



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lista de exercícios de programação em linguagem C

## Exercícios: Recursão

- 1) Faça uma função recursiva que calcule e retorne o fatorial de um número inteiro N.
- 2) 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...
- 3) Faça uma função recursiva que permita inverter um número inteiro N. Ex: 123 - 321
- 4) Faça uma função recursiva que permita somar os elementos de um vetor de inteiros.
- 5) Crie uma função recursiva que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N.
- 6) Crie um programa em C, que contenha uma função recursiva que receba dois inteiros positivos k e n e calcule  $k^n$ . Utilize apenas multiplicações. O programa principal deve solicitar ao usuário os valores de k e n e imprimir o resultado da chamada da função.
- 7) Crie um programa em C que receba um vetor de números reais com 100 elementos. Escreva uma função recursiva que inverta ordem dos elementos presentes no vetor.
- 8) 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 em C, 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  $\text{mdc}(x,y)$  é x; caso contrário,  $\text{mdc}(x,y)$  é  $\text{mdc}(y, x\%y)$ , onde % é o operador resto.
- 9) Crie uma função recursiva que receba um número inteiro positivo N e calcule o somatório dos números de 1 a N.
- 10) 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.
- 11) 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.
- 12) Faça uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem crescente.
- 13) Faça uma função recursiva que receba um número inteiro positivo N e imprima todos os números naturais de 0 até N em ordem decrescente.

14) 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.

15) 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.

16) A função fatorial duplo é definida como o produto de todos os números naturais ímpares de 1 até algum número natural ímpar N.

Assim, o fatorial duplo de 5 é

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

Faça uma função recursiva que receba um número inteiro positivo ímpar N e retorne o fatorial duplo desse número.

17) O fatorial quádruplo de um número N é dado por

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

Faça uma função recursiva que receba um número inteiro positivo N e retorne o fatorial quádruplo desse número.

18) O superfatorial de um número N é definida pelo produto dos N primeiros fatoriais de N. Assim, o superfatorial de 4 é

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

Faça uma função recursiva que receba um número inteiro positivo N e retorne o superfatorial desse número.

19) O hiperfatorial de um número N, escrito H(n), é definido por

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

Faça uma função recursiva que receba um número inteiro positivo N e retorne o hiperfatorial desse número.

20) 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,

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

Faça uma função recursiva que receba um número inteiro positivo N e retorne o fatorial exponencial desse número.

21) Os números tribonacci são definidos pela seguinte recursão

$$f(n) = \begin{cases} 0 & \text{se } n = 0 \\ 0 & \text{se } n = 1 \\ 1 & \text{se } n = 2 \\ f(n-1) + f(n-2) + f(n-3) & \text{se } n > 2 \end{cases}$$

Faça uma função recursiva que receba um número N e retorne o N-ésimo termo da sequência de tribonacci.

22) 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...

Faça uma função recursiva que receba um número N e retorne o N-ésimo termo da sequência de tetranacci.

23) 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

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

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

Faça uma função recursiva que receba um número N e retorne o N-ésimo termo da sequência de Padovan.

24) Os números de Pell são definidos pela seguinte recursão

$$p(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ 2p(n - 1) + p(n - 2) & \text{se } n > 2 \end{cases}$$

Alguns números desta sequência são: 0, 1, 2, 5, 12, 29, 70, 169, 408, 985...

Faça uma função recursiva que receba um número N e retorne o N-ésimo número de Pell.

25) Os números de Catalan são definidos pela seguinte recursão

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

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

Faça uma função recursiva que receba um número N e retorne o N-ésimo número de Catalan.

26) Uma palavra de Fibonacci é definida por

$$f(n) = \begin{cases} b & \text{se } n = 0 \\ a & \text{se } n = 1 \\ f(n - 1) + f(n - 2) & \text{se } n > 1 \end{cases}$$

Aqui + 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.

27) Desenvolva algoritmos recursivos para cada um dos seguintes problemas:

- a) Impressão de um número natural em base binária
- b) Multiplicação de dois números naturais através de somas consecutivas.
- c) Inversão de uma string.
- d) Gerador da sequência:

(a)  $F(1)=1$

(b)  $F(2)=2$

(c)  $F(n)=2 \cdot F(n-1)+3 \cdot F(n-2)$  <= fórmula Geral

Usando a fórmula podemos identificar que

$F(3)=2 \cdot F(2)+3 \cdot F(1)$

$F(3)=4+3=7$

(e) Verificar se uma palavra é palíndromo

(f) Busca sequencial em um vetor desordenado que retorna a posição da primeira ocorrência de um elemento procurado usando uma estratégia similar a da busca binária.

