Ejercicio guiado. Unidad 4

a. Recordemos la forma de escribir las funciones en Python con un ejemplo. Vamos a dibujar cuadrados rellenos utilizando un carácter **car** que se pasará como argumento. Se pasan también el desplazamiento de cada línea (valor de a) y el número de caracteres que es escriben (valor de b) que es igual al número de líneas que se escriben. Todos ellos son de entrada (pasados por valor).

```
def figura(a, b, car):
    for i in range(b):
        print(' ' * a, (car + ' ') * b)

figura(2, 4, '&')
```

El resultado es:

& & & &
& & & &
& & & &
& & & &

b. En este ejercicio se utiliza en la izquierda una variable local s cuyo valor se quiere imprimir antes de haberle asignado valor. Genera un error.

```
def f():
    print(s)
    s = 'Hola ¿Quién eres?'
    print(s)

s = 'Me llamo Antonio'
f()
print(s)
```

UnboundLocalError: local variable 's' referenced before assignment

```
def f():
    global s
    print(s)
    s = '¿Qué tal estás?'
    print(s)

s = 'Me llamo Antonio'
f()
```

Me llamo Antonio

¿Qué tal estás?

En el programa de la derecha, la variable **s** se declara global, con lo que tiene acceso a ella y se imprime correctamente.

c. En este ejercicio vamos a usar parámetros mutables para ver el efecto del procedimiento sobre esos parámetros. Para ello, vamos a pasar una lista de enteros a la función y vamos a hacer que devuelva la lista ordenada. Lo hacemos como ejercicio, ya que en la práctica existe una función (método) llamado sort que realiza esa operación. Por ejemplo:

```
lst = [1, 5, 3, 7]
lst.sort()
print(lst)
[1, 3, 5, 7]
```

El algoritmo de ordenación escrito en Python queda en la forma:

```
def ordenar(lista):
    for i in range(len(lista)):
        menor = i
        for k in range(i+1, len(lista)):
            if lista[k] < lista[menor]:
                menor = k
        intercambiar(lista, menor, i)

def intercambiar(a, x, y):
    temp = a[x]
    a[x] = a[y]
    a[y] = temp

mi_lista = [1, 4, 2, 7, 3, 9, 2]
ordenar(mi_lista)
print(mi_lista)
[1, 2, 2, 3, 4, 7, 9]
```

Se le ha pasado a la función procedimiento un parámetro mi_lista que es una lista, por tanto, mutable. La función cambia el orden de los elementos de la lista y tal cambio se refleja en el parámetro real, que imprime la lista ordenada.