n+1, \text{ se } m = 1$$

$$h(m, n) = h(m, n-1) + h(m-1, n), \text{ se } m > 1, n > 1$$

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

$$A(m, n) = n+1 \text{ se } m = 0$$

$$A(m, n) = A(m-1, 1) \text{ se } m > 0 \text{ e } n = 0$$

$$A(m, n) = A(m-1, A(m, n-1)) \text{ se } m > 0 \text{ e } n > 0$$

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

$$p(n) = 0 \text{ se } n = 0$$

$$p(n) = 1 \text{ se } n = 1$$

$$p(n) = 2p(n-1) + p(n-2) \text{ se } n > 1$$

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

$$C(n) = 1 \text{ se } n = 0$$

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

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

31. 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. Essa 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.

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

33. Crie um programa que receba um vetor de números reais com 100 elementos. Escreva uma função recursiva que inverta a ordem dos elementos presentes no vetor.

34. Faça uma função recursiva que permita inverter um número inteiro N. Ex.: 123 - 321

35. 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.

36. 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 retorne 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.

37. Escreva uma função recursiva que permita fazer a multiplicação à russa de 2 entradas.

A Multiplicação à russa consiste em:

(a) Escrever os números A e B, que se deseja multiplicar na parte superior das colunas.

(b) Dividir A por 2, sucessivamente, ignorando o resto até chegar à unidade, escrever os resultados da coluna A.

(c) Multiplicar B por 2 tantas vezes quantas se haja dividido A por 2, escrever os resultados sucessivos na coluna B.

(d) Somar todos os números da coluna B que estejam ao lado de um número ímpar da coluna A.

Exemplo:  $27 * 82$

A	B	Parcelas
27	82	82
13	164	164
6	328	-
3	656	656
1	1312	1312

**Soma = 2214**