

## Guía de Ejercicios 2: Funciones

### Objetivos:

- Comprender el concepto de espacio de memoria de una función.
  - Incorporar la costumbre de especificar una función antes de empezar a programarla, describiendo la cantidad y el tipo de parámetros, el tipo y valor de retorno y los requerimientos sobre los argumentos.
  - Ejercitar la construcción de conjuntos de ejemplos de uso, que ayuden a ilustrar el comportamiento esperado de la función, y que luego servirán como casos de test.
  - Comenzar a trabajar en la detección de distintos tipos de errores en el código.
- 

**Ejercicio 1.** Considerar los siguientes programas. Para cada uno, ejecutarlo primero a mano, haciendo un seguimiento detallado de las variables definidas en cada espacio de memoria, y determinar qué se imprime por pantalla. Después ejecutarlo en la computadora y comparar los resultados.

(a)

```
1  def f(x:int):  
2      x = x + 1  
3  x:int = 10  
4  f(x)  
5  print(x)  
6  f(x * x)  
7  print(x)
```

(b)

```
1  def f(x:int) -> int:  
2      x = x + 1  
3      return x  
4  x:int = 10  
5  x = f(x)  
6  print(x)  
7  print(x * x)
```

(c)

```
1  def f(x:int) -> int:  
2      x = x + 1  
3      return x  
4  def g(y:int) -> int:  
5      x:int = f(y)  
6      return f(x)  
7  x:int = 10  
8  x = g(x)  
9  print(x)  
10 x = g(x * x)  
11 print(x)
```

(d)

```
1  def f(x:int, y:int) -> int:
2      x = 2 * x + y
3      return x
4  x:int = 3
5  y:int = 7
6  y = f(y, x)
7  x = f(y, x)
8  print(x, y)
9  print(x, x*x)
```

**Ejercicio 2.** En el Ejercicio 11 de la Guía 1 escribimos un programa que convierte una temperatura expresada en grados Fahrenheit a grados Celsius. Ahora queremos encapsular ese programa en una función llamada `f2c`, que reciba una temperatura en  $^{\circ}\text{F}$  y la devuelva en  $^{\circ}\text{C}$ .

- (a) Escribir la especificación de la función `f2c`, describiendo los parámetros, las condiciones requeridas sobre los argumentos y el valor de retorno.
- (b) Proveer al menos 5 ejemplos que ilustren el comportamiento esperado.
- (c) Escribir el código de la función en Python.
- (d) Verificar que esta función se ejecute correctamente para los 5 ejemplos elegidos.

**Ejercicio 3.** Para cada uno de los siguientes problemas, **especificar** una función que lo resuelva, describiendo los parámetros, las condiciones requeridas sobre los argumentos y el valor de retorno. Armar además un conjunto de ejemplos (argumentos y valores de retorno) que ilustren el comportamiento de la función y que más adelante servirán como casos de test.

(Hacer este ejercicio en papel. Todavía no se pide pensar los algoritmos ni escribir programas en Python.)

- (a) Dado un entero  $n \geq 0$ , calcular  $\lfloor \sqrt{n} \rfloor$  (es decir, la parte entera de  $\sqrt{n}$ ).
- (b) Dado un entero  $n \geq 0$ , calcular  $n!$  (factorial de  $n$ ).
- (c) Dados dos enteros  $n, k \geq 0$  (con  $k \leq n$ ), calcular el combinatorio  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ .
- (d) Dado un entero  $n \geq 0$ , devolver un string con los valores de  $\binom{n}{i}$  para todo  $0 \leq i \leq n$ , separados por comas. Por ejemplo, para  $n = 4$ , debe devolver "1,4,6,4,1".
- (e) Dado un entero  $n \geq 0$ , devolver un string con una línea de  $n$  asteriscos.
- (f) Dado un entero  $n \geq 0$ , devolver un string que al imprimirse forme un cuadrado de asteriscos de lado  $n$ .
- (g) Dado un string, devolver su inversa. Por ejemplo, para "qwerty" debe devolver "ytrewq".
- (h) Dado un string, calcular la cantidad de veces que contiene la letra 'a'.
- (i) Dados dos strings, devolver la cantidad de veces que el primero está contenido en el segundo.

**Ejercicio 4.** El código del archivo `errores.py` tiene errores de tres tipos distintos: de sintaxis, de tiempo de ejecución (*runtime*) y semánticos. Inspeccionar el código, ejecutarlo y debuggearlo, a fin de encontrar al menos 7 errores, clasificándolos por tipo.