Consideraciones Generales:

- Antes de comenzar a resolver el examen tener en cuenta lo siguiente:
- Las funciones deben contener su receta completa.
- De ser posible, use modularización en la resolución, priorice la legibilidad del código producido y su concordancia con respecto a la estrategia de solución pensada para dar respuesta al problema a resolver.
- El testing se realizará mediante la librería pytest, se recomienda no menos de tres casos de test por función.

Ejercicio 1:

Complete la receta de la función: `puntaje_alien`, y luego **reescriba** el *condicional encadenado* usando un *condicional secuencial*.

```
1 def puntaje_alien(color):
2    if (color == “verde”):
3          puntos = 5
4    elif (color == “amarillo”):
5          puntos = 10
6    elif (color == “rojo”):
7          puntos = 15
8    else:
9          puntos = 0
10
11   return puntos
```

Ejercicio 2: ¡Vamos al parque de diversiones!

Se desea registrar y procesar las ventas de las entradas en un parque de diversiones. Las entradas tienen diferentes valores según la edad del cliente:

- tipo 0: Para los/as menores de 4 años es gratis.
- tipo 1: Para las personas comprendidas entre 4 y 18 años es $2500.
- tipo 2: Para las personas mayores de 18 años el valor es de $4000.

No se sabe cuántos visitantes vendrán al parque durante el día.
La persona que opera el sistema dará cierre a la caja ingresando un valor especial, en este caso el valor -1.

Al finalizar la jornada, se mostrará en pantalla:

1. Cuántas entradas de cada tipo se vendieron.
2. Cuánto se recaudó durante el día.
3. Qué tipo de entrada fue la que más se vendió.

Escriba primero una estrategia de solución con sus palabras, y luego desarrolle un programa que resuelva este problema planteado siguiendo su estrategia. Todas las funciones que escriba deben estar diseñadas.

Ejercicio 3: Dada la siguiente definición matemática de la función `doble_factorial`

$$n!! = \begin{cases} 1 & si\ n = 0\ o\ n = 1; \\ n \times (n-2)!! & si\ n \geqslant 2. \end{cases}$$

$$8!! = 8 \times 6 \times 4 \times 2 \times 1 = 384.$$

$$5!! = 5 \times 3 \times 1 = 15.$$

Te pedimos:

Diseñar de **forma recursiva** en Python la función `doble_factorial`, suponiendo que la persona que hará uso del programa ingresa un valor por teclado y el programa imprime el resultado en pantalla. Indique qué tipo de recursión utilizó en su solución.