



Lista de exercícios 6

- 1) Crie uma função que receba como parâmetro um número inteiro e devolve seu dobro.
- 2) Crie uma função que receba três números inteiros como parâmetros, representando horas, minutos e segundos, e os converta em segundos. Ex: 2h, 40min e 10s correspondem a 9.610s.
- 3) Crie uma função que receba como parâmetro um valor inteiro positivo e retorne a soma dos divisores desse valor.
- 4) Crie um programa que receba três valores (obrigatoriamente maiores do que zero), representando as medidas dos três lados de um triângulo. Elabore funções para:
 - determinar se esses lados formam um triângulo (sabe-se que, para ser triângulo, a medida de um lado qualquer deve ser inferior à soma das medidas dos outros dois);
 - determinar e mostrar o tipo de triângulo (equilátero, isósceles ou escaleno), caso as medidas formem um triângulo.

Todas as mensagens deverão ser mostradas no programa principal.

- 5) Faça um programa que receba a temperatura média de cada mês do ano e armazene-as em um vetor. O programa deverá calcular e mostrar a maior e a menor temperatura do ano, junto com o mês em que elas ocorreram

Obs: o mês deverá ser mostrado por extenso: 1 = janeiro, 2 = fevereiro, ...

Obs₂: Cada cálculo deve ser realizado e mostrado em uma função diferente.

- 6) Foi realizada uma pesquisa entre 15 habitantes de uma região. Os dados coletados de cada habitante foram: idade, sexo, salário e número de filhos. Faça uma função que leia esses dados armazenando-os em vetores. Depois crie funções que recebam esses vetores como parâmetros e retornem

- a média de salário entre os habitantes;
- a menor e a maior idade do grupo;
- a quantidade de mulheres com três filhos que recebem até R\$500,00

Obs: Cada cálculo deve ser realizado em uma função diferente.

- 7) A prefeitura de uma cidade fez uma pesquisa entre seus habitantes, coletando dados sobre o salário e o número de filhos. Faça uma função que leia esses dados para um número não determinado de pessoas armazenando-os. Depois escreva funções que utilizem esses dados para retornar

- a média de salário da população;
- a media do número de filhos;
- o maior salário
- o percentual de pessoas com salário inferior a R\$380,00.

Obs: Cada cálculo deve ser realizado em uma função diferente.

8) Escreva um programa que crie um vetor de inteiros positivos **V**. Em seguida, faça o que se pede:
a) Escreva uma função **show** que receba **V**, o número de elementos **n** e imprima, para cada posição do vetor, uma quantidade de asteriscos correspondente a cada posição do vetor.

Ex:

V = {3,0,20,15,0,10}

n = 6

show(**V**,**n**)

0 ***

1

2 *****

3 *****

4

5 *****

b) Escreva uma função **smooth** que receba **V** e retorne um vetor **W**, onde cada elemento de **W** pode ser dado pela fórmula a seguir:

$$w_i = \frac{v_{i-1} + 2v_i + v_{i+1}}{4}$$

c) Escreva um programa que utilize um vetor **V** com 30 posições, as funções anteriores, e em seguida peça para o usuário aplicar 3 valores inteiros em 3 posições diferentes. E por fim, permita o usuário aplicar as funções **show** e **smooth** sobre o vetor criado até que ele deseje encerrar o programa.

9) Seja dois vetores, **v** e **w**, e seus respectivos tamanhos, **n** e **m**. Os elementos de ambos os vetores devem estar obrigatoriamente ordenados. Escreva uma função **combinar** que receba esses dados e retorne um vetor **x** que possui todos os elementos de ambos os vetores, em ordem. Ex:

v = {1,3,5,8}, **n** = 4

w = {2,4,6,7,9}, **m** = 5

combinar(**v**,**n**,**w**,**m**) = {1,2,3,4,5,6,7,8,9}

10) Escreva uma função **split** que receba um vetor de inteiros **v** e um número **n** e que reposicione os elementos de **v** da seguinte forma: o primeiro elemento da lista deve ser colocado em um lugar de forma que todos os valores menores que ele são reposicionados para sua frente e todos os valores maiores são reposicionados para depois dele. Ex:

v = {5, 3, 1, 8, 7, 4, 6, 2, 9}, **n** = 9

split(**v**,**n**) = {3, 1, 4, 2, 5, 8, 7, 6, 9}

11) Escreva uma função recursiva que receba um número inteiro positivo **n** e calcule o somatório dos números de 1 até **n**.

12) Escreva uma função recursiva que receba dois inteiros positivos **k** e **n** e calcule **kⁿ** sem usar a biblioteca **math.h**.

13) Escreva um funções recursivas que recebam um número inteiro positivo **n** e:

a) imprima todos os números naturais de 0 até **n** em ordem crescente.

b) imprima todos os números naturais de 0 até **n** em ordem decrescente.

c) imprima todos os números pares de 0 até **n** em ordem crescente.

