

Fundamentos de la Programación - Grado en Física - Curso 2020/2021

Examen C4 (12/07/2021)

Instrucciones

- Tienes **2 horas y 30 minutos** para realizar el examen. Lee tranquilamente el examen y decide por dónde empezar.
- No puedes comunicarte con nadie (excepto con el profesor) durante el examen.
- **Importante:** Pon tu DNI, nombre y apellidos al principio de cada ejercicio que entregues.
- Debes comprimir tu examen con todos los ejercicios que hayas realizado en un archivo **zip** cuyo nombre debe ser tu DNI (**NUMERO_DNI.zip**).
- Debes entregar dicho archivo comprimido a través de la tarea creada en MoodleUA, donde has encontrado este enunciado. Dicha aplicación se cerrará automáticamente una vez transcurrido el tiempo del examen, de modo que debes tener cuidado en no retrasarte.

Ejercicios

1. (ej1.py) (**2 puntos**) Escribe un programa que reciba dos cadenas por la línea de órdenes y muestre el número de veces que la segunda cadena está presente en la primera¹. El programa mostrará dicho valor y nada más. Se entiende que las minúsculas y mayúsculas son caracteres distintos.

Ejemplos:

```
$ python ej1.py "esto es un examen" "es"
2
$ python ej1.py "esto es un examen" "es un exa"
1
$ python ej1.py "esto es un examen" "e"
4
$ python ej1.py "esto es un examen" "esun"
0
```

2. (ej2.py) (**2 puntos**) La sucesión de *Fibonacci* 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 reciba un número natural $n > 0$ por la línea de órdenes y muestre la suma de los n primeros números de dicha serie (y nada más).

Ejemplos:

```
$ python3 ej2.py 6
12
$ python3 ej2.py 2
1
$ python3 ej2.py 1
0
$ python3 ej2.py 50
203650111073
```

3. (ej3.py) (**2 puntos**) Se dispone de un fichero de texto cuyo contenido son números enteros dispuestos en una única línea y separados mediante espacios en blanco. Escribe un programa que muestre por pantalla el valor que más veces se repite en ese fichero cuyo nombre se proporcionará en la línea de órdenes. El programa no debe mostrar nada más que ese valor.

Ejemplo: Si el contenido del fichero `numeros.txt` es

¹El método `find` del tipo cadena es útil para este ejercicio. En el apartado 5.1.11 del libro de referencia se explica su uso y funcionamiento.

1 1 0 1 2 3 3 3 3 5 6 0 1 1 1 3 2 2 1 1 1 2 2 2 2 1 1 1 -4 1

se tendrá

```
$ python ej3.py numeros.txt
13
```

4. (ej4.py) **(2 puntos)** El método de la secante para encontrar la solución de una ecuación $f(x) = 0$ es un método iterativo que en cada paso actualiza el valor de la aproximación a la solución mediante la expresión:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)$$

El método acaba cuando $|x_{n+1} - x_n| < \varepsilon$, donde $\varepsilon > 0$ es el error permitido².

Implementa una función llamada `metodoSecante(f, a, b, error)` que devuelva la solución de la ecuación $f(x) = 0$ entre los valores a y b con una precisión menor que `error`³.

A continuación escribe un programa que haciendo uso de la función implementada resuelva la ecuación $x^3 + 3x^2 - 6x - 5 = 0$ en el intervalo $[1, 10]$ con un error menor de 10^{-6} .

El programa debe mostrar la raíz de dicha ecuación en dicho intervalo, que en este caso es única, y nada más. (Nota: Para simplificar, tanto la ecuación a resolver, como el intervalo y el error permitido forman parte del código, es decir, no hay que solicitarlo por teclado).

5. (ej5.py) **(2 puntos)** Utiliza las librerías `numpy` y `matplotlib.pyplot` y el método `quad` de la librería `scipy.integrate` para dibujar la gráfica de la función

$$f(x) = \int_0^x \frac{u}{e^{-u}} du$$

entre 0 y 5 con 100 puntos.

²Suponemos que la solución es única, no hace falta comprobarlo.

³Ayuda: que la solución es única, no hace falta comprobarlo. toma como valores iniciales $x_0 = a$ y $x_1 = b$.