

## MAC0110 Introdução à Computação

### Exercício-Programa 3 (EP3) - Atualizado

Entregar até 22 de novembro de 2020

Escreva um programa em Python 3.x para resolver três problemas, de forma independente. Esses três problemas utilizam o Crivo de Eratóstenes, direta ou indiretamente.

#### Algoritmo do Crivo de Eratóstenes

No século três A.C., o matemático e astrônomo grego Eratóstenes desenvolveu um algoritmo para determinar todos os números primos até um dado número inteiro positivo  $n$ . Esse algoritmo é chamado de Crivo de Eratóstenes. (Implementações deste algoritmo foram dadas numa das aulas.)

Veja na página Crivo de Eratóstenes na Wikipédia uma animação desse algoritmo ([https://pt.wikipedia.org/wiki/Crivo\\_de\\_Eratóstenes](https://pt.wikipedia.org/wiki/Crivo_de_Eratóstenes)).

#### PROBLEMA 1:

Dado um inteiro positivo  $n$ ,  $n \geq 2$ , determinar todos os números primos menores ou iguais a  $n$ , utilizando o Crivo de Eratóstenes.

A sua solução deve utilizar as seguintes funções, cujos protótipos estão definidos mais adiante: `criaListaCrivoEratostenes`, `criaListaPrimos` e `imprimeNumerosInteirosLista`. Deve imprimir a quantidade de primos e os números primos encontrados.

#### PROBLEMA 2:

Dado um inteiro positivo  $n$ ,  $n > 2$ , encontrar uma sequência consecutiva mais longa de inteiros menores do que  $n$  sem nenhum número primo. Ou seja, determinar um par de números primos  $r$  e  $s$  tais que  $2 \leq r < s \leq n$ , o valor da diferença  $s - r$  é máximo e para todo inteiro  $i$  tal que  $r < i < s$ , tem-se que  $i$  não é primo. Basta encontrar um tal par (não precisa encontrar todos tais pares).

A sua solução deve utilizar a função `maiorIntervaloSemPrimos`, cujo protótipo está definido mais adiante e escrever uma mensagem clara sobre os primos  $r$  e  $s$  encontrados com relação a  $n$  e a quantidade de números não primos entre  $r$  e  $s$ .

#### PROBLEMA 3:

Em 1742, Christian Goldbach conjecturou que todo número par maior do que 2 pode ser escrito como soma de dois números primos. Por exemplo,  $16 = 3 + 13$ .

A Conjectura de Goldbach ainda não foi resolvida, mas sabe-se que ela é verdadeira para todo inteiro par  $n$  tal que  $n < 10^{14}$ .

Dado um inteiro  $k > 2$ , verificar se a Conjectura de Goldbach é verdadeira para todo inteiro par  $n$  tal que  $2 < n < k$ . Para isto, para cada tal inteiro  $n$ , encontrar dois números primos  $p$  e  $q$  tais que  $n = p + q$ . Veja mais adiante especificações mais detalhadas a respeito de  $p$  e  $q$ .

A sua solução deve utilizar a função `testaConjecturaGoldbach`, cujo protótipo está definido mais adiante. Esta função fornece também um certificado (para cada  $n$ , fornece o valor de  $p$ , onde  $p, q$  é um par de primos tais que  $n = p + q$ ).

