

Os exercícios seguintes devem ser codificados em português funcional. As codificações nas linguagens Hope e Haskell são opcionais.

1 > Elabore uma função chamada $\text{min}(x, y)$ que receba como argumento o parâmetro que represente dois valores numéricos e apresente o menor valor entre os dois valores fornecidos.

2 > Elabore uma função chamada $\text{min3}(x, y, z)$ que receba como argumento o parâmetro que represente três valores numéricos e apresente o menor valor entre os três valores fornecidos. Para essa ação faça o efeito de composição a partir da função $\text{min}(x, y)$.

3 > Elabore uma função chamada $\text{somat}(n)$ que receba como argumento o parâmetro de um valor numérico inteiro e apresente o resultado do somatório de 1 até o valor fornecido. Se fornecido o valor 5 a função deve apresentar o resultado 15 (para $1 + 2 + 3 + 4 + 5$). Efetue a solução com o uso de recursividade simples a partir da definição de desvio condicional indireto.

4 > Elabore uma função chamada $\text{fat}(n)$ que receba como argumento o parâmetro de um valor numérico inteiro e apresente o resultado do fatorial deste valor. Se fornecido o valor 5 a função deve apresentar o resultado 120 (para $1 * 2 * 3 * 4 * 5$). Efetue a solução com o uso de recursividade simples a partir da definição de desvio condicional indireto.

5 > Elabore uma função chamada $\text{somat2}(n)$ que receba como argumento o parâmetro de um valor numérico inteiro e apresente o resultado do somatório de 1 até o valor fornecido. Efetue a solução com o uso de recursividade simples a partir da definição de desvio condicional direto.

6 > Elabore uma função chamada $\text{fat2}(n)$ que receba como argumento o parâmetro de um valor numérico inteiro e apresente o resultado do fatorial deste valor. Efetue a solução com o uso de recursividade simples a partir da definição de desvio condicional direto.

7 > Elabore uma função chamada $\text{fatduplo}(n)$ que receba como argumento o parâmetro de um valor numérico inteiro e apresente o resultado do duplo fatorial ou fatorial duplo do valor fornecido. O duplo fatorial é o produto dos números de 1 até o número limite fornecido de 2 em 2. O fatorial duplo de 7 é $7 * 5 * 3 * 1 = 105$. Use recursividade simples com desvio condicional indireto.

8 > Elabore uma função chamada $\text{fattriplo}(n)$ que receba como argumento o parâmetro de um valor numérico inteiro e apresente o resultado do fatorial triplo do valor fornecido de 1 até o número limite fornecido saltando de 3 em 3. O fatorial triplo de 9 é $9 * 6 * 3 = 162$. Use recursividade simples com desvio condicional indireto.

9 > Elabore uma função chamada $\text{somat3}(n)$ que receba como argumento o parâmetro de um valor numérico inteiro e apresente o resultado do somatório de 1 até o valor fornecido. Efetue a solução com o uso de recursividade de cauda a partir da definição de desvio condicional indireto.

10 > Elabore uma função chamada $\text{fat3}(n)$ que receba como argumento o parâmetro de um valor numérico inteiro e apresente o resultado do fatorial deste valor. Efetue a solução com o uso de recursividade de cauda a partir da definição de desvio condicional indireto.

11 > Elabore uma função chamada $\text{intervp}(m, n)$ que receba dois valores inteiros “m” e “n” e retorne o produto de todos os valores numéricos do intervalo especificado. Use recursividade simples com desvio condicional direto.

12 > Elabore uma função chamada $\text{intervs}(m, n)$ que receba dois valores inteiros “m” e “n” e retorne o somatório de todos os valores numéricos do intervalo especificado. Use recursividade simples com desvio condicional direto.

13 > Elabore uma função chamada $\text{mult}(x, y)$ que receba dois valores “x” e “y” e retorne a multiplicação desses valores obtida a partir da ação de recursividade simples com desvio condicional direto.

14 > Elabore uma função chamada $\text{potência_de_2}(i)$ que receba um valor “i” como argumento, o qual representa o índice da potência, e apresenta o valor 2 elevado ao índice de potência “i”. Se fornecido o valor de “i” como 1 a função deve retornar 2, caso contrário deve realizar o cálculo recursivo simples a partir de desvio condicional indireto.

