

Universidade Federal de Uberlaˆndia - UFU Faculdade de Computac¸a˜o - FACOM

Lista de exerc´ıcios de programac¸a˜o em linguagem Python

Exerc´ıcios: Recursa˜o

1. Crie uma func¸a˜o recursiva que receba um nu´mero inteiro positivo N e calcule o somato´rio dos nu´meros de 1 a N.
2. Fac¸a uma func¸a˜o recursiva que calcule e retorne o fatorial de um nu´mero inteiro N.
3. Escreva uma func¸a˜o recursiva que calcule a soma dos primeiros *n* cubos: *S*(*n*) = 13 + 23 + *...* + *n*3
4. Crie uma func¸a˜o recursiva que receba dois inteiros positivos *k* e *n* e calcule *kn*.
5. Fac¸a uma func¸a˜o recursiva que calcule e retorne o N-e´simo termo da sequeˆncia Fibo- nacci. Alguns nu´meros desta sequeˆncia sa˜o: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...
6. A multiplicac¸a˜o de dois nu´meros inteiros pode ser feita atrave´s de somas sucessivas. Proponha um algoritmo recursivo Multip Rec(n1,n2) que calcule a multiplicac¸a˜o de dois inteiros.
7. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo N e imprima todos os nu´meros naturais de 0 ate´ N em ordem crescente.
8. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo N e imprima todos os nu´meros naturais de 0 ate´ N em ordem decrescente.
9. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo par N e imprima todos os nu´meros pares de 0 ate´ N em ordem crescente.
10. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo par N e imprima todos os nu´meros pares de 0 ate´ N em ordem decrescente.
11. Escreva uma func¸a˜o recursiva que exibe todos os elementos em um array de inteiros, separados por espac¸o.
12. Crie um programa que contenha uma func¸a˜o recursiva para encontrar o menor elemento em um vetor.
13. Escreva uma func¸a˜o recursiva SomaSerie(i,j,k). Esta func¸a˜o devolve a soma da se´rie de valores do intervalo [i,j], com incremento k.
14. Escreva uma func¸a˜o recursiva ImprimeSerie(i,j,k). Esta func¸a˜o imprime na tela a se´rie de valores do intervalo [i,j], com incremento k.
15. Fac¸a uma func¸a˜o recursiva que calcule o valor da se´rie S descrita a seguir para um valor

*n >* 0 a ser fornecido como paraˆmetro para a mesma.

*S* = 2 +

5 10

+

2 3

+ *...* +

1 + *n*2

*n*

1. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo impar N e retorne

o fatorial duplo desse nu´mero. O fatorial duplo e´ definido como o produto de todos os

nu´meros naturais ´ımpares de 1 ate´ algum nu´mero natural ´ımpar N. Assim, o fatorial duplo de 5 e´

5!! = 1 ∗ 3 ∗ 5 = 15

1. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo N e retorne o fatorial qua´druplo desse nu´mero. O fatorial qua´druplo de um nu´mero N e´ dado por:

(2*n*)!

*n*!

1. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo N e retorne o super-

fatorial desse nu´mero. O superfatorial de um nu´mero N e´ primeiros fatoriais de N. Assim, o superfatorial de 4 e´

*sf* (4) = 1! ∗ 2! ∗ 3! ∗ 4! = 288

definida pelo produto dos N

1. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo N e retorne o hiperfa- torial desse nu´mero. O hiperfatorial de um nu´mero N, escrito H(n), e´ definido por

*n*

Y

*H*(*n*) = *kk* = 11 ∗ 22 ∗ 33*...*(*n* − 1)*n−*1 ∗ *nn*

*k*=1

1. Fac¸a uma func¸a˜o recursiva que receba um nu´mero inteiro positivo N e retorne o fatorial

exponencial desse nu´mero. Um fatorial exponencial e´ um inteiro positivo N elevado a`

poteˆncia de N-1, que por sua vez e´ elevado a` poteˆncia de N-2 e assim em diante. Ou seja,

*n*(*n−*1)(*n−*2)*...*

1. Escreva uma func¸a˜o recursiva que calcule a sequeˆncia dada por:

F(1) = 1

F(2) = 2

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

1. Uma sequeˆncia de Fibonacci generalizada, de f0 a f1 e´ fibg(f0, f1, 1), fibg(f0, f1, 2), ..., onde:

definida como fibg(f0, f1, 0),

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.

