

**Materia: Algoritmos y
Programación I**

Cátedra: Essaya

Práctica: EJ1

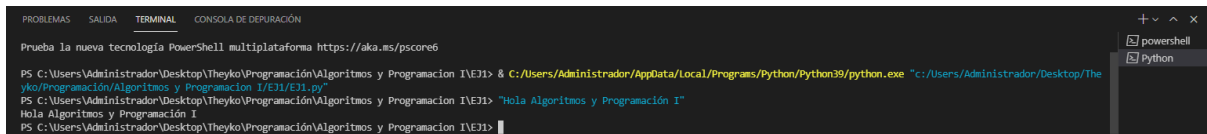
Nombre: Lucas Javier Cambon

Padrón: 105085

**Ayudante a cargo: Alejo
Rodriguez**

Parte 1:

- Parte 1.1

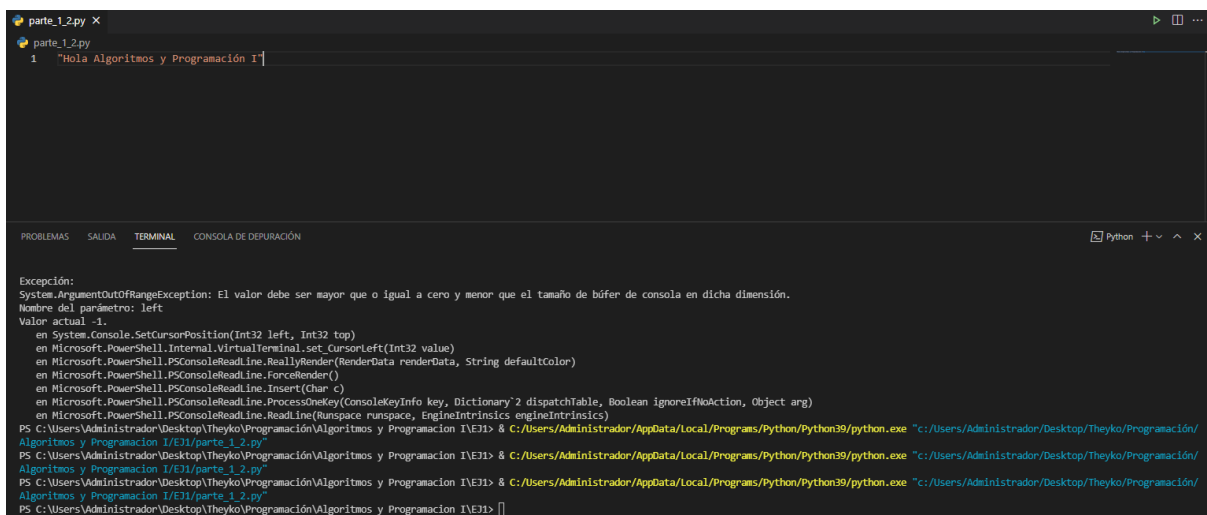


```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/The
yko/Programación/Algoritmos y Programación I/EJ1/EJ1.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> "Hola Algoritmos y Programación I"
Hola Algoritmos y Programación I
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> |
```

