

Quinta Lista de Exercícios - Computational Thinking

1. Escreva uma função que recebe dois números reais e retorna a média geométrica entre esses dois números.

```
1 def media_geometrica(a, b):  
2     return (a * b)**0.5
```

2. Escreva uma função que dado um inteiro positivo n , verifica se n é primo ou não.

```
1 def is_primo(n):  
2     if n == 1:  
3         return False  
4     else:  
5         div = 2  
6         while n % div != 0:  
7             div = div + 1  
8         if n == div:  
9             return True  
10        else:  
11            return False
```

3. 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.

```
1 def is_primo(n):  
2     contador = 0  
3     div = 1  
4     while div <= n:  
5         if n % div == 0:  
6             contador = contador + 1  
7             div = div + 1  
8  
9     if contador == 2:  
10        return True  
11    else:  
12        return False
```

4. Usando a função do exercício anterior, escreva um programa que imprime os 100 primeiros números primos começando do número 2.

```
1 contador = 0  
2 numero = 1  
3 while contador < 100:  
4     if is_primo(numero):  
5         print(numero)  
6         contador = contador + 1  
7     numero = numero + 1
```

5. 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).

```
1 def is_perfeito(n):  
2     soma = 0  
3     div = 1  
4     while div < n:  
5         if n % div == 0:  
6             soma = soma + div  
7             div = div + 1  
8     if soma == n:  
9         return True  
10    else:  
11        return False
```

6. 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 .

```

1 def contadigitos(numero, d):
2     contador = 0
3     while numero != 0:
4         resto = numero % 10
5         if resto == d:
6             contador = contador + 1
7
8         numero = numero // 10
9
10    return contador

```

- 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 .

```

1 a = int(input("Digite a:"))
2 b = int(input("Digite b:"))
3
4 i = 0
5 permutacao = True
6 while i < 10:
7     if contadigitos(a, i) != contadigitos(b, i):
8         permutacao = False
9     i = i + 1
10
11 if permutacao:
12     print("{} permuta o de {}".format(a, b))
13 else:
14     print("{} n o permuta o de {}".format(a, b))

```

7. Construa uma função `encaixa` que dados dois inteiros positivos a e b verifica se b corresponde aos últimos dígitos de a . veja um exemplo:

```

1 def encaixa(a, b):
2     pot = 1
3     while pot < b:
4         pot = pot * 10
5     #print(pot)
6     resto = a % pot
7     if resto == b:
8         return True
9     else:
10        return False

```

8. 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.

```

1 a = int(input("Digite a:"))
2 b = int(input("Digite b:"))
3
4 maior = a
5 menor = b
6 if a < b:
7     maior = b
8     menor = a
9

```

```
10 match = False
11 while maior > menor and not match:
12     if encaixa(maior, menor):
13         match = True
14
15 if match:
16     print("S o segmentos")
17 else:
18     print("N o s o segmentos")
```

9. 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.

```
1 def valida_data(dia, mes, ano):
2     dataValida = True
3
4     if dia < 1 or dia > 31 or mes < 1 or mes > 12: #DATAS INEXISTENTES
5         dataValida = False
6
7     if dia == 31 and mes == 4 or mes == 6 or mes == 9 or mes == 11:
8         dataValida = False
9
10    if mes == 2 and dia > 28:
11        if dia == 29:
12            if ano % 4 != 0:
13                dataValida = False
14            elif ano % 100 == 0 and ano % 400 != 0:
15                dataValida = False
16
17        else:
18            dataValida = False
19    return dataValida
```

Boa sorte!