Nombre Onofre Benjumea **Fecha** 17/11/22

EXAMEN DE CONOCIMIENTOS EN PYTHON

1. **¿Python es un lenguaje fuertemente tipado?**

- [ ] Si
- [ ] No

2. **Para añadir una condición alternativa a una declaración condicional if se utiliza**

- [ ] elseif
- [ ] elif
- [ ] else if
- [ ] elsif

3. **¿Python es un lenguaje interpretado?**

- [ ] Si
- [ ] No

4. **Para mostrar el valor de la posición 2 de un array llamada 'test' utilizamos**

- [ ] print(test [1])
- [ ] print(test[2])
- [ ] puts(test[2])
- [ ] pp(test[2])

5. **¿Qué tan necesaria la indentación de código en Python?**

- [ ] Es muy necesaria, si no está bien arrojará errores.
- [ ] No es necesaria.
- [ ] Es opcional, depende del programador.
- [ ] El código en Python no se puede indentar.

6. **¿Qué es el __init__ y haga un ejemplo de su uso?**

Es un metodo que permite incializar los atributos de un objeto

```
class Persona:
  def __init__(self, nombre, edad):
    self.nombre = nombre
    self.edad = edad
```

7. **¿Cuál de los siguientes es un objeto de tipo diccionario?**

- [ ] diccionario = ('Numero': 1, 'Nombre': 'Juan')
- [ ] diccionario = {'Numero' -> 1, 'Nombre' -> 'Juan'}
- [ ] diccionario = {'Numero': 1, 'Nombre': 'Juan'}
- [ ] diccionario = {'Numero' => 1, 'Nombre' => 'Juan'}

8. **¿Qué es el Self en Python?**

- [ ] Es un parámetro opcional de un método.
- [ ] Es un tipo de dato.
- [ ] Se utiliza para mostrar en consola el valor de una variable o cadena de texto.
- [ ] Es una instancia u objeto de una clase.

9. **¿Cuál es la forma correcta de escribir un bucle for?**

- [ ] for(a in range[0..3])

- [ ] for a in range(0..3)
- [ ] for(a=0; a<3; a++)
- [ ] for a in range(0, 3)

10. ¿Cuál es la diferencia entre *args, **kwargs?

Con **kwargs los parametros opcionales se pasan como un diccionario

11. **¿Qué diferencia hay entre una clase y un objeto?**

- [ ] Un objeto es una instancia de una clase.
- [ ] Ninguna.
- [ ] Una clase es una instancia de un objeto.
- [ ] Un objeto no tiene tipo.

12. **El resultado de la declaración print ('%.2f' % 1714.666) es**

- [ ] 1714.66
- [ ] 1714.67
- [ ] 1714.0
- [ ] 1715

13. **Defina una lista e insértele valores:**

```
lista = [25, 34, 453, 534]
```

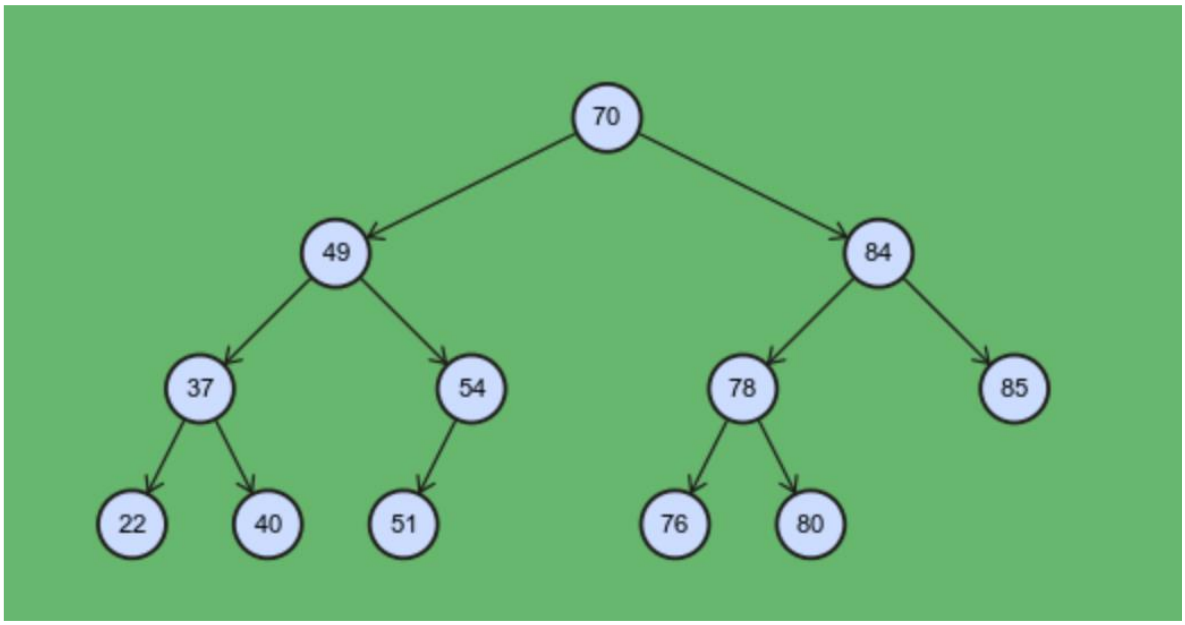
14. **Defina una tupla e insértele valores:**

```
tupla = (25, 34, 453, 534)
```

15. **Dado un árbol binario, encuentre el ancestro común más cercano entre dos nodos**.

Inputs:
70,84,85
70,84,78,80
70,84,78,76
70,49,54,51
70,49,37,40
70,49,37,22

El árbol correspondiente a estos datos es el siguiente:



Output:
ancestor(40,78) = 70
ancestor(51,37) = 49
ancestor(76,85) = 84

Diseñe un API REST que permita:

1. Crear un árbol.
2. Dado un árbol y dos nodos, retorne el ancestro común más cercano.