- Parte 1.2



```
parte_1.2.py X
parte_1.2.py
1  "Hola Algoritmos y Programación I"

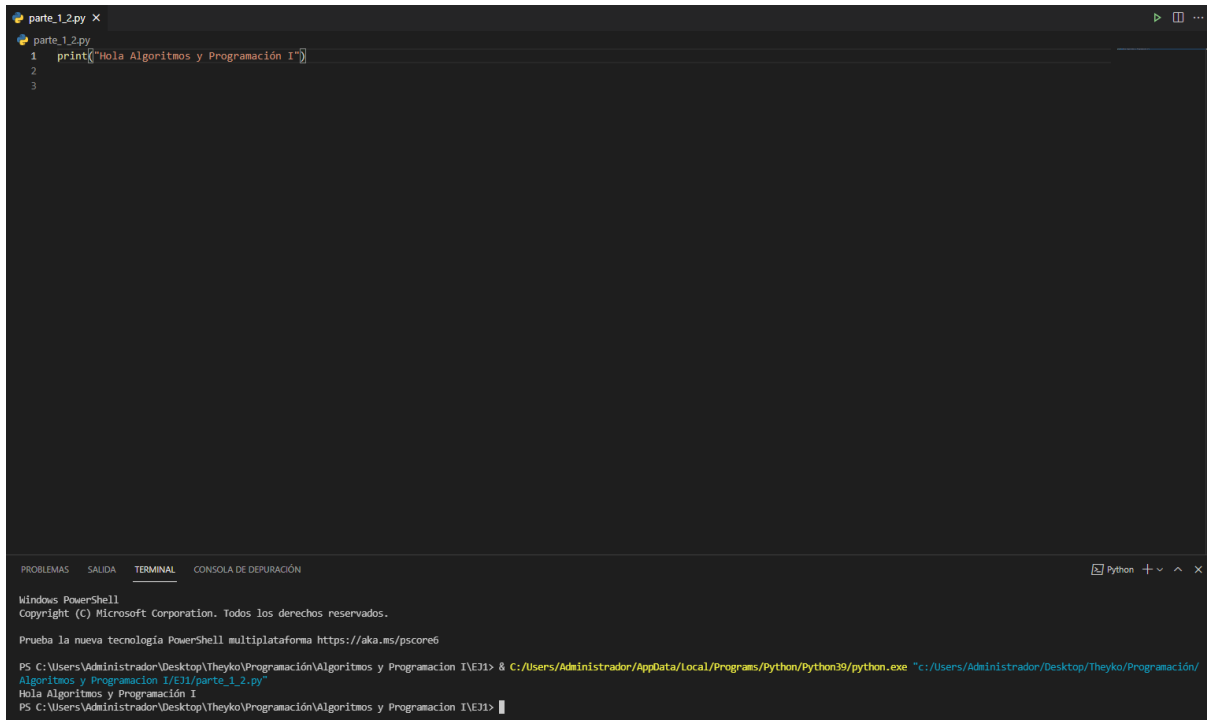
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN

Excepción:
System.ArgumentOutOfRangeException: El valor debe ser mayor que o igual a cero y menor que el tamaño de búfer de consola en dicha dimensión.
Nombre del parámetro: left
Valor actual -1.
en System.Console.SetCursorPosition(Int32 left, Int32 top)
en Microsoft.PowerShell.Internal.VirtualTerminal.set_CursorLeft(Int32 value)
en Microsoft.PowerShell.PSConsoleReadLine.ReallyRender(RenderData renderData, String defaultColor)
en Microsoft.PowerShell.PSConsoleReadLine.ForceRender()
en Microsoft.PowerShell.PSConsoleReadLine.Insert(Char c)
en Microsoft.PowerShell.PSConsoleReadLine.ProcessOneKey(ConsoleKeyInfo key, Dictionary`2 dispatchTable, Boolean ignoreIfNoAction, Object arg)
en Microsoft.PowerShell.PSConsoleReadLine.ReadLine(Runspace runspace, EngineIntrinsics engineIntrinsics)
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/
Algoritmos y Programación I/EJ1/parte_1.2.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/
Algoritmos y Programación I/EJ1/parte_1.2.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/
Algoritmos y Programación I/EJ1/parte_1.2.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> |
```

5) Se debe utilizar la función **print()** la misma imprime por consola lo que le pasemos por parámetro dentro de los paréntesis. Se pueden imprimir todo tipo de datos y concatenar los mismos para imprimir varios a la vez. Python nos permite imprimir por consola cualquier tipo de valor que enviemos por consola sin ningún tipo de ejecución de función y/o programa, como en el caso de la parte 1.1. De esta forma solo imprimimos, pero al utilizar el **print()** podemos utilizar algún valor utilizado en la ejecución de nuestro código.

Extra Parte 1:

- Ejemplo del uso de la función **print()**:



The image shows a screenshot of a Python IDE. The editor window displays a file named `parte_1.2.py` with the following code:

