

Fundamentos de la Programación (Grado en Física, 2021/22)

Práctica 3

Fecha límite para la entrega: 11/01/2022

La práctica 3 se deberá entregar antes de las 23:59 del día 11 de enero de 2022 a través de la tarea creada al efecto en la plataforma MoodleUA.

Se podrá entregar tantas veces como se desee. Sólo se corregirá la última entrega.

Hay que entregar un único archivo comprimido p3.tgz o p3.zip que contenga el código de todos los ejercicios.

Puesto que se realizará una precorrección automática de la práctica, se deben respetar escrupulosamente tanto el formato de entrada como el de salida indicados en el enunciado.

- (p3-1.py) Diseña un programa que lea una lista de números enteros separados por espacios en blanco de un archivo, cuyo nombre se pasará por línea de comandos, los transforme en un array de enteros de numpy e imprima por pantalla:
 - La suma de todos los números leídos.
 - El mínimo de todos ellos.
 - El resultado de sumar el cuadrado de todos ellos.

Debes utilizar funciones de numpy para realizar los cálculos.

Ejemplo:

Entrada	Salida
1 2 3 1 4	11 1 31

- (p3-2.py) Escribe un programa que abra un archivo de texto cuyo nombre será pasado por línea de comandos e imprima por pantalla el número de líneas, el número de palabras y el número de caracteres (incluidos espacios y cambios de línea) de dicho archivo¹ en el formato:

```
lineas: <número de líneas>
palabras: <número de palabras>
caracteres: <número de caracteres>
```

- (p3-3.py) Escribe un programa que lea carácter a carácter un archivo de texto cuyo nombre será leído del teclado con el mensaje 'Nombre del archivo: '. El programa sustituirá cada carácter leído, excepto si es un espacio en blanco de cualquier tipo, por el siguiente carácter en orden ASCII. El carácter transformado será enviado a otro archivo con el nombre obtenido añadiendo al de lectura (antes de la extensión²) la palabra '_cod'.

Se debe tener en cuenta la posibilidad de que el archivo de entrada o el de salida no se puedan abrir emitiendo en este caso el mensaje **Problema con el fichero de entrada o de salida**.

Ejemplo: si el archivo de entrada se llama `entrada.txt` y su contenido es

```
El otro día 24 de septiembre
tuvimos clase de
    Fundamentos de la Programación
```

El archivo de salida se llamará `entrada_cod.txt` y contendrá

```
Fm pusp e1b 35 ef tfqujfnscf
uvwjnpt dmbtf ef
Gvoebnfoupt ef mb Qsphsbnbdj0o
```

¹Existe un comando en Unix llamado `wc` que realiza dicha tarea.

²Se considera que la extensión (si la hubiera) comienza después del primer punto.

4. (**p3-4.py**) Escribe un programa que resuelva la ecuación diferencial $y''(x) + \sin(y(x)) + \sin(x) = 0$ con las condiciones iniciales $y(-10) = 0$, $y'(-10) = \pi$ en el intervalo $[-10, 10]$. Muestra el resultado mediante una gráfica guardada con el nombre **grafica.png** (no se debe mostrar la gráfica por pantalla).
5. (**p3-5.py**) calcular la integral:

$$\int_0^{\infty} \left(\frac{1}{1+x} - e^{-x} \right) \frac{dx}{x}$$

mostrando el resultado por pantalla³.

6. (**p3-6.py**)⁴ Diseña un programa que lea un vector de componentes enteras, un entero p que llamaremos posición, un entero l que llamaremos longitud y construya un nuevo vector con el trozo del vector original que empieza en la posición p y tiene longitud l . Antes, el programa debe comprobar $0 \leq p \leq n-1$, (n es el tamaño del vector introducido), y que $p+l \leq n$. Si no se cumple alguna de estas dos condiciones el programa debe escribir “error”. La entrada y la salida se leerán/escribirán en sendos archivos cuyos nombres serán pasados por línea de comandos. Se puede suponer que ambos archivos estarán siempre disponibles y, en el caso del de entrada, en formato correcto.

Ejemplos:

Entrada	Salida
5 7 8 2 4 3 1 7 1	error
5 7 8 2 4 3 1 6 2	error
5 7 8 2 4 3 1 2 3	8 2 4
5 7 8 2 4 3 1 0 6	5 7 8 2 4 3

7. (**p3-7.py**) Sin utilizar bucles (for o while) implementa la función

$$f(\vec{x}) = \sum_{i=1}^{N-1} 100(x_{i+1} - x_{i-1}^2)^2 + (1 - x_i)^2$$

donde $\vec{x} = (x_0, x_1, x_2, \dots, x_N)$.

Ayuda: debes usar cortes en arrays de numpy y funciones y operadores asociados a esa librería.

8. (**p3-8.py**) Nos pasan el siguiente código que define la clase Conjunto

```
class Conjunto(object):

    def __init__(self, lista): # Constructor de la clase
        self.elementos = []    # atributo de instancia
        for x in lista:
            if x not in self.elementos:
                self.elementos.append(x)

    def union(self, A): # métodos
        return Conjunto(self.elementos+A.elementos)
```

³El resultado de esta integral es la constante de Euler $\gamma = 0,57721 \dots$

⁴Este ejercicio es similar al p2-4.py, con la única diferencia de que los datos se leerán/escribirán en archivos de texto.

```

def interseccion(self, A):
    aux = []
    for x in self.elementos:
        if x in A.elementos:
            aux.append(x)
    return Conjunto(aux)

def cardinal(self):
    return len(self.elementos)

def __str__(self):
    # Este método especial define cómo se imprimirá un Conjunto en una
    # llamada a la función print
    cad = '{'
    for x in self.elementos[:-1]:
        cad += str(x) + ', '
    return cad + str(self.elementos[-1]) + '}'

```

Añade a esta clase dos métodos **diferencia** y **difSimetrica** que realice estas operaciones entre el conjunto desde el que se las llama y el conjunto que se le pasa por parámetro. En el programa principal se deben crear los conjuntos $A = \{1, 2, 3, 4\}$ y $B = \{0, 2, 3, 5, 6\}$, imprimiendo a continuación por pantalla $A - B$ y $A \Delta B$.

9. (**p3-9.py**) Escribir un programa que guarde en un diccionario los precios de las frutas de la tabla, pregunte al usuario por una fruta y un número de kilos y muestre por pantalla el precio de ese número de kilos de fruta. Si la fruta no está en el diccionario debe mostrar un mensaje informando de ello.

Fruta	Precio
Plátano	1.35
Manzana	0.80
Pera	0.85
Naranja	0.70