

```

# coding: utf-8
import math

# -----laboratorio 7
-----#
''' 1. Supondo que a população de um país A seja da ordem de 80000 habitantes com
uma taxa anual de
crescimento de 3% e que a população de B seja 200000 habitantes com uma taxa de
crescimento de
1.5%. Faça uma função que calcule e retorne o número de anos necessários para que
a população do país
A ultrapasse ou iguale a população do país B, mantidas as taxas de crescimento.'''

def cresc_pop0():
    pop_A=80000
    pop_B=200000
    n=0

    while pop_A < pop_B:
        pop_A += pop_A * 0.03 # similar -> pop_A = pop_A + pop_A * 0.03
        pop_B += pop_B * 0.015 # similar -> pop_B = pop_B + pop_B * 0.03
        n+=1

    return "A populacao de A ultrapassa a de B em "+str(n)+" anos."

'''2. Altere a função anterior para receber como entrada as populações e as taxas
de crescimento iniciais dos
países A e B.'''

def cresc_pop(pop_A, pop_B, cresc_A, cresc_B):
    n=0

    while pop_A < pop_B:
        pop_A += pop_A * cresc_A # similar -> pop_A = pop_A + pop_A * cresc_A
        pop_B += pop_B * cresc_B # similar -> pop_B = pop_B + pop_B * cresc_B
        n+=1

    return "A populacao de A ultrapassa a de B em "+str(n)+" anos."

'''3. Faça uma função que simule um jogo de dois dados. A função deve contar
quantas vez o dado foi jogado
até que saiam números repetidos. Use a função randint do módulo random para
simular a jogada de um
dado.'''

import random

def dois_dados():
    n=0
    stop = False

    while stop == False: # similar -> while not stop ('not stop' significa 'stop
== False')
        n+=1
        dado1 = random.randint(1,6)
        dado2 = random.randint(1,6)

        if dado1 == dado2:
            stop == True

    return "Os dados foram rolados "+str(n)+" vezes ate os dois apresentarem o
valor "+str(dado1)

```

```
# -----laboratorio 8
# -----#

''' 1. Faça uma função que dada um número inteiro, retorne seu primeiro divisor
maior do que 1
'''

def primeiro_divisor(n):
    divisores = []

    for i in range(2, n+1):
        if n % i == 0:
            list.append(divisores, i)

    return divisores[0]

''' 2. Faça uma função que retorne a soma dos n primeiros números ímpares, onde n
é passado como parâmetro.'''

def soma_impares(n):
    soma = 0
    stop = n * 2

    for j in range(1, stop, 2):
        soma += j # similar a soma = soma + j

    return soma

''' 3. Faça uma função que calcule a soma dos fatoriais dos números inteiros de 1
a 10. '''
import math

def soma_fatoriais():
    summ=0

    for k in range(1,11):
        summ += math.factorial(k) # similar a summ = summ + math.factorial(k)

    return summ

''' 4. Sendo H = [...] , faça uma função para calcular e retornar o valor H com N
termos, onde N é inteiro e é dado como
entrada.'''

def calcula_H (n):
    H=0

    for i in range(1,n+1):
        H += 1.0/i

    return H
```