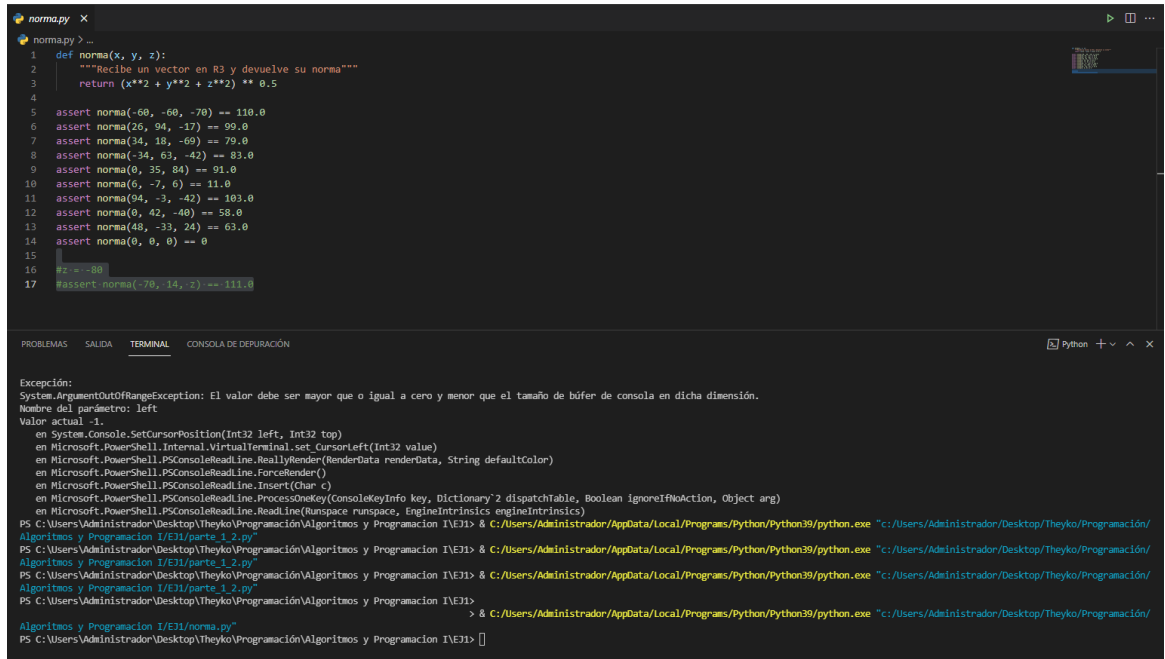
```
1 print("Hola Algoritmos y Programación I")
2
3
```

Below the editor is a terminal window. The terminal output shows the command to run the script and the resulting output:

```
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/EJ1/parte_1.2.py"
Hola Algoritmos y Programación I
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1>
```

Parte 2:

- Captura de la primera ejecución sin descomentar las líneas 16 y 17:

