

Obtención de números primos

Desarrollar un programa en Python que solicite al usuario una cantidad de números primos a agregar a un archivo preexistente denominado primos.txt, y agregue al final de este, respetando sus disposiciones en 5 columnas de 15 espacios, la cantidad de números primos indicada en sucesores del último que haya en el archivo.

La idea es que el programa se pueda ejecutar cuantas veces se desee, y que en cada ejecución se incremente la cantidad total de números primos registrados la cantidad pedida por el usuario.

Para la primera ejecución del programa el archivo primos.txt debe estar creado con un editor de texto cualquiera conteniendo al menos los 3 primeros números primos en columnas de 15 espacios:

primos.txt

2	3	5
---	---	---

RESTRICCIONES

En el prólogo del programa se debe abrir el archivo para leer, contar la cantidad de líneas que contiene y, de la última línea, obtener el último número primo y la cantidad de números en esa línea, ya que en el desarrollo del programa se deben agregar los primos que existan a continuación del último, siempre respetando que en cada línea que no sea la última haya registrados exactamente 5 (y puede suceder que la última línea del archivo tenga menos de 5).

Para encontrar los primos siguientes al último del archivo, se debe testear cada entero positivo sucesor que se obtenga a partir de la fórmula $6*n \pm 1$, siguiendo la conjetura de que para todo número primo p mayor que 3 existe un entero positivo n tal que $p=6*n-1$ o $p=6*n+1$, buscando divisores que cumplan la misma condición (conjunto que contiene a los primos con los que se puede factorizar) hasta el primero que supere a la parte entera de la raíz cuadrada del número que se testea.

Para agregar números al archivo, éste se debe abrir

```
arch=open('primos.txt', 'a')
```

Y para escribir un entero positivo que se testeó verificó primo, se debe escribir

```
arch.write(f'{p:15d}')
```

Tener en cuenta que la escritura en un archivo de texto no agrega por defecto un salto de línea luego de escribir, por lo que se deberá llevar cuenta de la cantidad de números de cada línea a efectos de que, luego de escribir el quinto número en la línea actual, se escriba `arch.write('\n')`, es decir, un salto de línea.

Se recomienda definir funciones para testear si un número es primo o no, y para, dado un entero positivo p , expresable como $6*n-1$ o $6*n+1$, devuelva a su sucesor, es decir

- para $p=6*n-1$ devuelva $6*(n+1)-1$
- para $p=6*n+1$ devuelva $6*(n+1)+1$