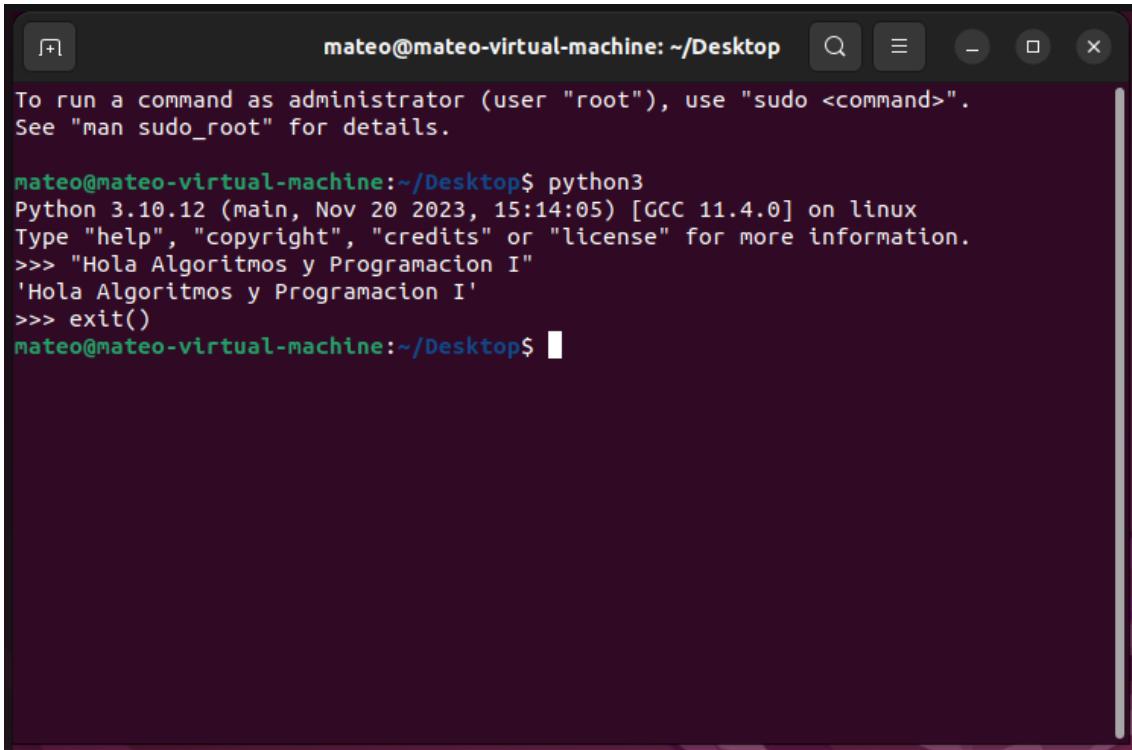


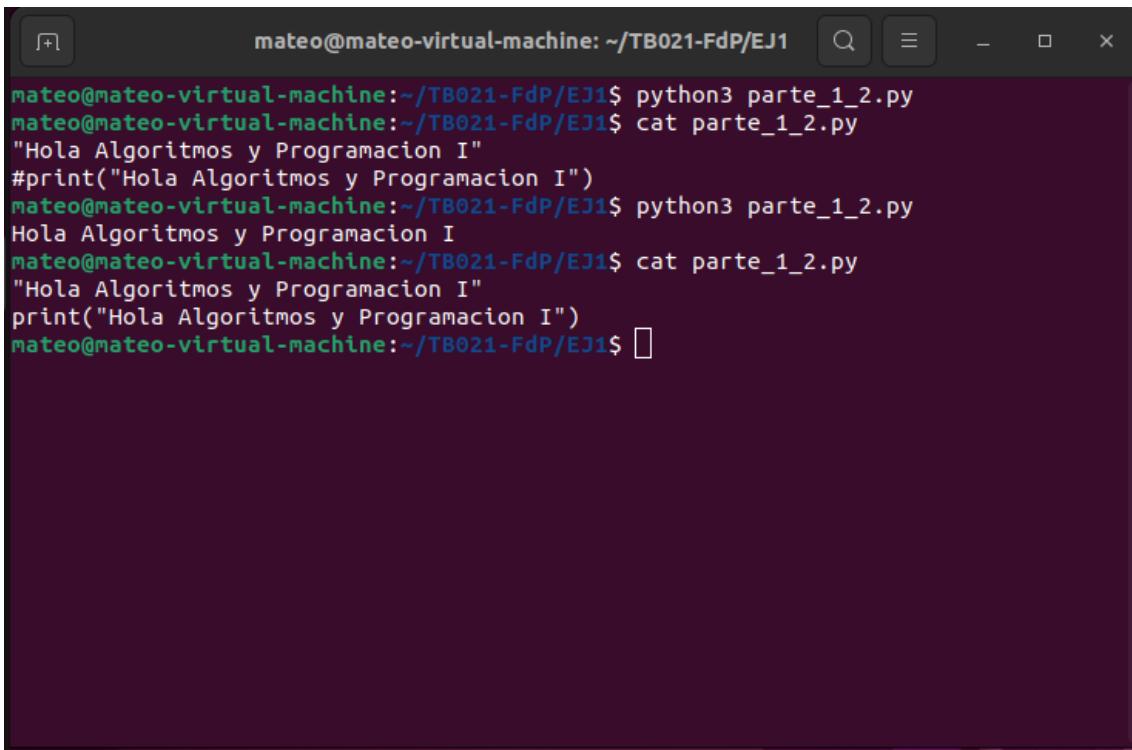
Facultad de Ingeniería de la Universidad de Buenos Aires	Apellido y nombre del alumno: Gonzalez Pautaso Mateo	Folio N°: 1
Materia: Fundamentos de Programación (TB021)	Ejercicio Obligatorio N° 1 Área de polígonos	Padrón: 111699
Catedra: Essaya	Fecha de entrega: 22/03/2024	