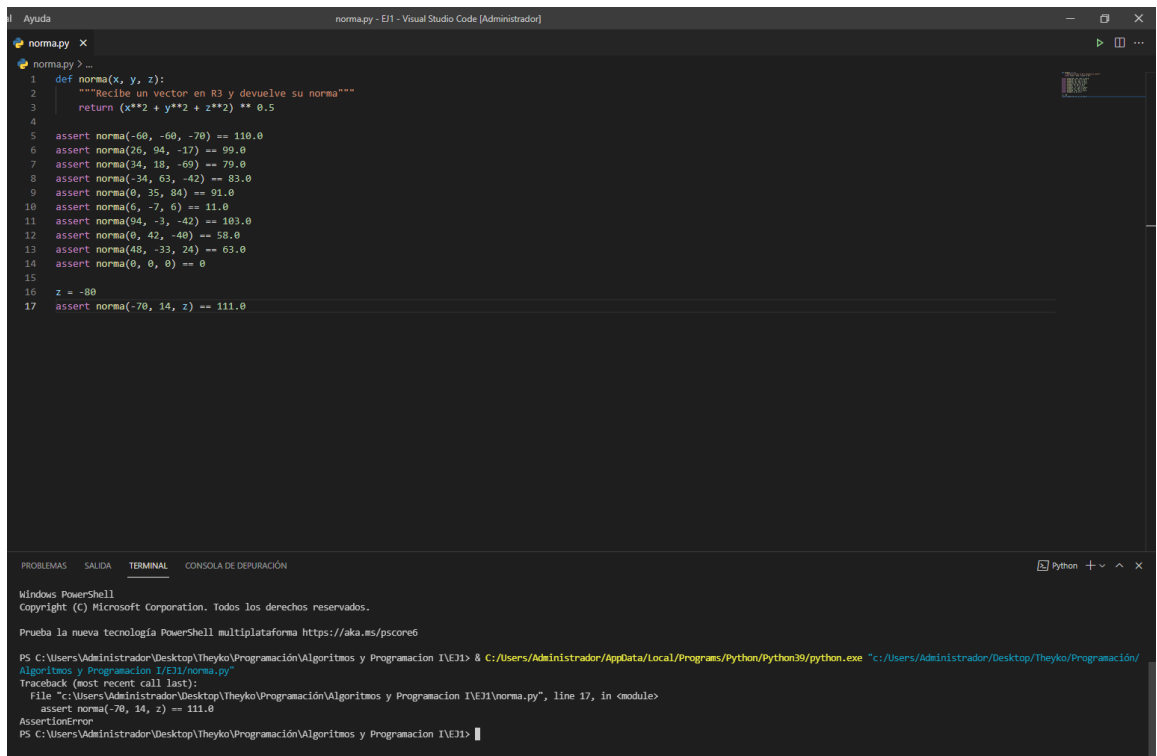


```
norma.py X
norma.py > ...
1 def norma(x, y, z):
2     """Recibe un vector en R3 y devuelve su norma"""
3     return (x**2 + y**2 + z**2) ** 0.5
4
5 assert norma(-60, -60, -70) == 110.0
6 assert norma(26, 94, -17) == 99.0
7 assert norma(34, 18, -69) == 79.0
8 assert norma(-34, 63, -42) == 83.0
9 assert norma(0, 35, 84) == 91.0
10 assert norma(6, -7, 6) == 11.0
11 assert norma(94, -3, -42) == 103.0
12 assert norma(0, 42, -40) == 58.0
13 assert norma(48, -33, 24) == 63.0
14 assert norma(0, 0, 0) == 0
15
16 z = -80
17 assert norma(-70, 14, z) == 111.0

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN Python + ^ x

Excepción:
System.ArgumentOutOfRangeException: El valor debe ser mayor que o igual a cero y menor que el tamaño de búfer de consola en dicha dimensión.
Nombre del parámetro: left
Valor actual: -1.
en System.Console.SetCursorPosition(Int32 left, Int32 top)
en Microsoft.PowerShell.Internal.VirtualTerminal.set_CursorLeft(Int32 value)
en Microsoft.PowerShell.PSConsoleReadLine.ReallyRender(RenderData renderData, String defaultColor)
en Microsoft.PowerShell.PSConsoleReadLine.ForceRender()
en Microsoft.PowerShell.PSConsoleReadLine.Insert(Char c)
en Microsoft.PowerShell.PSConsoleReadLine.ProcessOneKey(ConsoleKeyInfo key, Dictionary`2 dispatchTable, Boolean ignoreIfNoAction, Object arg)
en Microsoft.PowerShell.PSConsoleReadLine.ReadLine(Runspace runspace, EngineIntrinsics engineIntrinsics)
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E11> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/E11/parte_1_2.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E11> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/E11/parte_1_2.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E11> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/E11/parte_1_2.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E11> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/E11/norma.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E11> 
```

