

 <b>Deusto</b> Facultad de Ingeniería Ingeniaritza Fakultatea	EXAMEN DE PROGRAMACIÓN I	25-OCTUBRE-2023
---	-----------------------------	-----------------

**Nombre:** \_\_\_\_\_  
**DNI:** \_\_\_\_\_ **Aula:** \_\_\_\_\_ **Fila:** \_\_\_\_\_ **Columna:** \_\_\_\_\_

### Instrucciones

**Confidential**

- El ejercicio a pie de máquina se entregará al terminar el examen.
- NO está permitida la conexión a internet en el examen. Tampoco la utilización de tu propio portátil ni la consulta de apuntes o de otros códigos.
- Guarda el código de todos los ejercicios en un único fichero que se llame "Examen1-NombreApellidos.py", por ejemplo: Examen1-PabloGaraizar.py.
- El examen debe incluir un comentario al inicio con tu nombre, apellidos y DNI.

### Ejercicio 1

**Ejercicio 1a: (2 puntos).** Programa una función llamada **multiplicar** que reciba dos números enteros como argumentos y devuelva el resultado de multiplicar todos los números enteros entre esos dos números (ambos incluidos). Comprueba que la función funciona correctamente con varios ejemplos:

```
print(multiplicar(1, 5)) # Devuelve 120
print(multiplicar(3, 5)) # Devuelve 60
print(multiplicar(2, 9)) # Devuelve 362880
```

**Ejercicio 1b: (2 puntos).** Programa una función llamada **factoriales** que reciba una lista de enteros como argumento y devuelva una lista que contenga los factoriales de los enteros recibidos. Comprueba que la función funciona correctamente con varios ejemplos:

```
print(factoriales([1, 2, 3])) # Devuelve [1, 2, 6]
print(factoriales([4, 5, 6, 7])) # Devuelve [24, 120, 720, 5040]
```

### Ejercicio 2 (3 puntos)

Programa una función llamada **mas\_vocales** que reciba una lista de palabras como argumento y devuelva la palabra que tenga más vocales de todas ellas. Comprueba que la función funciona correctamente con varios ejemplos:

```
print(mas_vocales(["aceituna", "tomate", "hola"])) # Devuelve "aceituna"
```

### Ejercicio 3 (3 puntos)

Programa una función llamada **posiciones\_primas** que reciba una lista de elementos y devuelva una lista que contenga los elementos de la lista recibida que se encuentren en las posiciones que sean números primos, es decir, los elementos de las posiciones 2, 3, 5, 7, 11, 13, 17, etc.

Por ejemplo, si la función `posiciones_primas` recibiera esta lista, las posiciones que son números primos serían la 2, la 3 y la 5:

```
lista = ["a", "b", "c", "d", "e", "f", "g"]
posición: 0    1    2    3    4    5    6
primos:           *    *           *
resultado = [      "c", "d",          "f" ]
```

(sigue en la página de atrás...)



**Confidential**

Comprueba que la función funciona correctamente con varios ejemplos:

```
print(posiciones_primas(["a", "b", "c", "d", "e", "f", "g", "h"]))  
# Devuelve ["c", "d", "f", "h"]  
print(posiciones_primas([0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]))  
# Devuelve [2, 3, 5, 7, 11, 13]
```

### Ejercicio Extra (1 punto)

Programa una función llamada **comprimir\_vocales** que reciba una frase compuesta enteramente por vocales y devuelva una frase que contenga una versión comprimida de la frase recibida, indicando cuántas veces se repite cada letra y qué letra es. Por ejemplo, la frase "**aaaaaeaaaii**" se comprimiría como "**5a4e1a2i**" (5 veces a, 4 veces e, 1 vez a, 2 veces i).

Comprueba que la función funciona correctamente con varios ejemplos:

```
print(comprimir_vocales("aaaaaeaaaii")) # Devuelve "5a4e1a2i"  
print(comprimir_vocales("uuuuuuuuuuui")) # Devuelve "11u1i"
```