2, se $i = 1$

$2 * \text{potencia_de_2}(n - 1)$, se $n > 1$

15 > Considerando o jogo “Torre de Hanói” escreva a função $\text{hanói}(n)$ que apresente a quantidade de movimentos necessários para realizar todas as jogadas com “n” discos. Por exemplo, se usados 3 discos serão necessários 7 movimentos. Use recursividade simples com desvio condicional indireto. Para a realização deste exercício considere a regra:

0, se $n = 0$

1, se $n = 1$

$2 * \text{hanói}(n - 1) + 1$, se $n > 1$

16 > Desenvolva uma função recursiva chamada $\text{série}(n)$ que calcule o n-ésimo termo da série definida por:

0, se $n = 0$

3, se $n = 1$

$3 * \text{série}(n - 1) - 2$, se $n > 1$

17 > Elabore uma função chamada $\text{negativo}(n)$ que receba como argumento um valor numérico natural positivo e retorne seu valor correspondente negativo. É sabido que para obter um valor negativo basta multiplicar o valor positivo por “-1”. No entanto, este exercício

não deve ser realizado por multiplicação, use subtração. Se o valor fornecido for negativo este deverá ser mantido como negativo.

18 > Dois números naturais são primos entre si, ou seja, são coprimos se o MDC (máximo divisor comum) entre eles for igual a 1. Assim sendo, defina uma função chamada `coprimo(x, y)` que retorne verdadeiro se os valores fornecidos forem coprimos, caso contrário a função deve retornar o resultado falso. Se fornecido os valores `coprimo(13, 27)` o resultado será verdadeiro, já os valores `coprimo(13, 26)` resulta em falso.

19 > Crie uma função `mmc(x, y)` para calcular o mínimo múltiplo comum de dois números inteiros. Se quiser operar mais valores a função poderá ser encadeada no estilo `mmc(x, mmc(y, z))`.

20 > Crie uma função `senal(x, y)` que retorne o valor -1 se "x" for menor que "y", que retorne o valor 1 se "x" maior que "y" ou que retorne 0 se "x" for igual a "y".

21 > Elabore uma função chamada `hms_tempo(h, m, s)` para realizar a conversão das horas, minutos e segundos em um valor serial de tempo. Se fornecido os valores `hms_tempo(09, 32, 50)` deverá ser retornado o valor serial 34370. Dica: para gerar valor serial de tempo é necessário trazer a segundos os tempos das horas e dos segundos. A função deverá operar as horas entre 0-23, os minutos e segundos entre 0-59. Para qualquer valor fornecido fora da faixa aceitável a função deve retornar mensagem de erro indicando "algum dado fornecido está incorreto".

22 > Elabore uma função chamada `tempo_hms(tmp)` para realizar a conversão do valor serial de tempo em horas, minutos e segundos. Se fornecido o valor `tempo_hms(34370)` deverá ser retornado os valores 9, 32 e 50. A função deverá operar o tempo serial correspondente as faixas de 0-23 para as horas e 0-59 para os minutos e segundos. Para qualquer valor fornecido fora da faixa aceitável a função deve retornar mensagem de erro indicando "valor serial fornecido está incorreto".

23 > Elabore uma função chamada `tempo_horário(começo, término)` para realizar o cálculo da duração de tempo em horas, minutos e segundos entre certa hora de entrada e de saída. Se o valor do começo da contagem de tempo for maior que o valor de término a função deve retornar mensagem informando "início da contagem de tempo é maior que o término".

24 > Elabore uma função chamada `binário(n)` para realizar a apresentação de um valor numérico inteiro no formato binário. O resultado em binário deve ser armazenado em uma lista. Se executada a operação `binário(10)` deverá ser apresentado [1,0,1,0].

25 > Elabore uma função chamada `div84(n)` que retorna verdadeiro caso o valor "n" ao ser dividido por 8 tenha seu resto de divisão igual a 4.

26 > Elabore uma função chamada `divx(n, d, r)` que retorna verdadeiro caso o valor do dividendo "n" ao ser dividido pelo divisor "d" tenha seu resto da divisão igual a 4.