- Captura de la segunda ejecución tras descomentar las líneas 16 y 17:



```
norma.py X
norma.py > ...
1 def norma(x, y, z):
2     """Recibe un vector en R3 y devuelve su norma"""
3     return (x**2 + y**2 + z**2) ** 0.5
4
5 assert norma(-60, -60, -70) == 110.0
6 assert norma(26, 94, -17) == 99.0
7 assert norma(34, 18, -69) == 79.0
8 assert norma(-34, 63, -42) == 83.0
9 assert norma(0, 35, 84) == 91.0
10 assert norma(6, -7, 6) == 11.0
11 assert norma(94, -3, -42) == 103.0
12 assert norma(0, 42, -40) == 58.0
13 assert norma(48, -33, 24) == 63.0
14 assert norma(0, 0, 0) == 0
15
16 z = -80
17 assert norma(-70, 14, z) == 111.0

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN Python + ^ x

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.es/pscore6

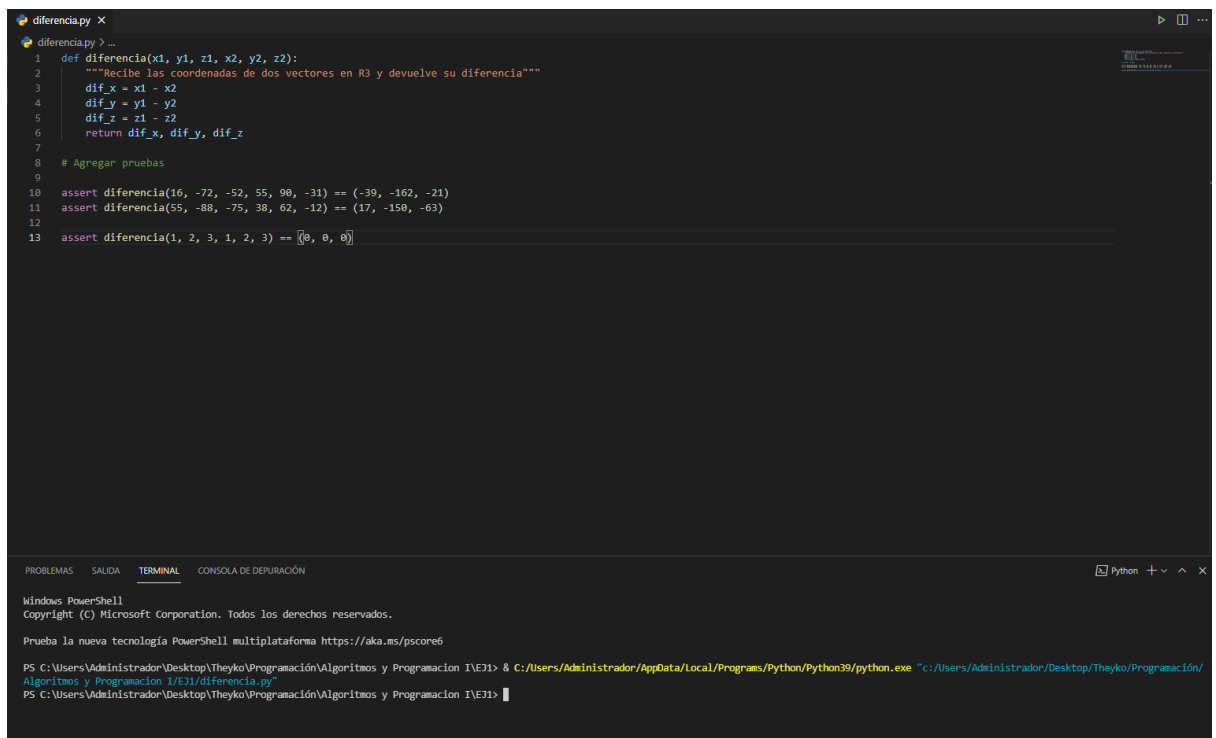
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E11> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/E11/norma.py"
Traceback (most recent call last):
  File "c:/Users/Administrador/Desktop\Theyko\Programación\Algoritmos y Programación I\E11\norma.py", line 17, in <module>
    assert norma(-70, 14, z) == 111.0
AssertionError
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E11> 
```

5)

1. La salida del programa es un error. (AssertionError)
2. Al leer la consola podemos saber que el error está en la línea 17.
3. La instrucción **assert** nos permite expresar una condición que debe ser cierta siempre, y en caso de no serlo se interrumpe la ejecución del programa. En este caso el valor de **Z** no puede ser el indicado para que al ejecutarse la función **norma()** el valor de retorno sea = 111.00.
4. El valor de **Z** para solucionar el problema y que el programa no arroje ningún error puede ser **Z=85 / Z=-85**

Parte 3:

- Captura de la ejecución assert con los casos pedidos:



```
diferencia.py X
diferencia.py > ...
1 def diferencia(x1, y1, z1, x2, y2, z2):
2     """Recibe las coordenadas de dos vectores en R3 y devuelve su diferencia"""
3     dif_x = x1 - x2
4     dif_y = y1 - y2
5     dif_z = z1 - z2
6     return dif_x, dif_y, dif_z
7
8 # Agregar pruebas
9
10 assert diferencia(16, -72, -52, 55, 90, -31) == (-39, -162, -21)
11 assert diferencia(55, -88, -75, 38, 62, -12) == (17, -150, -63)
12
13 assert diferencia(1, 2, 3, 1, 2, 3) == [0, 0, 0]
```

