**LABORATORIO**

**Hora prevista**

15-20 minutos

**Nivel de dificultad**

Medio

**Objetivos**

* familiarizar al alumno con nociones y algoritmos clásicos;
* mejorando las habilidades del estudiante para definir y usar funciones.

**Guión**

*Un número natural es****primo****si es mayor que 1 y no tiene divisores distintos a 1 y a sí mismo.*

¿Complicado? De ningún modo. Por ejemplo, 8 no es un número primo, ya que puede dividirlo entre 2 y 4 (no podemos usar divisores iguales a 1 y 8, ya que la definición lo prohíbe).

Por otro lado, 7 es un número primo, ya que no podemos encontrar divisores legales para él.

Su tarea es escribir una función comprobando si un número es primo o no.

La función:

* se llama isPrime;
* toma un argumento (el valor para verificar)
* devuelve True si el argumento es un número primo, y de lo False contrario.

Sugerencia: intente dividir el argumento entre todos los valores subsiguientes (a partir de 2) y verifique el resto; si es cero, su número no puede ser primo; piense detenidamente cuándo debe detener el proceso.

Si necesita conocer la raíz cuadrada de cualquier valor, puede utilizar el \*\*operador. Recuerde: la raíz cuadrada de x es la misma que x 0.5

Complete el código en el editor.

Ejecute su código y verifique si su salida es la misma que la nuestra.

**Datos de prueba**

Resultado esperado:

2 3 5 7 11 13 17 19

def isPrime(num):

#

# Su codigo Aqui

#

for i in range(1, 20):

if isPrime(i + 1):

print(i + 1, end=" ")

print()