

Ejercicio 1

Nombre: Orsi Tomas Fabrizio

Catedra: Curso 04 - Essaya

Fecha de entrega: 29/03/2022

Ejercicio 1

Parte 1.1

```
[fabri@hparch ~]$ python3
Python 3.10.4 (main, Mar 23 2022, 23:05:40) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information
>>> "Hola Algoritmos y Programacion I"
'Hola Algoritmos y Programacion I'
>>>
[1]+  Stopped                  python3
[fabri@hparch ~]$
```

Parte 1.2

Pregunta 4

```
[fabri@hparch EJ1]$ python3 parte_1_2.py
[fabri@hparch EJ1]$
```

1

Pregunta 5

- Para obtener el mismo resultado que en la parte 1.1 deberíamos haber usado la función *print()*:
 - `print("Hola Algoritmos y Programacion")`
- En la Parte 1.1 vemos la cadena "Hola Algoritmos y Programacion" porque escribimos este texto en el modo interactivo de Python. Este es un modo especial de Python, el cual evalúa el "codigo" que nosotros le ingresamos y nos devuelve el valor resultante en la siguiente línea. En el caso 1.1, la cadena "Hola Algoritmos y Programacion", al no ser modificada, es evaluada como la cadena misma.

¹ EJ1 es el directorio donde el archivo "parte_1_2.py" esta guardado

Parte 2

Pregunta 2

```
[fabri@hparch EJ1]$ python3 norma.py  
[fabri@hparch EJ1]$
```

Pregunta 5

```
[fabri@hparch EJ1]$ python3 norma.py  
Traceback (most recent call last):  
  File "/home/fabri/Documents/Facultad/Academico/Algoritmos-Programacion-1/TPS/EJ1/norma.py", line 17, in <module>  
    assert norma(-70, 14, z) == 111.0  
AssertionError  
[fabri@hparch EJ1]$
```

1. La salida del programa es un error del tipo: AssertionError
2. Podemos saber en qué línea se dio el error si leemos el mensaje de error. Después de la ubicación del archivo figura en qué línea se dio el error (en este caso la línea 17); además en el siguiente renglón figura una copia de la línea donde está el error (en este caso: `assert norma(-70, 14, z) == 111.0`)
3. La instrucción `assert` evalúa una condición lógica; si esta resulta verdadera, el programa continúa, pero si esta es falsa el programa es detenido y devuelve el error `AssertionError`
4. Para hallar el valor de `z` podemos plantear la siguiente ecuación

$$\sqrt{(-70)^2 + (14)^2 + (z)^2} = 111$$

$$(z)^2 = (111)^2 - (-70)^2 - (14)^2$$

$$z = \sqrt{(111)^2 - (-70)^2 - (14)^2}$$

$$z \pm 85$$

Si $z = \pm 85$, la expresion antes del assert va a ser verdadera y el programa no va a detenerse (y no va a devolver ningun error)

Parte 3

Pregunta 3

```
[fabri@hparch EJ1]$ python3 diferencia.py
Traceback (most recent call last):
  File "/home/fabri/Documents/Facultad/Academico/Algoritmos-Programacion-1/TPS/EJ1/diferencia.py", line 10, in <module>
    assert diferencia(1, 2, 3, 1, 2, 3) == (0, 0, 0)
  File "/home/fabri/Documents/Facultad/Academico/Algoritmos-Programacion-1/TPS/EJ1/diferencia.py", line 6, in diferencia
    return dif_x, dif_y, diff_z
NameError: name 'diff_z' is not defined. Did you mean: 'dif_z'?
[fabri@hparch EJ1]$
```

Pregunta 4

El programa devolvio un error. Devolvio el error NameError. Este error significa que estamos llamando a una funcion/variable que no fue definida. En este caso llamamos a la variable “diff_z” en la línea 6, la cual nunca fue definida.

Pregunta 5

Si en la línea 6 reemplazamos “diff_z” por “dif_z”, la línea 6 nos queda:
 “`return dif_x, dif_y, dif_z`”. Esto produce que el programa ya no nos devuelva el error.

Cabe destacar que se cumple que los 3 asserts del programa son verdaderos; si este no fuese el caso el programa devolveria error AssertionError

Parte 4

Pregunta 4

Cuando corremos el programa, este nos devuelve el error AssertionError en la línea 10.

Pregunta 6

Es importante renombrar las funciones y variables para las etapas de Mantenibilidad y Testeabilidad del proyecto informatico. Cuanto más claros e ilustrativos sean los nombres de nuestras variables y funciones (ademas de los comentarios y documentacion); más fácil va a ser la labor de buscar errores en el código pre & post lanzamiento.

Pregunta 7

Si, se podria (hasta cierto punto). De la siguiente manera:

$\text{vectOrtogonal_x, vectOrtogonal_y, vectOrtogonal_z} = (y1*z2 - y2*z1), -(x1*z2 - x2*z1), (x1*y2 - x2*y1).$

El return, sin embargo, debe permanecer en la siguiente línea

Parte 5

Pregunta 4

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.

Ademas, si reutilizamos funciones de librerias pre-existentes con cierto “prestigio” nos “aseguramos” de que esta funciona eficientemente.