

## Quinta Lista de Exercícios - Computational Thinking

1. Um número inteiro positivo  $n$  é denominado primo se existirem apenas dois divisores inteiros positivos dele: o 1 e o próprio  $n$ . Escreva uma função que recebe um inteiro  $n$  e retorna o valor True se  $n$  é primo ou False se ele não for primo.
2. Usando a função do exercício anterior, escreva um programa que imprime os 100 primeiros números primos começando do número 2.
3. Usando a função que verifica se um número é perfeito ou não, escreva um algoritmo que mostra todos os números perfeitos no intervalo de 1 a 50000 (cinquenta mil).
4. Um número  $a$  é dito permutação de um número  $b$  se os dígitos de  $a$  formam uma permutação dos dígitos de  $b$ . Exemplo: 5412434 é uma permutação de 4321445, mas não é uma permutação de 4312455. Obs.: Considere que o dígito 0 (zero) não aparece nos números.
  - a) Faça uma função contadigitos que dados um inteiro  $n$  e um inteiro  $d$ ,  $0 \leq d \leq 9$ , devolve quantas vezes o dígito  $d$  aparece em  $n$ .
  - b) Usando a função do item anterior, faça um programa que lê dois inteiros positivos  $a$  e  $b$  e responda se  $a$  é permutação de  $b$ .
5. Construa uma função encaixa que dados dois inteiros positivos  $a$  e  $b$  verifica se  $b$  corresponde aos últimos dígitos de  $a$ . veja um exemplo:

$a$	$b$	
567890	890	=> encaixa
1243	1243	=> encaixa
2457	245	=> não encaixa
457	2457	=> não encaixa

6. Usando a função do item anterior, faça um programa que lê dois inteiros positivos  $a$  e  $b$  e verifica se o menor deles é segmento do outro.

$a$	$b$	
567890	678	=> $b$ é segmento de $a$
1243	2212435	=> $a$ é segmento de $b$
235	236	=> um não é segmento do outro

7. Escreva uma função em Python que recebe três números inteiros positivos representando uma data (dia, mês e ano), sua função deverá retornar True se for uma data válida ou False, caso contrário. Faça alguns testes para verificar se sua função está correta ou não.
- 8.

Boa sorte!