

Fundamentos de la Programación. Grado en Física - Curso 2017/2018

Examen de teoría de la convocatoria de febrero (10/01/2018)

Instrucciones

1. Tienes **2 horas y 30 minutos** para realizar el examen. Lee tranquilamente el examen 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 al profesor por instrucciones concretas de Python que no recuerdes.
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**) Escribe un programa que calcule el valor medio de los cuadrados de los números que va leyendo desde el teclado hasta que el usuario introduzca **s**. Para ello el programa pedirá los números con el mensaje **'Introduce el número i (s para acabar): '** (i debe ser el orden del número solicitado). Si la cadena introducida no es un número real válido (o **s**) se debe emitir el mensaje **Número incorrecto** y volver a pedir el número.

Ejemplo:

```
Introduce el número 1 (s para acabar): 12
Introduce el número 2 (s para acabar): 1.1
Introduce el número 3 (s para acabar): 2.4
Introduce el número 4 (s para acabar): re
Número incorrecto
Introduce el número 4 (s para acabar): -3
Introduce el número 5 (s para acabar): s
39.9925
```

2. (ej2.py) (**2 puntos**) Implementa una función llamada **recorta** a la que se le pasen como parámetros una matriz M y dos números enteros positivos i y j . La función debe devolver la matriz obtenida recortando M por la fila i y la columna j (eliminando las filas anteriores a la i y las columnas anteriores a la j). Consideraremos que la primera fila/columna de la matriz tiene índice 1. Ten en cuenta las siguientes posibilidades:
 - Si i es mayor que el número de filas de M o j es mayor que el número de columnas se debe devolver una matriz vacía.
 - Si $i \leq 1$ o $j \leq 1$ se deben devolver todas las filas o columnas respectivamente.

No se debe leer ni imprimir nada.

Ejemplo: Si $M = [[1, 2, 3, 4], [5, 6, 7, 8]]$, las siguientes llamadas a **recorta** deberían devolver la matriz indicada:

```
recorta(M, 0, 2) -> [[2, 3, 4], [6, 7, 8]]
recorta(M, 2, 3) -> [[7, 8]]
recorta(M, 3, 0) -> []
```

3. (ej3.py) (**2 puntos**) Deseamos escribir un programa para dibujar una gráfica con los datos obtenidos de un fichero de texto cuyo nombre se pasará por línea de comandos. Sabemos que el formato de dicho fichero es el siguiente:

- La primera línea contine una descripción de las variables y debe ignorarse.
- La segunda línea contine el número n de pares de puntos en el archivo.
- Las siguientes n líneas continen cada una de ellas las coordenadas x e y de un punto de la gráfica separadas por un espacio.
- El resto del archivo se debe ignorar.
- Se puede suponer que el archivo siempre estará em formato correcto.

Utiliza la librería `matplotlib.pyplot` para dibujar la gráfica¹.

4. (ej4.py) **(2 puntos)** Se trata de justificar numéricamente que

$$\int_0^a \frac{dx}{\sqrt{a^2 - x^2}} = \frac{\pi}{2}$$

independientemente del valor de a . Para ello, debes calcular dicha integral con el método `quad` de la librería `scipy.integrate` utilizando al menos 100 valores positivos diferentes para a en el rango $1 \leq a \leq 10$, mostrando que la diferencia entre cada uno de ellos y $\pi/2$ es siempre menor que el error cometido al calcular la integral². El programa debe devolver `True` si todas las integrales cumplen la condición y `False` si alguna de ellas no la cumple.

5. (ej5.py) **(2 puntos)** Nos han pasado el siguiente programa en Fortran 90 que es capaz de encontrar el número de Hardy-Ramanujan³. Escribe un programa en Python que realice este mismo cálculo.

```
program hardy_ramanujan
  implicit none
  integer :: hr,a,b,c,d

  hr = 0
  do a = 1, 100
    do b = a+1, 100
      do c = a+1, 100
        if (c /= b) then
          do d = c+1, 100
            if (a**3 + b**3 == c**3 + d**3) then
              hr = a**3 + b**3
              exit
            end if
          end do
        end if
      end if
      if (hr /= 0) then
        exit
      end if
    end do
    if (hr /= 0) then
      exit
    end if
  end do
  if (hr /= 0) then
    exit
  end if
  print*, hr, a, b, c, d
end program hardy_ramanujan
```

¹Recuerda que hay que llamar al método `plot(x,y)` de la librería, siendo x la lista de valores de las abscisas e y la lista de valores de las ordenadas.

²Recuerda que `quad` devuelve una tupla de dos elementos, el primero es el valor de la integral y el segundo la cota del error cometido.

³El número de Hardy-Ramanujan es el menor número entero que puede ser expresado como la suma de dos cubos perfectos de dos maneras distintas ($a^3 + b^3 = c^3 + d^3$).