

Funciones I

Ejercicios de clase

Informática - Grupo A3

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

Francisco Bellas - francisco.bellas@udc.es

Hoy haremos prácticas sobre...

- **Tema 7 de teoría: Introducción a Python 3 - Parte 1**
 - Diapositivas 41 a 45:
 - **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

- Implementar un programa en Python que:
 - Pida al usuario por teclado un número real (x) que debe estar entre -PI y PI.
 - Pida al usuario por teclado un número entero (k) que debe estar entre 1 y 100.
 - En ambos casos, si los valores están fuera de rango, se pedirán repetidamente.
- Calcule y muestre por pantalla, con 13 decimales, el valor de cos(x) y su aproximación mediante la siguiente serie para el valor de k introducido por teclado:

$$\cos(x) \simeq \sum_{n=0}^k \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

Ejercicio 3

- Este ejercicio ya lo hicimos en la clase anterior (ejercicio 3 semana 04). Partiendo de tu solución, o la del profesor disponible en el campus online, resuélvelo ahora utilizando las siguientes funciones:
 - *pedir_real()*: pide por teclado un número real entre un min y un max que se pasan como argumento
 - *pedir_entero()*: pide por teclado un número entero entre un min y un max que se pasan como argumento
 - *factorial()*: calcula el factorial de un número n que se pasa como argumento
 - *coseno_taylor()*: calcula la aproximación al coseno usando una serie de Taylor

La salida por pantalla cambia un poco, como se muestra en la diapositiva siguiente

Ejercicio 3

=====

Primero necesito el valor de x para calcular su coseno.

Introduce un número real entre -3.14 y 3.14: 4.6

Valor no válido, debe estar entre -3.14 y 3.14: 0.5

Ahora necesito el valor de k para la precisión del cálculo.

Introduce un número entero entre 1 y 100: 0

Valor no válido, debe estar entre 1 y 100: 5

El valor aprox. de $\cos(x)$ es 0.8775825618899 y el real es 0.8775825618904

=====

Primero necesito el valor de x para calcular su coseno.

Introduce un número real entre -3.14 y 3.14: 0.5

Ahora necesito el valor de k para la precisión del cálculo.

Introduce un número entero entre 1 y 100: 10

El valor aprox. de $\cos(x)$ es 0.8775825618904 y el real es 0.8775825618904

=====

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 41 a 45:
 - **Funciones**