## Funções a serem implementadas

Implemente em seu programa, obrigatoriamente, todas as funções cujos protótipos estão descritos a seguir, sem nenhuma alteração, e sem alterar a ordem de definição das funções. Não defina em seu programa nenhuma outra função além dessas obrigatórias.

```
def main():
    """ ( ) -> NoneType
    ...
    """

def resolverProblema1():
    """ ( ) -> NoneType
    ...
    """

def resolverProblema2():
    """ ( ) -> NoneType
    ...
    """

def resolverProblema3():
    """ ( ) -> NoneType
    ...
    """

def criaListaCrivoEratostenes(n):
    """ (int) -> list

    Recebe um inteiro  $n \geq 2$  e cria uma lista crivo[0...n] com zeros e uns
    tal que para cada  $i$ ,  $0 \leq i \leq n$ , crivo[i] é 1 se  $i$  é primo e crivo[i] é 0
    se  $i$  não é primo.
    A lista crivo é criada implementando o algoritmo do Crivo de Eratóstenes.
    Esta função retorna a lista crivo.
    """

def criaListaPrimos(crivo):
    """ (list) -> list

    Recebe uma lista crivo que foi criada utilizando o algoritmo do Crivo de
    Eratóstenes. A partir da lista crivo, esta função cria e retorna uma lista
    chamada primos, contendo todos os números primos, em ordem crescente.
    """

def imprimeNumerosInteirosLista(a):
    """ (list) -> NoneType

    Recebe uma lista a de números inteiros e imprime todos os números da lista,
    escrevendo no máximo dez números em cada linha e de modo que fiquem
    ajustados nas colunas.
    """
```

```
def maiorIntervaloSemPrimos(n):
    """ (int) -> int, int

    Recebe um inteiro  $n > 2$  e determina um par de números primos  $r$  e  $s$ 
    tais que  $2 \leq r < s \leq n$ , o valor  $s-r$  é máximo e para todo  $i$  tal que
     $r < i < s$ , tem-se que  $i$  não é primo (ou seja, entre  $r$  e  $s$  não há nenhum
    primo).
    Esta função retorna os primos  $r$  e  $s$ .
    """

    Sugestão: utilize a função criaListaPrimos para determinar um tal par  $r, s$ .
```

  

```
def testaConjecturaGoldbach(k):
    """ (int) -> list, bool

    Recebe um inteiro  $k > 2$  e verifica se a Conjectura de Goldbach é verdadeira
    para todo inteiro  $n$  par,  $2 < n < k$ .
    Para isto, para cada tal inteiro  $n$ , esta função tenta encontrar dois
    números primos  $p$  e  $q$  tais que  $n = p + q$ . Se existir mais do que um tal par,
    escolha o par com o menor  $p$  (e tal que  $p \leq q$ ).

    Obs.: Para alguns números pode existir mais de um par de primos.
    Por exemplo,  $40 = 3 + 37 = 11 + 29 = 17 + 23$ .
    Neste caso, o par escolhido deve ser  $p, q$ , com  $p = 3$  e  $q = 37$ .

    Para dar um certificado da validade da conjectura para os números pares  $n$ 
    entre 2 e  $k$ , o programa constrói uma lista chamada pares_primos que
    tem a seguinte propriedade:

    Como  $n$  é par,  $n \geq 4$ , então  $n = 2 * i$ , onde  $i \geq 2$ .
    Para cada  $i$ ,  $i \geq 2$ , pares_primos[i] armazena o primo  $p$  tal que  $n = p + q$ ,
    onde  $p \leq q$  e  $q = n - p$  é primo. Apenas o valor de  $p$  é armazenado já
    que o valor de  $q$  é precisamente  $n - p$ .

    A função testaConjecturaGoldbach retorna a lista pares_primos e retorna
    também True ou False dependendo se a conjectura for válida ou não para
    todo  $n$  par,  $2 < n < k$ .
    """

    Sugestão: para encontrar um tal par  $p, q$ , utilize a lista de números
    primos construída e retornada pela função criaListaPrimos.

    A lista que vai receber a lista pares_primos retornada pela função
    testaConjecturaGoldbach deverá ser utilizada para imprimir, para todo
     $n$  par tal que  $2 < n < k$ , os pares  $p$  e  $q$  tais que  $n = p + q$ .

    Obs.: Sabe-se que a Conjectura de Goldbach é verdadeira para todo inteiro
     $n$  par,  $2 < n < 10^{14}$ . Assim, para tais valores de  $n$ , o seu programa
    deve sempre conseguir encontrar primos  $p$  e  $q$  tais que  $n = p + q$ .
```

## **Saída do programa**

(Dentro de alguns dias serão dados exemplos de saída.)

(Comecem a fazer o EP3 sem se preocupar agora com a saída.)