

Fundamentos de la Programación. Grado en Física - Curso 2018/2019

Examen de teoría de la convocatoria de julio (08/07/2019)

Instrucciones

1. Tienes **2 horas y 30 minutos** para realizar el examen. Lee tranquilamente el enunciado y decide por dónde empezar.
2. No puedes utilizar apuntes ni programas ya escritos por ti o por otros para hacer el examen. Tampoco puedes comunicarte con nadie (excepto con el profesor) durante el examen. Puedes consultar el material disponible en UACloud (libro y programas de ejemplo utilizados por el profesor en clase).
3. **Importante:** Pon tu DNI, nombre y apellidos al principio de cada ejercicio que entregues.
4. Finalmente debes entregar tu examen con todos los ejercicios que hayas realizado en un archivo comprimido en **zip** o en **tar.gz** cuyo nombre debe ser tu DNI (**NUMERO_DNI.zip** o **NUMERO_DNI.tar.gz**).
5. Debes entregar dicho archivo comprimido a través de la aplicación de UACloud → Evaluación → Entrega de Prácticas. Dicha aplicación se cerrará automáticamente a la hora fijada para la finalización del examen, de modo que debes tener cuidado en no retrasarte.

Ejercicios

1. (ej1.py) (**2 puntos**) Implementa una función `nomultiplos(num, n1, n2)` que devuelva una lista con todos los números naturales menores o iguales que `num`, que no sean múltiplos ni de `n1` ni de `n2`. En caso de que algún dato sea incorrecto (negativo o cero), devolverá `None`.

Por ejemplo, al invocar a `nomultiplos(24, 3, 7)`, el valor devuelto será la lista (en notación Python): `[1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20, 22, 23]`.

A continuación crea otra función `printLista(lista)` que imprima dicha lista en el formato `1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20, 22, 23`.

Si la lista es vacía se debe imprimir el mensaje

No hay ningún número que cumpla las especificaciones.

Si la función `nomultiplos` devuelve `None` se debe imprimir el mensaje

Error en los parámetros.

No se debe entregar un programa principal, aunque deberías probar tus funciones con uno.

2. (ej2.py) (**2 puntos**) La *secuencia Collatz* de un número entero se construye de la siguiente forma:
 - si el número es par, se divide por dos;
 - si es impar, se le multiplica por tres y se le suma uno;
 - la sucesión termina al llegar a uno.

Por ejemplo, la secuencia de Collatz de 3 es `{3, 10, 5, 16, 8, 4, 2, 1}`.

Implementa una función `secCollatz(n)` que muestre por pantalla tantas líneas como enteros hay entre 1 y `n`. Cada línea debe contener el número de línea y, separado por un espacio en blanco, el número de elementos de su secuencia Collatz.

Ejemplo: la llamada a `secCollatz(5)` mostrará por pantalla:

```
1 1
2 2
3 8
4 3
5 6
```

3. (ej3.py) (**2 puntos**) Escribe un programa al que se le pasarán dos nombres de archivos de texto por línea de comandos. El programa debe leer el primer archivo y debe escribir en el segundo archivo toda la información del primero invertida (de atrás hacia adelante). Si hay algún error de cualquier tipo el programa debe emitir por pantalla el mensaje:

error

y terminar sin hacer nada más.

Ejemplo: si el contenido del archivo `entrada.txt` es:

```
En un lugar de
la Mancha
de cuyo nombre no
quiero
acordarme
...
```

y ejecutamos en la consola

```
python3 ej3.py entrada.txt salida.txt
```

el archivo `salida.txt` acabará conteniendo

```
...
emradroca
oreiuq
on erbm on oyuc ed
ahcnaM al
ed ragul nu nE
```

4. (ej4.py) (**2 puntos**) El *método de relajación* de resolución de ecuaciones no lineales de la forma $x = f(x)$ consiste en asignar un valor inicial x_0 a la incógnita x y sustituir dicho valor en la parte derecha de la ecuación para encontrar un nuevo valor $x_1 = f(x_0)$. Este proceso se repite $x_2 = f(x_1), x_3 = f(x_2), \dots$, hasta que la diferencia absoluta entre x_i y x_{i+1} es menor que el error permitido.

Resuelve por este método la ecuación $x = 2 - e^{-x}$ partiendo de $x = 1$ con un error menor que 10^{-8} . Imprime además de la solución el número de iteraciones necesario para asegurar el error.

Importante: en este ejercicio sólo está permitido importar el módulo `math`.

5. (ej5.py) (**2 puntos**) Utilizando las librerías `numpy` y `matplotlib.pyplot` y el método `quad` de la librería `scipy.integrate`,

a) Implementa la función `Gamma(x)` que devuelve el valor de la integral

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

b) Haz una gráfica de la función anterior para valores de x en el intervalo $[1, 2]$ con 100 puntos.