

## Alumno/a:

---

1. Indica cuáles de los siguientes son identificadores válidos en Python: (3 puntos)

1erMiembro		libre?		año_nac	
SegundoMiembro		puerta.abierta		\$saldo\$	
coche_1		cámara4		import	

---

2. Considera las siguientes operaciones en Python: (2 puntos)

```
(1) x = 1+2*3+4  
(2) x = (1+2)*(3+4)  
(3) x = 1+(2*3)+4  
(4) x = 1+2*(3+4)
```

¿Cuál de las siguientes frases es correcta?

- A) Todas las operaciones son iguales
- B) La (1), (2) y (3) son iguales
- C) La (1), (3) y (4) son iguales
- D) La (1) y (3) son iguales
- E) Todas son diferentes

---

3. Considera las siguientes operaciones en Python: (3 puntos)

```
a = True  
b = False
```

```
(1) res = a or not b and not a or b  
(2) res = (a or not b) and (not a or b)  
(3) res = a or (not b and not a) or b  
(4) res = ((a or not b) and not a) or b
```

¿Cuál de las siguientes frases es correcta?

- A) Todas son True
- B) Todas son False
- C) La (1) y la (3) son True, y las otras dos False
- D) La (1) y la (3) son False, y las otras dos True

4. Para transformar una cadena de texto en entero, se usa la siguiente función: (2 puntos)

- A) int()
- B) append()
- C) str()
- D) float()

5. Considera las siguientes operaciones en Python: (6 puntos)

```
mystr = "programación orientada a objetos"  
mystr2 = mystr[n:m]
```

5.1 ¿Qué es mystr2 cuando n igual a 0, y m es igual al tamaño de mystr (i.e. len(mystr) == m)?

**\* Elige la o las respuestas correctas (puede haber más de una, elige todas)**

- (a) mystr
- (b) "programación orientada a objetos"
- (c) "programación ori"
- (d) "entada a objetos"

5.2 ¿Qué es mystr2 cuando n igual a 0, y m es igual la mitad del tamaño de mystr? (4 pts)

**\* Elige la o las respuestas correctas (puede haber más de una, elige todas)**

- (a) mystr
- (b) "programación orientada a objetos"
- (c) "programación ori"
- (d) "entada a objetos"

5.3 ¿Qué es mystr2 cuando n es igual a la mitad del tamaño de mystr y m es igual a -1?

**\* Elige la o las respuestas correctas (puede haber más de una, elige todas)**

- (a) mystr
- (b) "programación orientada a objetos"
- (c) "programación ori"
- (d) "entada a objeto"

---

6. Una lista en Python puede contener: (2 puntos)

- A) sólo cadenas de texto
- B) cualquier tipo
- C) sólo 100 elementos
- D) solamente números

---

7. Escribe (en **Python**), usando comprensión de listas, cómo generar una lista con los números pares comprendidos entre 0 y 99 en una sola línea de código. (5 puntos)

8. El siguiente código es de un programa que calcula el mes más lluvioso a partir del registro mensual en litros. Debes completar este código sustituyendo los comentarios en negrita con las operaciones sobre listas adecuadas para una correcta ejecución del programa. (4 puntos)

```
lluvia_mensual = [65, 70, 87, 62, 44, 14, 5, 5, 24, 50, 57, 69]
meses = ['enero', 'febrero', 'marzo', 'abril',
         'mayo', 'junio', 'julio', 'agosto',
         'septiembre', 'octubre', 'noviembre', 'diciembre']
max_lluvia = # calcula aquí el máximo ..
              # ..valor de lluvia registrada
mes_max = # obtén el índice del mes ..
          # correspondiente
print('El mes más lluvioso ha sido', meses[mes_max],
      'con', max_lluvia, 'litros')
```

---

9. ¿Qué instrucción nos permite tener esta salida? (2 puntos)

```
a = {1, 2, 3}
b = {4, 5, 6}
```

```
{1, 2, 3, 4, 5, 6}
```

- A) a.append(b)
  - B) a.append(b) tres veces seguidas
  - C) a.union(b)
  - D) b.union(a)
- 

10. Indica qué valores aparecerían en pantalla al ejecutar el siguiente código: (5 puntos)

```
def calcula():
    a = b*2
    c = 23
    print(a*c)

b = 5
a = 8
c = 10

print(a)
print(b)
print(c)
calcula()
print(a*c)
```

