1. Una partícula realiza un movimiento vertical bajo la influencia de la gravedad de modo que su altura $z(t)$ respecto al suelo es dada por la siguiente ecuación de la trayectoria,

$$z(t) = z_0 + v_0 t - \frac{1}{2} g t^2, \tag{1}$$

con $g = 9.8$ m/s^2. Considere el caso en que $z_0 = 1$ m y $v_0 = 24$ m/s.

Escriba un programa en Python que, usando un ciclo `for`, calcule e imprima el valor de la altura $z(t)$ para los siguientes valores de tiempo (en segundos): $t = 0, 0.1, 0.2, \ldots, 5.0$ (51 valores distintos de tiempo).

2. Modifique el programa anterior, para que ahora éste pregunte al usuario los valores de z_0 y v_0. Para esto, use el comando `input` que aprendió en su trabajo con la guía 08.

3. El factorial de un número entero positivo n, denotado por $n!$ es definido por

$$n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n. \tag{2}$$

Por ejemplo, $3! = 1 \cdot 2 \cdot 3 = 6$ y $10! = 3628800$. Escriba un programa en Python que pregunte al usuario por el valor de n, y que calcule e imprima su factorial, es decir, $n!$.

4. Modifique ahora el código que creó para calcular el factorial de un número entero para que ahora su programa verifique, antes de calcular el factorial, que el número suministrado es realmente un entero positivo, y sólo calcule el factorial en ese caso, y que en caso contrario informe al usuario que el número ingresado no es apropiado.

5. Escriba un programa que pregunte al usuario el valor de algún número natural e imprima todos los números primos que hay hasta ese número. Por ejemplo, si se ingresa el número 8, el programa debe imprimir los números 2, 3, 5 y 7.

6. Escriba un programa que imprima en pantalla si un número entero positivo ingresado desde el teclado es o no un número primo. Utilice un ciclo `while`.

7. Considere una pelota que se lanza verticalmente desde una altura inicial $z_0 = 3\,\mathrm{m}$, con una velocidad inicial $v_0 = 1\,\mathrm{m/s}$ hacia arriba. Calcule e imprima en pantalla los valores de la altura

$$z(t) = z_0 + v_0 t - \frac{g}{2} t^2, \tag{3}$$

y el correspondiente tiempo t para $t = 0, 0.01, 0.02, \ldots$ hasta (justo antes) que la pelota llegue al suelo (es decir, llegue a $z = 0$). Utilice un ciclo `while` y considere $g = 9.8\,\mathrm{m/s^2}$.

8. Vea este video disponible en Canvas, en el que se explica la definición de **funciones** en Python. Reproduzca y verifique todos los ejemplos ahí descritos.

9. Modifique el código que creó para calcular el factorial de un número entero (guía 09) para que ahora se defina una función `mifactorial`, de modo que el factorial de n se pueda luego llamar como `mifactorial(n)`.

10. La exponencial e^x de un número real x puede ser calculada usando la siguiente serie

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \tag{4}$$

Reutilizando su código para calcular factoriales, escriba un programa que pregunte al usuario el valor de x y calcule e imprima el valor de e^x, usando la expresión (4). Para el cálculo considere 100 términos en la suma (4), es decir, que el programa calcule la suma hasta el término $x^{99}/99!$.

Nota 1: En el caso $n = 0$, se define el factorial de 0 igual al valor 1, es decir, $0! := 1$.

Nota 2: El "truncar" la serie (es decir, evaluarla hasta cierto número de términos) tiene como consecuencia que el valor calculado es sólo una *aproximación* del valor exacto (e^x). Esta aproximación es mejor si se incluyen más términos.

11. Utilice el programa que acaba de escribir para calcular (una aproximación d)el valor de e (el número de Euler). Compare su resultado con el valor listado en este artículo de wikipedia.

12. Escriba un programa que evalúe (una aproximación de) el número π. Para esto, use la siguiente expresión en serie (desarrollada por el gran Leonhard Euler),

$$\pi = \sum_{n=0}^{\infty} \frac{2^{n+1}(n!)^2}{(2n+1)!} = \left[2^1 \frac{(0!)^2}{1!} + 2^2 \frac{(1!)^2}{3!} + 2^3 \frac{(2!)^2}{5!} + 2^4 \frac{(3!)^2}{7!} + \cdots \right]. \tag{5}$$

Lo anterior es un ejemplo de un método con el que se puede calcular el valor de π, con precisión cada vez mayor al agregar más y más términos. Dado que π es un número irracional, sólo se conoce su valor (calculado con métodos similares) hasta un cierto número de decimales. El record actual lo tiene Shigeru Kondo, quien logró calcular π con 10000000000000 decimales!. Compare el valor que usted obtenido con el listado en este artículo de wikipedia.