Parte 1.1



```
mateo@mario-virtual-machine: ~/Desktop$ python3
Python 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> "Hola Algoritmos y Programacion I"
'Hola Algoritmos y Programacion I'
>>> exit()
mateo@mario-virtual-machine: ~/Desktop$
```

Parte 1.2

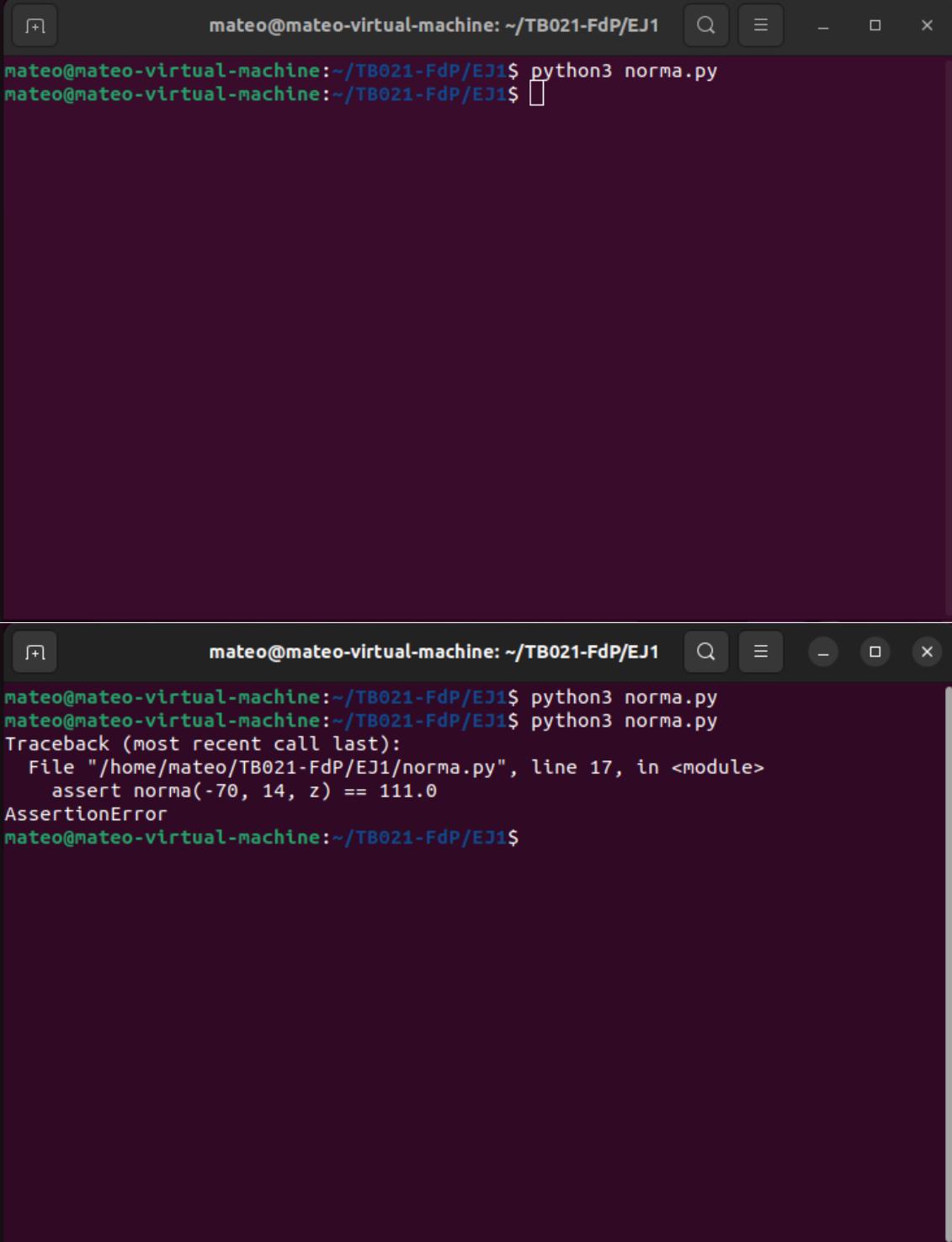


```
mateo@mario-virtual-machine: ~/TB021-FdP/EJ1$ python3 parte_1_2.py
mateo@mario-virtual-machine: ~/TB021-FdP/EJ1$ cat parte_1_2.py
"Hola Algoritmos y Programacion I"
#print("Hola Algoritmos y Programacion I")
mateo@mario-virtual-machine: ~/TB021-FdP/EJ1$ python3 parte_1_2.py
Hola Algoritmos y Programacion I
mateo@mario-virtual-machine: ~/TB021-FdP/EJ1$ cat parte_1_2.py
"Hola Algoritmos y Programacion I"
print("Hola Algoritmos y Programacion I")
mateo@mario-virtual-machine: ~/TB021-FdP/EJ1$
```