1. Fac¸a uma func¸a˜o recursiva que permita somar os elementos de um vetor de inteiros.
2. Fac¸a uma func¸a˜o recursiva que receba um nu´mero N e retorne o N-e´simo termo da sequeˆncia de tribonacci. Os nu´meros tribonacci sa˜o definidos pela seguinte recursa˜o

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

1. Fac¸a uma func¸a˜o recursiva que receba um nu´mero N e retorne o N-e´simo termo da sequeˆncia de tetranacci. Os nu´meros tetranacci iniciam com quatro termos pre´-determinados e a partir da´ı todos os demais nu´meros sa˜o obtidos pela soma dos quatro nu´meros ante- riores. Os primeiros nu´meros tetranacci sa˜o: 0, 0, 0, 1, 1, 2, 4, 8, 15, 29, 56, 108, 208...
2. Fac¸a uma func¸a˜o recursiva que receba um nu´mero N e retorne o N-e´simo termo da sequeˆncia de Padovan. A sequeˆncia de Padovan e´ uma sequeˆncia de naturais P(n) de- finida pelos valores iniciais

P(0) = P(1) = p(2) = 1

e a seguinte relac¸a˜o recursiva

P(n) = P(n - 2) + P(n - 3) se n > 3

Alguns valores da sequeˆncia sa˜o: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16, 21, 28...

1. Implemente a func¸a˜o *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

1. Fac¸a uma func¸a˜o recursiva para computar a func¸a˜o de Ackerman. A func¸a˜o de Acherman e´ definida recursivamente nos nu´meros na˜o 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

1. Fac¸a uma func¸a˜o recursiva para calcular os nu´meros de Pell. Os nu´meros de Pell sa˜o definidos pela seguinte recursa˜o

p(n) = 0 se n = 0 p(n) = 1 se n = 1

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

1. Fac¸a uma func¸a˜o recursiva para calcular os nu´meros de Catalan. Os nu´meros de Cata- lan sa˜o definidos pela seguinte recursa˜o

*C*(*n*) = 1 se n = 0

*C*(*n*) = 2(2*n−*1) *C*(*n* − 1) se *n >* 0

*n*+1

Alguns nu´meros desta sequeˆncia sa˜o: 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796,

58786...

1. Uma palavra de Fibonacci e´ 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 s´ımbolo “+” denota a concatenac¸a˜o de duas strings. Esta sequeˆncia inicia com as seguintes palavras:

b, a, ab, aba, abaab, abaababa, abaababaabaab, ...

Fac¸a uma func¸a˜o recursiva que receba um nu´mero N e retorne a N-e´sima palavra de Fibonacci.

1. Dado um nu´mero n na base decimal, escreva uma func¸a˜o recursiva que converte este nu´mero para bina´rio.
2. Crie um programa que receba um vetor de nu´meros reais com 100 elementos. Escreva uma func¸a˜o recursiva que inverta a ordem dos elementos presentes no vetor.
3. Fac¸a uma func¸a˜o recursiva que permita inverter um nu´mero inteiro N. Ex: 123 - 321
4. Escreva uma func¸a˜o recursiva que determine quantas vezes um d´ıgito K ocorre em um nu´mero natural N. Por exemplo, o d´ıgito 2 ocorre 3 vezes em 762021192.
5. O ma´ximo divisor comum dos inteiros x e y e´

o maior inteiro que e´

divis´ıvel por x e y.

Escreva uma func¸a˜o recursiva **mdc** que retorna o ma´ximo divisor comum de x e y. O mdc de x e y e´ definido como segue: se y e´ igual a 0, enta˜o mdc(x,y) e´ x; caso contra´rio, mdc(x,y) e´ mdc (y, x%y), onde % e´ o operador resto.

1. Escreva uma func¸a˜o recursiva que permita fazer a multiplicac¸a˜o a` russa de 2 entradas. A Multiplicac¸a˜o a` russa consiste em:
   1. Escrever os nu´meros A e B, que se deseja multiplicar na parte superior das colunas.
   2. Dividir A por 2, sucessivamente, ignorando o resto ate´ chegar a` unidade, escrever os resultados da coluna A.
   3. Multiplicar B por 2 tantas vezes quantas se haja dividido A por 2, escrever os resul- tados sucessivos na coluna B.
   4. Somar todos os nu´meros da coluna B que estejam ao lado de um nu´mero ´ımpar 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**