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN

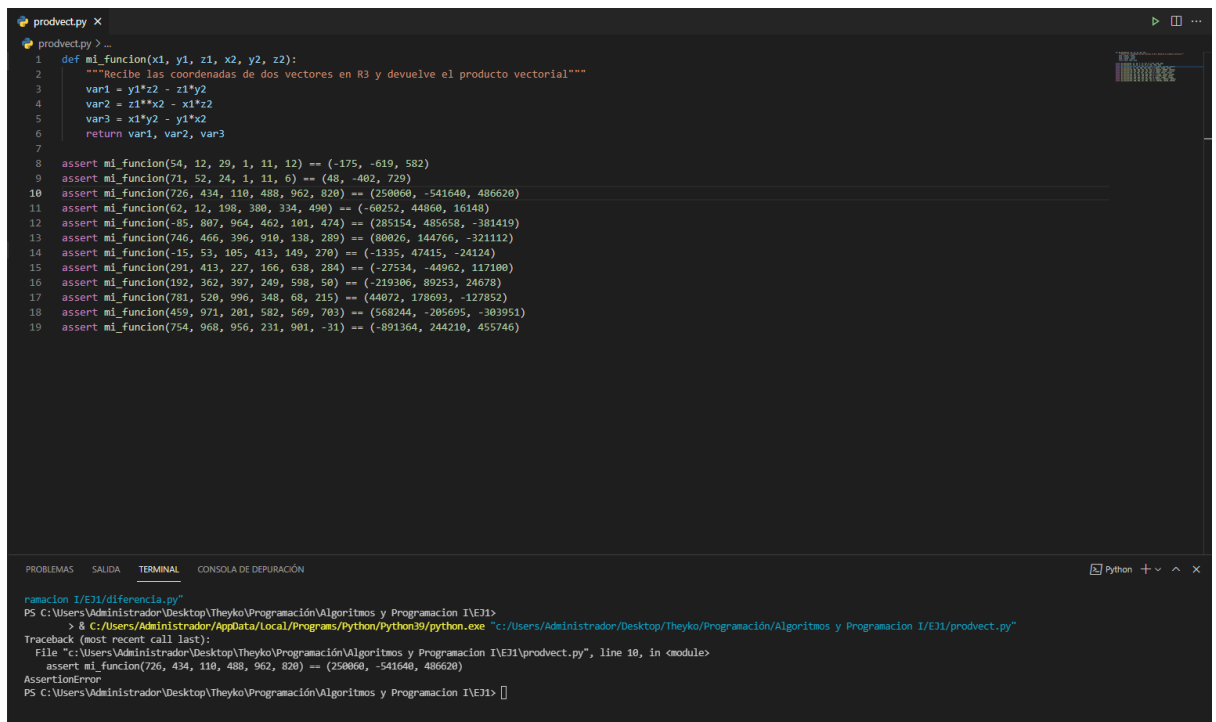
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.
Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/powershell>

PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1> & C:/Users/Administrador/AppData/Local/Programs/Python/Python39/python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/EJ1/diferencia.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\EJ1>

- 4) No se detectó ningún error como se puede observar en la consola, en este caso la instrucción **assert** no corta la ejecución del programa ya que las condiciones son verdaderas.

Parte 4:

- Captura de la primer ejecución:



```
prodvect.py X
prodvect.py >
1 def mi_funcion(x1, y1, z1, x2, y2, z2):
2     """Recibe las coordenadas de dos vectores en R3 y devuelve el producto vectorial"""
3     var1 = y1*z2 - z1*y2
4     var2 = z1*x2 - x1*z2
5     var3 = x1*y2 - y1*x2
6     return var1, var2, var3
7
8 assert mi_funcion(54, 12, 29, 1, 11, 12) == (-175, -619, 582)
9 assert mi_funcion(71, 52, 24, 1, 11, 6) == (48, -402, 729)
10 assert mi_funcion(726, 434, 110, 488, 962, 820) == (250060, -541640, 486620)
11 assert mi_funcion(62, 12, 198, 380, 334, 490) == (-60252, 44860, 16148)
12 assert mi_funcion(-85, 807, 964, 462, 101, 474) == (285154, 485658, -381419)
13 assert mi_funcion(746, 466, 396, 910, 138, 289) == (80026, 144766, -321112)
14 assert mi_funcion(-15, 53, 105, 413, 149, 270) == (-1335, 47415, -24124)
15 assert mi_funcion(201, 413, 227, 166, 638, 204) == (-27534, -44962, 117100)
16 assert mi_funcion(192, 362, 397, 249, 590, 50) == (-219206, 80253, 24678)
17 assert mi_funcion(781, 520, 996, 348, 68, 215) == (44072, 178693, -127852)
18 assert mi_funcion(459, 971, 201, 582, 560, 703) == (568244, -205695, -303951)
19 assert mi_funcion(754, 968, 956, 231, 901, -31) == (-891364, 244210, 455746)

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN
Python + ^ X

ramacion I/E31/diferencia.py"
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E31>
> & C:\Users\Administrador\AppData\Local\Programs\Python\Python39\python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/E31/prodvect.py"
Traceback (most recent call last):
  File "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/E31/prodvect.py", line 10, in <module>
    assert mi_funcion(726, 434, 110, 488, 962, 820) == (250060, -541640, 486620)
AssertionError
PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E31>
```