Facultad de Ingeniería de la Universidad de Buenos Aires	Apellido y nombre del alumno: Gonzalez Pautaso Mateo	Folio N°: 2
--	---	-------------

Para conseguir el mismo resultado que en la parte 1.1 hay que usar la función print() así cuando se ejecuta el archivo, se muestra el mensaje. En la parte 1.1 vemos directamente el resultado ya que se ejecuta la línea, gracias al prompt que nos permite evaluarla por Python. Otra forma sería escribiendo cat parte1_2.py pero nos mostraría todo el contenido del archivo, no solo lo que queremos ver

Parte 2



```
mateo@mateo-virtual-machine:~/TB021-FdP/EJ1$ python3 norma.py
mateo@mateo-virtual-machine:~/TB021-FdP/EJ1$ 111.0

mateo@mateo-virtual-machine:~/TB021-FdP/EJ1$ python3 norma.py
mateo@mateo-virtual-machine:~/TB021-FdP/EJ1$ python3 norma.py
Traceback (most recent call last):
  File "/home/mateo/TB021-FdP/EJ1/norma.py", line 17, in <module>
    assert norma(-70, 14, z) == 111.0
AssertionError
mateo@mateo-virtual-machine:~/TB021-FdP/EJ1$
```

Facultad de Ingeniería de la Universidad de Buenos Aires	Apellido y nombre del alumno: Gonzalez Pautaso Mateo	Folio N°: 3
--	---	-------------

La salida del programa es un `AssertionError` debido a la función `assert`. La línea en la que se generó el error es la 17, la terminal nos muestra “line 17” para que sepamos donde está. La instrucción `assert` corrobora que lo que devuelve la función sea correcto, devolviéndonos un `True` o `False` según la condición. Si la devolución es `True`, sigue de largo y no pasa nada, pero si es `False`, imprime `AssertionError`. El valor de `z` para que ya no de error es $|85| = \begin{cases} -85 \\ 85 \end{cases}$

Parte 3

```
mateo@mateo-virtual-machine:~/TB021-FdP/EJ1$ python3 diferencia.py
Traceback (most recent call last):
  File "/home/mateo/TB021-FdP/EJ1/diferencia.py", line 10, in <module>
    assert diferencia(1, 2, 3, 1, 2, 3) == (0, 0, 0)
  File "/home/mateo/TB021-FdP/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'?
mateo@mateo-virtual-machine:~/TB021-FdP/EJ1$
```

Se detectó un error `NameError` en la línea 6, la variable `diff_z` no coincidía con la del `return diff_z`, había una doble f. Eso generó que la variable `diff_z` no esté definida y luego el `assert` no tuviese la suficiente cantidad de parámetros para comprobar.

Parte 4

El programa muestra un `AssertionError` en la línea 10, por lo que algo dentro de la función está fallando.

Es importante documentar el código para que a la hora de abrirlo nuevamente podamos entenderlo más fácil con la información proporcionada, mismo caso cuando se trabaja con otras personas así lo entienden. Con leer unas simples líneas de texto podemos entender para qué sirve el código

Parte 5