14) Na matemática, o número harmônico designado por **H(n)** define-se como sendo a soma da série harmônica:

$$H(n) = 1 + \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

Escreva uma função recursiva que leia um valor **n** inteiro positivo e calcule o valor de **H(n)**.

15) Escreva uma função recursiva que receba um número inteiro positivo n e escreva a sequência de fibonacci. Ex

$n = 10$

$\text{fibonacci}(n) = 1, 1, 2, 3, 5, 8, 13, 21, 34, 55$

16) Escreva uma função recursiva que receba um vetor de inteiros v e dois valores i e f e retorne a soma de todos os elementos do vetor v entre as posições i e f . Ex:

$v = \{2, 6, 4, 8, 1, 9, 20, 14, 6, 7, 0, 64\}$

$i = 3$

$f = 7$

$\text{somaIntervalo}(v, i, f) = 8 + 1 + 9 + 20 + 14 = 52$

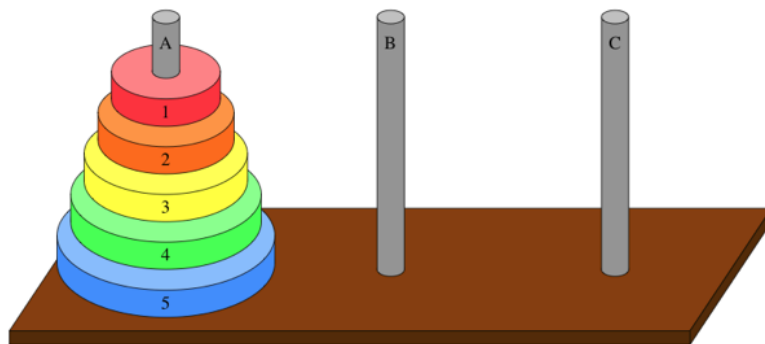
17) Escreva uma função recursiva que determine quantas vezes um dígito k ocorre em um número natural n . Ex:

$n = 762021192$

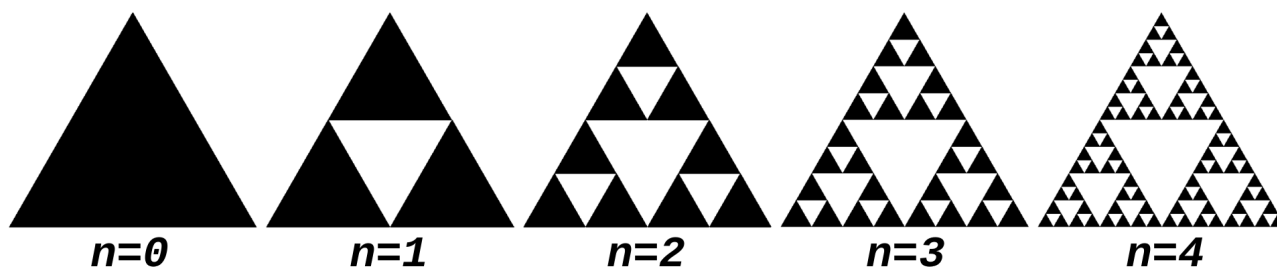
$k = 2$

$\text{contaDigito}(n, k) = 3$

18) A Torre de Hanói consiste em uma base contendo três pinos, em um dos quais são dispostos alguns discos uns sobre os outros, em ordem crescente de diâmetro, de cima para baixo. O problema consiste em passar todos os discos de um pino para outro qualquer, usando um dos pinos como auxiliar, de maneira que um disco maior nunca fique em cima de outro menor em nenhuma situação. Escreva uma função recursiva que escreva os movimentos necessários para mover uma Torre de Hanói com n discos do pino A para o pino B.



19) O Triângulo de Sierpinski é uma das formas elementares da geometria fractal. Trata-se de um conjunto autossimilar de um triângulo. Ao unir os pontos médios dos lados de um triângulo, dividimos o triângulo original em 4 triângulos semelhantes, com metade da base e metade da altura do triângulo original. O triângulo do meio não pertence ao fractal. Os demais são outros triângulos de Sierpinski por recursão. Ex: abaixo estão os triângulos de Sierpinski com zero a quatro níveis de recursão.



A área de um triângulo pode ser calculada por $(\text{base} \times \text{altura})/2$. Escreva uma função recursiva que receba como parâmetro a base, a altura e o número de níveis para calcular a área escura de um triângulo de Sierpinski.

20) O algoritmo de ordenação merge-sort é um exemplo de algoritmo dividir-para-conquistar. Sua ideia básica consiste em dividir o vetor em dois recursivamente e depois ir montando-os novamente de forma ordenada. Utilizando a função **combinar** construída na questão 9, escreva uma função que implemente o algoritmo merge-sort.

