

Exercícios (para serem entregues)

Agora é sua vez: faça os exercícios para praticar os conceitos vistos na aula sobre funções.

Lembre-se de **salvar** sempre o seu notebook. Ele deverá ser **entregue pelo tidia** para que você receba sua nota!

Questão 1

Escreva um programa que mostre os dois números primos mais próximos de um dado número n . Por exemplo, se $n = 24$, os dois números primos mais próximos dele são 23 e 29. Se $n = 47$, então os dois números primos mais próximos dele são 47 e 53.

Dica: primeiro faça uma função de nome `eh_primo` que recebe um inteiro x qualquer e retorna `True` se x for primo ou `False` caso contrário.

Entrada: O programa deve receber um inteiro $n > 2$.

Saída: A resposta consistirá de uma única linha contendo dois números inteiros separados por espaço onde o primeiro dos números é o maior número primo que é menor ou igual a n e o segundo dos números é o menor número primo que é maior do que n .

```
In [25]: def eh_primo(num):

    divisao = 0
    i = 2
    for i in range(2,num):
        if num%i == 0:
            divisao = divisao + 1

    if divisao == 0:
        return 1
    else:
        return 0

num = int(input("Digite um número inteiro positivo: "))

contador = num
while eh_primo != contador:
    contador-=1
antecessor = contador

contador = num + 1
while eh_primo != contador:
    for i in range(contador, -1):
        sucessor = contador

print("O antecessor é",antecessor,", e o sucessor é",sucessor)
```

File "<ipython-input-25-22f2b3ef0fe6>", line 24

sucessor = contador

^

IndentationError: expected an indented block

Questão 2

Faça um programa que teste quais números de uma sequência fazem parte da sequência de Fibonacci.

Dica: primeiro faça uma função de nome `eh_fibonacci` que recebe um único inteiro $n \geq 0$ e retorna `True` se n faz parte da sequência de Fibonacci ou `False` caso contrário.

Entrada: O programa deve receber inicialmente um inteiro n ($n \geq 0$). Em seguida, deve receber n números inteiros positivos.

Saída: A resposta consiste de uma única linha, com os números da sequência da entrada que fazem parte da sequência de Fibonacci separados por um espaço.

```
In [7]: def fibonacci(testar):
        if testar == (fibonacci(testar - 1) + fibonacci(testar - 2)):
            return testar
        else:
            return 0

quantidade = int(input("Informe a quantidade de números que serão testados: "))

for i in range(quantidade):
    testar = int(input("Digite um número: "))

    fibonacci(testar)

    print(fibonacci(testar))
```

Informe a quantidade de números que serão testados: 4

Digite um número: 4

```
-----
RecursionError                                Traceback (most recent call last)
<ipython-input-7-503328260201> in <module>()
     10     testar = int(input("Digite um número: "))
     11
--> 12     fibonacci(testar)
     13
     14     print(fibonacci(testar))

<ipython-input-7-503328260201> in fibonacci(testar)
      1 def fibonacci(testar):
--> 2     if testar == (fibonacci(testar - 1) + fibonacci(testar - 2)):
      3         return testar
      4     else:
      5         return 0

... last 1 frames repeated, from the frame below ...

<ipython-input-7-503328260201> in fibonacci(testar)
      1 def fibonacci(testar):
--> 2     if testar == (fibonacci(testar - 1) + fibonacci(testar - 2)):
      3         return testar
      4     else:
      5         return 0
```

RecursionError: maximum recursion depth exceeded

Questão 3

Dois números inteiros são primos entre si (ou coprimos) se o único divisor comum a eles é o número 1. Faça um programa que testa se dois números são coprimos.

Dica: primeiro faça uma função de nome `mdc` que recebe dois inteiros positivos e retorna o valor do máximo divisor comum entre eles. Você pode usar, por exemplo, o algoritmo de Euclides, ou qualquer outra forma que preferir (faça uma pesquisa).

Entrada: O programa deve receber dois valores inteiros e positivos.

Saída: A resposta consistirá de uma única linha contendo a mensagem `São coprimos` ou `Não são coprimos`, dependendo se os números forem coprimos ou não, respectivamente.

In []:

Questão 4

Faça um programa que imprima os n primeiros números primos.

Entrada: O programa deverá receber um único inteiro n , com $n \geq 0$.

Saída: A resposta consistirá em n linhas, cada uma contendo um número primo.

In [2]:

```
def primo(x):
    if x > 2 and x % 2 == 0:
        return False
    for i in range(3,x):
        if x%i == 0:
            return False
    return True

numero = int(input("Digite um número inteiro positivo: "))
if numero < 2:
    print("Só existem números primos maiores do que 1.")
for j in range(2, numero+1):
    if primo(j):
        print(j)
```

Digite um número inteiro positivo: 37
2
3
5
7
11
13
17
19
23
29
31
37

In []: