

Funciones I

Ejercicios de clase

Informática - Grupo A3

Grados en Ing. Mecánica e Ing. en Tecnologías Industriales - 1^{er} curso

Fran Bellas - francisco.bellas@udc.es

Hoy haremos prácticas sobre...

- **Tema 7 de teoría: Introducción a Python 3 - Parte 1**
 - Diapositivas 43 a 48:
 - **Funciones**

Ejercicio 1

- Implementa un programa en Python que pida al usuario por teclado dos números enteros n y k que cumplan $n \geq k \geq 1$, calcule las **combinaciones** de n elementos tomados de k en k y muestre el resultado por pantalla:

$$C(n, k) = \frac{n!}{k!(n - k)!}$$

- Se deberá implementar la siguiente función que realiza los cálculos:
 - *factorial*: calcula el factorial de un número n que se pasa como argumento

Ejercicio 1 (ejemplos)

```
=====
Número k. Introduce un valor mayor o igual que 1: -1
Error en el rango. Introduce el número: 2
Número n. Introduce un valor mayor o igual que 2: 1
Error en el rango. Introduce el número: 5
Las combinaciones de 5 elementos tomadas de 2 en 2 son: 10
=====
=====
Número k. Introduce un valor mayor o igual que 1: 3
Número n. Introduce un valor mayor o igual que 3: 10
Las combinaciones de 10 elementos tomadas de 3 en 3 son: 120
=====
=====
Número k. Introduce un valor mayor o igual que 1: 4
Número n. Introduce un valor mayor o igual que 4: 4
Las combinaciones de 4 elementos tomadas de 4 en 4 son: 1
=====
```

Ejercicio 2

- Implementa un programa en Python que pida al usuario por teclado dos números enteros n y k que cumplan $n \geq k \geq 1$, calcule las **combinaciones** de n elementos tomados de k en k y muestre el resultado por pantalla:

$$C(n, k) = \frac{n!}{k!(n - k)!}$$

- Añadir, a la solución anterior, la siguiente función:
 - *pedir_numero*: pide un número por teclado y controla que sea mayor que un mínimo que se pasa como argumento

Ejercicio 2 (ejemplos)

```
=====
Número k. Introduce un valor mayor o igual que 1: -1
Error en el rango. Introduce el número: 2
Número n. Introduce un valor mayor o igual que 2: 1
Error en el rango. Introduce el número: 5
Las combinaciones de 5 elementos tomadas de 2 en 2 son: 10
=====
=====
Número k. Introduce un valor mayor o igual que 1: 3
Número n. Introduce un valor mayor o igual que 3: 10
Las combinaciones de 10 elementos tomadas de 3 en 3 son: 120
=====
=====
Número k. Introduce un valor mayor o igual que 1: 4
Número n. Introduce un valor mayor o igual que 4: 4
Las combinaciones de 4 elementos tomadas de 4 en 4 son: 1
=====
```

Ejercicio 3

- Desarrollar un programa en Python que calcule una aproximación del valor de la constante matemática e dada por la siguiente serie:

$$e = \frac{1}{2} \sum_{k=0}^{\infty} \frac{k+1}{k!}$$

- Para el cálculo del número e no se usarán los infinitos términos necesarios en la serie sino que se le pedirá al usuario un número finito de términos (mayor que cero) a emplear. El resultado se mostrará en pantalla con 15 decimales, comparado con el valor real de e .
- Para realizar el programa es obligatorio el uso de las siguientes funciones:
 - *leer_positivo()*: lee un número entero mayor que 0 por teclado y lo devuelve
 - *factorial()*: calcula y devuelve el factorial de un número entero que recibe como argumento
 - *calcula_e()*: calcula y devuelve el valor aproximado del número e usando el número de términos que recibe por teclado

Ejercicio 3

```
=====
Introduce el numero de términos (>0): -1
Error, introduce de nuevo el dato: 4
El valor aproximado de -e- es 2.687500000000000 y el real 2.718281828459045
=====

=====
Introduce el numero de términos (>0): 12
El valor aproximado de -e- es 2.718281827242330 y el real 2.718281828459045
=====

=====
Introduce el numero de términos (>0): 50
El valor aproximado de -e- es 2.718281828459045 y el real 2.718281828459045
=====
```


Ejercicio 4

- Desarrollar un programa en Python que calcule todos los números **primos gemelos** hasta un número máximo dado por el usuario, que debe ser positivo:
- Dos números primos se denominan gemelos si uno de ellos es igual al otro más dos unidades.
- Por ejemplo, los números primos 3 y 5 forman una pareja de primos gemelos. Otros ejemplos de pares de primos gemelos son 5 y 7 ó 11 y 13
- Funciones a implementar como mínimo:
 - *pedir_max()*: petición del número máximo
 - *es_primo()*: devuelve True si el número que se le pasa como argumento es primo y False si no lo es

Ejercicio 4 (ejemplos)

```
=====
Introduce el numero máximo: -2
Error, el numero debe ser mayor que cero.
Introduce el numero máximo: 15
Los primos gemelos hasta 15 son:
3 y 5
5 y 7
11 y 13
=====
=====
Introduce el numero máximo: 100
Los primos gemelos hasta 100 son:
3 y 5
5 y 7
11 y 13
17 y 19
29 y 31
41 y 43
59 y 61
71 y 73
=====
```

Trabajo autónomo

- **Realizar ejercicios propuestos al resto de grupos.**
- Preparación clase siguiente:
- **Tema 7: Introducción a Python 3**
 - Diapositivas 43 a 48:
 - **Funciones**