1.2.5 ¿Qué función debo usar para conseguir el mismo resultado que en la parte 1.1?¿Por qué en la parte 1.1 vemos el resultado aun sin haber usado esta función?

(tecnicismos) *... Este es un modo especial de Python, el cual evalúa el "código" que nosotros le ingresamos y nos devuelve el valor resultante en la siguiente línea. ....* Lo importante es que el intérprete lo que está haciendo (y siempre hace) es **imprimir** el resultado de la evaluación. ***(importante)*** *En el caso 1.1, la cadena "Hola Algoritmos y Programacion", al no ser modificada, es evaluada como la cadena misma. ....* A qué te referís con "*ser modificada*"? En ninguno de los dos casos se modifica la cadena. En ambos casos se evalúa la cadena de la misma forma. La diferencia recae simplemente en lo explicado previamente: el intérprete imprime el resultado de la evaluación.

2.5.1 ¿Cuál es la salida del programa?

(tecnicismo poco importante) *La salida del programa* ***es un error*** *del tipo: AssertionError.* Es un **mensaje** de error, es decir, la forma de explicar el error al usuario. El error en sí vive dentro del programa. Esto es un super tecnicismos así que importa poco, lo corrijo solo por si te interesa saberlo.

2.5.3 ¿Qué hace la instrucción assert?

(tecnicismos) *La instrucción assert evalúa una condición lógica; si esta resulta verdadera, el programa continúa, pero si esta falsa el programa es detenido y* ***devuelve el error*** *AssertionError.* En realidad el `assert` no devuelve un error sino que lo **lanza**. Esto también es un super tecnicismo.

4.6 Renombrar la función y las variables de forma que sus nombres sean representativos. ¿Por qué es importante hacer esto?

(tecnicismos) *... Cuanto más claros e ilustrativos sean los nombres de nuestras variables y funciones (además de los comentarios y documentación); ...* Recordá que si bien queremos que nuestro código esté bien documentado, intentamos evitar que este muy comentado. Esto quiere decir, que no debería ser necesario hacer tantos comentarios (comentario = todo lo que viene después de un `#` en python). Nuestro código debería ser simple y autoexplicativo. Es decir, cualquiera debería leer nuestro código y entender perfectamente que estamos haciendo sin necesidad de leer ningún comentario. Los comentarios los dejamos para cuando estamos haciendo algo feo, difícil de leer que hay que explicarlo (esto no debería pasar nunca en la materia). De todas formas entiendo que los comentarios los podés haber puesto para vos. Y es entendible que en tu primer cuatri necesites comentarios para entender el código, pero la idea es que a futuro no lo necesites y que sepas que yo (o quien lea tu código) no debería necesitarlos.

5.4 ¿Cuál es la importancia de reutilizar funciones?

La importancia de reutilizar funciones reside en el hecho de que no se pierde tiempo escribiendo algo que ya existe; ese tiempo ganado se puede usar para tareas más importantes. Además, si reutilizamos funciones de librerías pre-existentes con cierto

“prestigio” nos “aseguramos” de que esta funciona eficientemente. Tu respuesta está muy bien, te agrego otro motivo que para mi es el principal. Imagína que tenemos código repetido y por algún motivo hay que hacer un cambio. Entonces, deberías ir a todas las repeticiones de este código y cambiarlo. Esto ya de por sí es tedioso pero no lo mas grave. Lo peor es que se presta a errores. Si te olvidas de hacer el cambio en una de las repeticiones el código no va a funcionar como querías. Y en general estos errores son dificiles e encontrar.