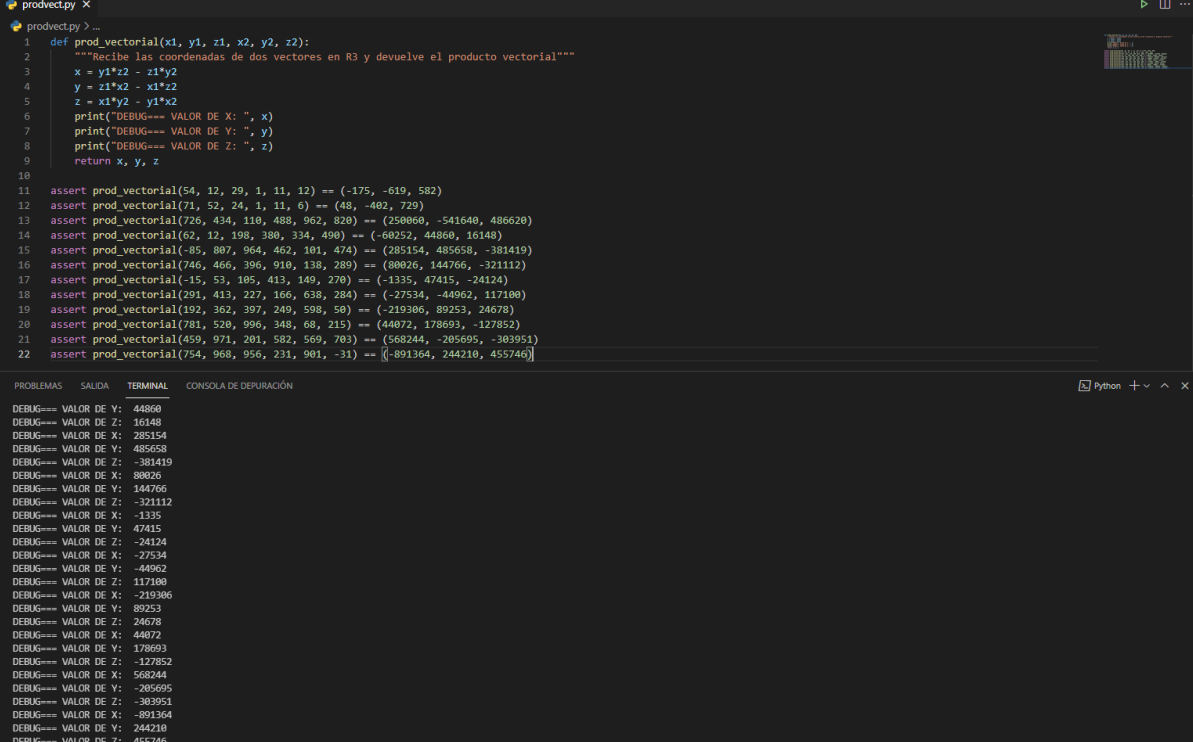
- 4) Muestra un AssertionError en la línea 10.

5) Ejecución en la depuración agregando la función **print()** dentro de la función principal del programa:

```
product.py > mi_funcion
1 def mi_funcion(x1, y1, z1, x2, y2, z2):
2     """Recibe las coordenadas de dos vectores en R3 y devuelve el producto vectorial"""
3     var1 = y1*z2 - z1*y2
4     var2 = z1*x2 - x1*z2
5     var3 = x1*y2 - y1*x2
6     print("DEBUG== VALOR DE X: ", var1)
7     print("DEBUG== VALOR DE Y: ", var2)
8     print("DEBUG== VALOR DE Z: ", var3)
9     return var1, var2, var3
10
11 assert mi_funcion(54, 12, 29, 1, 11, 12) == (-175, -619, 582)
12 assert mi_funcion(71, 62, 24, 1, 11, 6) == (48, -402, 729)
13 assert mi_funcion(726, 434, 110, 488, 962, 820) == (250060, -541640, 486620)
14 assert mi_funcion(62, 12, 198, 380, 334, 490) == (-60252, 44860, 16148)
15 assert mi_funcion(-85, 807, 964, 462, 101, 474) == (285154, 485658, -381419)
16 assert mi_funcion(746, 466, 396, 910, 138, 289) == (80026, 144766, -321112)
17 assert mi_funcion(-15, 53, 105, 413, 149, 270) == (-1335, 47415, -24124)
18 assert mi_funcion(291, 413, 227, 166, 638, 284) == (-27534, -44962, 117100)
19 assert mi_funcion(192, 362, 397, 249, 586, 50) == (-219306, 89253, 24678)
20 assert mi_funcion(781, 520, 996, 348, 68, 215) == (-44072, 178693, -127852)
21 assert mi_funcion(459, 971, 201, 582, 569, 703) == (568244, -205695, -303951)
22 assert mi_funcion(754, 968, 956, 231, 901, -31) == (-891364, 244210, 455746)
```