Dica: Se não é o elemento do meio, procure dos dois lados e retorne o menor dos índices encontrados. Se for o elemento do meio, continue a busca apenas do lado esquerdo, retornando a posição de lá se encontrar e o meio senão encontrar

28) 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:  $27 \times 82$

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

**Soma: 2214**

Programar em C um algoritmo recursivo que permita fazer a multiplicação a russa de 2 entradas;

29) Implemente, usando a linguagem C, a função h definida recursivamente por:

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

30) A função de Ackerman é definida recursivamente nos números não negativos como segue:

$$A(m, n) = \begin{cases} n + 1 & \text{se } m = 0 \\ A(m - 1, 1) & \text{se } m > 0 \text{ e } n = 0 \\ A(m - 1, A(m, n - 1)) & \text{se } m > 0 \text{ e } n > 0 \end{cases}$$

Faça uma função recursiva em C para computar a função de Ackerman.

31) Escreva, usando a linguagem C, uma função recursiva, SomaSerie(i,j,k: inteiro): inteiro, que devolva a soma da série de valores do intervalo [i,j], com incremento k.

32) Faça uma função recursiva, em linguagem C, que calcule o valor da série S descrita a seguir para um valor n>0 a ser fornecido como parâmetro para a mesma.

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

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

33) Crie um programa em C, que contenha uma função recursiva para encontrar o menor elemento em um vetor. A leitura dos elementos do vetor e impressão do menor elemento devem ser feitas no programa principal.

34) Escreva, usando a linguagem C, uma função recursiva, ImprimeSerie(i,j,k: inteiro), que imprime na tela a série de valores do intervalo [i,j], com incremento k.

35) Dado um número n na base decimal, escreva uma função recursiva em C que converte este número para binário.

36) Um palíndromo é uma string que é lida da mesma maneira da esquerda para a direita e da direita para a esquerda. Alguns exemplos de palíndromo são radar e a bola da loba (se os espaços forem ignorados) Escreva uma função recursiva que retorna 1 se a string armazenada no array for um palíndromo e 0, caso contrário. O método deve ignorar espaços e pontuação na string.

37) Uma matriz maze de 0s e 1s, de 10X10, representa um labirinto no qual um viajante precisa encontrar um caminho de maze[0][0] a maze[9][9]. O viajante pode passar de um quadrado para qualquer outro adjacente na mesma fileira ou coluna, mas não pode saltar quadrados nem se movimentar na diagonal. Além disso, o viajante não pode entrar num quadrado contendo um 1. maze[0][0] e maze[9][9] contêm 0s. Escreva uma rotina recursiva que aceite este labirinto maze e imprima uma mensagem informando a inexistência de um caminho através do labirinto, ou que imprima uma lista de posições representando um caminho de [0][0] a [9][9].

38) Escreva uma função recursiva que calcule a soma de dois números naturais, através de incrementos sucessivos (Ex.: 3 + 2 = ++(++3)).

39) Escreva uma função recursiva que calcule a multiplicação de dois números naturais, através de incrementos sucessivos

40) Escreva uma função recursiva que calcule a sequência dada por:

$$F(1) = 1$$

$$F(2) = 2$$

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

41) Escreva uma função recursiva que dado um número  $n$ , gere todas as possíveis combinações com as  $n$  primeiras letras do alfabeto. Ex.:  $n = 3$ . Resposta: ABC, ACB, BAC, BCA, CAB, CBA

42) Escreva uma função recursiva que gere todas as possíveis combinações para um jogo da MegaSena com 6 dezenas

43) Uma sequência de Fibonacci generalizada, de  $f_0$  a  $f_1$  é definida como  $\text{fibg}(f_0, f_1, 0)$ ,  $\text{fibg}(f_0, f_1, 1)$ ,  $\text{fibg}(f_0, f_1, 2)$ , ..., onde:

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

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

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

Escreva uma função recursiva em C para calcular  $\text{fibg}(f_0, f_1, n)$ .

44) Escreva uma função recursiva que calcule a soma dos primeiros  $n$  cubos:

$$S(n) = 1^3 + 2^3 + \dots + n^3$$

45) Escreva uma função recursiva que calcule a soma dos dígitos de um número inteiro. Por exemplo, se a entrada for 123, a saída deverá ser  $1+2+3 = 6$ .

46) Faça uma função recursiva que permita calcular a média um vetor de tamanho  $N$ .