

Escriba un programa en Python que imprima si un número es primo o no. El algoritmo sugerido para saber si un número es primo es el siguiente:

- Dado un número n , verificar si es divisible entre $2, 3, 4, \dots \dots \sqrt{n}$
- Si es divisible entre algún número dentro de ese intervalo, no es primo; caso contrario sí es primo.

a) (2 puntos) Escriba una función que reciba como parámetro de entrada el argumento n y retorne *True* si el número es primo, o *False* en caso contrario (Implementación serial)

Imprima el tiempo de ejecución para $n = 2\ 345\ 678\ 911\ 111\ 111$

b) (5 puntos) Escriba una función que permita paralelizar el cálculo de si un número es primo o no. Use 2 procesos.

Imprima el tiempo de ejecución de esta función para el mismo valor de n de la parte a). Calcule el Speedup. Use la función `assert()` para verificar que el resultado en la parte a) sea igual a la parte b)

c) (3 puntos) Calcule el tiempo de ejecución para cuando se usa 4, 8, y 16 procesos. Grafique el número de procesos (eje x) vs el tiempo de ejecución (eje y). ¿Aumentar el número de procesos hace que disminuya el tiempo de ejecución, o no necesariamente?

Aumentar el número de procesos no hace que el tiempo disminuya necesariamente, solo hasta con 4 puede ser que disminuya, pero en el resto incrementa considerablemente, esto se debe a la cantidad de cores que tiene nuestra pc, si le mandamos más procesos que la cantidad de cores que tiene, un solo core estará ejecutando varios procesos que ocupan memoria y lo hará más lento.

Nota: Puede hacer el gráfico en Excel o hacerlo a mano y tomarle foto, pero debe subir la imagen del gráfico para que se le considere puntaje.