Fundamentos de la Programación
(Grado en Física - 2021/2022)

Trabajo de evaluación continua puntuable para la calificación final de la asignatura, con una ponderación del 5 %

Fecha de entrega: Hasta el domingo 17 de octubre de 2021 a las 23:59.
Modo de entrega: A través de la tarea disponible en ModdleUA. Se podrá entregar tantas veces como se desee, sólo se corregirá la última entrega. Se debe entregar un único archivo comprimido `p1.tgz` o `p1.zip` creado en Linux con todos los ejercicios respetando para cada uno de ellos el nombre del enunciado.

Ejercicios:

1. (**p1-1.py**) Escribe un programa en Python que lea tres números enteros y devuelva el mayor y el menor de ellos.
   Ejemplos:

Entrada	**Salida**

`Introduce un número: 2`
`Introduce un número: -1`
`Introduce un número: 4`
	`El mayor es 4.`
	`El menor es -1.`

2. (**p1-2.py**) Un número perfecto es un número natural que es igual a la suma de sus divisores propios positivos (consideramos que 1 es un divisor propio, pero el propio número no lo es). Así, 6 es un número perfecto porque sus divisores propios son 1, 2 y 3; y $6 = 1 + 2 + 3$. Escribe un programa que lea un número natural $n > 0$ y decida si es perfecto o no.

   Ejemplos:

Entrada	**Salida**
`Introduce un número: 6`	`6 es un número perfecto.`
`Introduce un número: 12`	`12 no es un número perfecto.`

3. (**p1-3.py**) Escribe un programa que funcione como una calculadora de bolsillo. El programa debe leer una expresión compuesta por números (como no se sabe si van a ser enteros o reales, se debe suponer que son reales), y operadores aritméticos, que termina con el carácter "`=`", y debe escribir los resultados de cada operación. Se supone (no es necesario comprobarlo) que la expresión va a ser correcta y que no va a tener espacios en blanco. Los operadores que pueden aparecer son los cuatro operadores aritméticos: "`+`", "`-`", "`*`" y "`/`". Ten en cuenta que la división debe ser siempre real.

   Ejemplos:

Entrada	Salida
2 + 3.5 =	5.5
2 + 3.5 * 4 =	5.5 22.0
5 / 2 =	2.5
2 + 3.5 * 4 - 1.7 =	5.5 22.0 20.3

4. (**p1-4.py**) Escribe un programa que lea un número natural $n > 0$ y muestre todos sus divisores primos, exceptuando el 1 y el propio número, separados por comas y finalizando en punto. Si no encuentra ninguno deberá mostrar el mensaje "El número n es primo.".

   Ejemplos:

Entrada	Salida
`Introduce un número: 90`	`2, 3, 5.`
`Introduce un número: 13`	`El número 13 es primo.`

5. (**p1-5.py**) Escribe un programa que lea un número natural natural $n > 0$ y a continuación escriba en una línea, separados por comas y terminando en punto, los n primeros números primos cuya última cifra (por la derecha) no sea un 7.
   Ejemplo:

Entrada	Salida
8	`2, 3, 5, 11, 13, 19, 23, 29.`

6. (**p1-6.py**) Escribe un programa que lea un número natural natural $n > 0$ y a continuación lea n números enteros y escriba la media aritmética del

cuadrado de dichos números con dos cifras decimales.

Ejemplo:

Entrada	**Salida**

`Introduce la cantidad de números a leer: 4`
`Introduce el número 1: 7`
`Introduce el número 2: 1`
`Introduce el número 3: 55`
`Introduce el número 4: 2`
	`769.25`

7. (**p1-7.py**) Escribe un programa que vaya pidiendo números positivos y menores o iguales que 100 al usuario y, cuando éste termine (para lo cual escribirá `0`), imprima la media geométrica de todos ellos con tres cifras decimales (sin incluir el `0`). Las números que no cumplan la condición $0 < x \leq 100$ se deben ignorar para calcular la media geométrica (incluyendo el `0`, obviamente). La media geométrica se calcula con la expresión

$$\bar{x} = \sqrt[N]{\prod_{i=1}^{N} a_i} = \sqrt[N]{a_1 a_2 \dots a_N}$$

Ejemplo:

Entrada	**Salida**

`Introduce un número: 80`
`Introduce un número: 75`
`Introduce un número: 11.2`
`Introduce un número: 101`
`Introduce un número: 2.5`
`Introduce un número: 23.5`
`Introduce un número: 161`
`Introduce un número: -7`
`Introduce un número: 0`
	`20.858`

8. (**p1-8.py**) Escribe un programa que lea un número natural $n > 0$ y escriba una *figura* de números como la del ejemplo. Ten en cuenta que no hay espacios en blanco entre los números.
Ejemplo:

Entrada	Salida
5	543212345 4321234 32123 212 1 212 32123 4321234 543212345

9. (**p1-9.py**) La sucesión de *Fibonnaci* es una serie infinita de números naturales donde cada elemento $n \geq 1$ se obtiene de la siguiente forma:

$$f_n = \begin{cases} n-1 & n \leq 2 \\ f_{n-1} + f_{n-2} & n > 2 \end{cases}$$

Escribe un programa que lea un número natural $n > 0$ y a continuación escriba los n primeros números de dicha serie, separados por comas y terminando en punto.

Ejemplo:

Entrada	Salida
Introduce un número: 1 Introduce un número: 5 Introduce un número: 9	0. 0, 1, 1, 2, 3. 0, 1, 1, 2, 3, 5, 8, 13, 21.

10. (**p1-10.py**) Se puede demostrar (lo veréis en breve en la asignatura de Análisis) que

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

Implementa un programa que pida al usuario un número real y cálcule e^x mediante la serie anterior. El cálculo debe parar cuando la diferencia entre el valor actual y el anterior sea menor que 10^{-6}. El resultado se mostrará, por tanto, con 6 cifras decimales. **Ejemplo:**

Entrada	Salida
5	148.413159