11. Indica el resultado de ejecutar las siguientes expresiones o líneas de código. (14 puntos)

A. `list(range(10, 2, -3))`

B. 

```
l = []
for i in range(4):
    l.append(list(range(i)))
print(l)
```

C. `max(list(range(10, 21, 2)))`

D. `min(list(range(10, 21, 2)))`

E. `'abc' * 2`

F. `'abracadabra'.count('a')`

---

12. Completa el código para obtener la **salida** mostrada en la imagen (4 puntos)

```
def add_many(*args):
    s = 0
    for _____:
        s += n
    print(s)

add_many(100, 50, 3)
```

153

---

13. Dada esta cabecera de la función `def nuevo_pedido(producto, precio, descuento=0)`, determina cuáles son llamadas válidas entre las propuestas: (10 puntos)

- (a) `nuevo_pedido('probeta', 5, 0.25)`
- (b) `nuevo_pedido()`
- (c) `nuevo_pedido('libro')`
- (d) `nuevo_pedido('lupa', 12)`
- (e) `nuevo_pedido(producto='pinzas', 10)`
- (f) `nuevo_pedido(producto='laser', precio=25)`
- (g) `nuevo_pedido(producto='telescopio', descuento=0.2, precio=25)`
- (h) 

```
pedido = ('laptop', 1200, 0.6)
nuevo_pedido(pedido)
```
- (i) 

```
pedido = ('producto': 'calculadora', 'valor': 19.99, 'descuento': 0)
nuevo_pedido(**pedido)
```

14. ¿Qué parámetro es necesario poner en la definición de la función? (3 puntos)

```
def make_dict(_____):  
    return kwargs  
  
make_dict(a = 1, b = 2)
```

- A) \*\*args
- B) a = x, b = y
- C) \*kwargs
- D) \*\*kwargs

---

15. ¿Qué código de una sola línea serviría para asignar al primer contacto los mismos valores asociados al último? (3 puntos)

```
contactos = [  
    {'nombre': 'Mick', 'apellido': 'Jagger', 'edad': 77},  
    {'nombre': 'Keith', 'apellido': 'Richards', 'edad': 77},  
    {'nombre': 'Ronnie', 'apellido': 'Wood', 'edad': 73},  
    {'nombre': 'Charlie', 'apellido': 'Watts', 'edad': 79}  
]
```

---

16. ¿Qué es necesario poner en el hueco del código para obtener esa salida? (2 puntos)

```
packages = ["numpy", "pandas", "scipy"]  
  
for i in packages:  
    print(_____)
```

```
numpy  
pandas  
scipy
```

---

17. Escribe un programa (en **Python**) que itere sobre la lista de nombres ['Pedro', 'Pablo', 'Judas', 'Juan']. Dentro del bucle, comprobará si el nombre es 'Judas'. De ser así, el programa debería escribir en pantalla el nombre junto con el texto 'es el culpable', si no, debería escribir el nombre junto con el texto 'es inocente'. El resultado esperado es: (5 puntos)

```
Pedro es inocente  
Pablo es inocente  
Judas es el culpable  
Juan es inocente
```

18. Escribe un programa (en **Python**, **pseudocódigo**, o **lenguaje natural**) que pida al usuario su edad y la de su mejor amigo. El programa mostrará en pantalla quién es el mayor de los dos.

(5 puntos)

---

19. Escribe un programa (en **Python**, **pseudocódigo**, o **lenguaje natural**) que lea una cadena escrita por teclado y compruebe si el primer y el último carácter son iguales. Si son iguales, mostrará un mensaje con el número total de caracteres de la cadena distintos a dicho carácter. En caso contrario, mostrará un mensaje con el número total de caracteres de la cadena iguales al carácter inicial y al carácter final (sin incluirlos). (10 puntos)

---

20. Escribe un programa (en **Python**, **pseudocódigo**, o **lenguaje natural**) que calcule la frecuencia de aparición de cada letra en una cadena. Por ejemplo, para la cadena “abracadabra” debería mostrar: (10 puntos)

{'a': 5, 'b': 2, 'r': 2, 'c': 1, 'd': 1}