Con esta depuración podemos observar más específicamente de qué trata el error, además de la información que ya conocíamos (indicada en el inciso 4, que tras agregar 3 líneas para depurar el error se muestra en la línea 13, mismo código pero distinta línea por el código extra para depurar). Tras observar la información extra y el código nuevamente podemos observar que donde se define la variable **var2** hay un operador lógico extra que no debería estar (*). Quitando dicho operador el programa corre sin ningún problema. (foto en inciso 6)

6) Ejecución del programa tras la depuración, arreglo del error y renombre de las funciones y variables de forma representativas:



```
prodvect.py X
prodvect.py >...
1 def prod_vectorial(x1, y1, z1, x2, y2, z2):
2     """Recibe las coordenadas de dos vectores en R3 y devuelve el producto vectorial"""
3     x = y1*z2 - z1*y2
4     y = z1*x2 - x1*z2
5     z = x1*y2 - y1*x2
6     print("DEBUG=== VALOR DE X: ", x)
7     print("DEBUG=== VALOR DE Y: ", y)
8     print("DEBUG=== VALOR DE Z: ", z)
9     return x, y, z
10
11 assert prod_vectorial(54, 12, 29, 1, 11, 12) == (-175, -619, 582)
12 assert prod_vectorial(71, 52, 24, 1, 11, 6) == (48, -482, 729)
13 assert prod_vectorial(726, 434, 110, 488, 962, 820) == (258066, -541640, 486620)
14 assert prod_vectorial(62, 12, 198, 388, 334, 490) == (-68252, 44868, 16148)
15 assert prod_vectorial(-85, 807, 964, 462, 101, 474) == (285154, 485658, -381419)
16 assert prod_vectorial(746, 466, 396, 910, 138, 289) == (80026, 144766, -321112)
17 assert prod_vectorial(-15, 53, 105, 413, 149, 270) == (-1335, 47415, -24124)
18 assert prod_vectorial(291, 413, 227, 166, 638, 284) == (-27534, -44962, 117100)
19 assert prod_vectorial(192, 362, 397, 249, 598, 50) == (-219306, 89253, 24678)
20 assert prod_vectorial(781, 520, 996, 348, 68, 215) == (44072, 178693, -127852)
21 assert prod_vectorial(459, 971, 201, 582, 569, 703) == (568244, -205695, -383951)
22 assert prod_vectorial(754, 968, 956, 231, 901, -31) == (-891364, 244210, 455746)
```

PROBLEMAS	SALIDA	TERMINAL	CONSOLA DE DEPURACIÓN
			DEBUG=== VALOR DE Y: 44868
			DEBUG=== VALOR DE Z: 16148
			DEBUG=== VALOR DE X: 285154
			DEBUG=== VALOR DE Y: 485658
			DEBUG=== VALOR DE Z: -381419
			DEBUG=== VALOR DE X: 80026
			DEBUG=== VALOR DE Y: 144766
			DEBUG=== VALOR DE Z: -321112
			DEBUG=== VALOR DE X: -1335
			DEBUG=== VALOR DE Y: 47415
			DEBUG=== VALOR DE Z: -24124
			DEBUG=== VALOR DE X: -27534
			DEBUG=== VALOR DE Y: -44962
			DEBUG=== VALOR DE Z: 117100
			DEBUG=== VALOR DE X: -219306
			DEBUG=== VALOR DE Y: 89253
			DEBUG=== VALOR DE Z: 24678
			DEBUG=== VALOR DE X: 44072
			DEBUG=== VALOR DE Y: 178693
			DEBUG=== VALOR DE Z: -127852
			DEBUG=== VALOR DE X: 568244
			DEBUG=== VALOR DE Y: -205695
			DEBUG=== VALOR DE Z: -383951
			DEBUG=== VALOR DE X: -891364
			DEBUG=== VALOR DE Y: 244210
			DEBUG=== VALOR DE Z: 455746

PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E31>

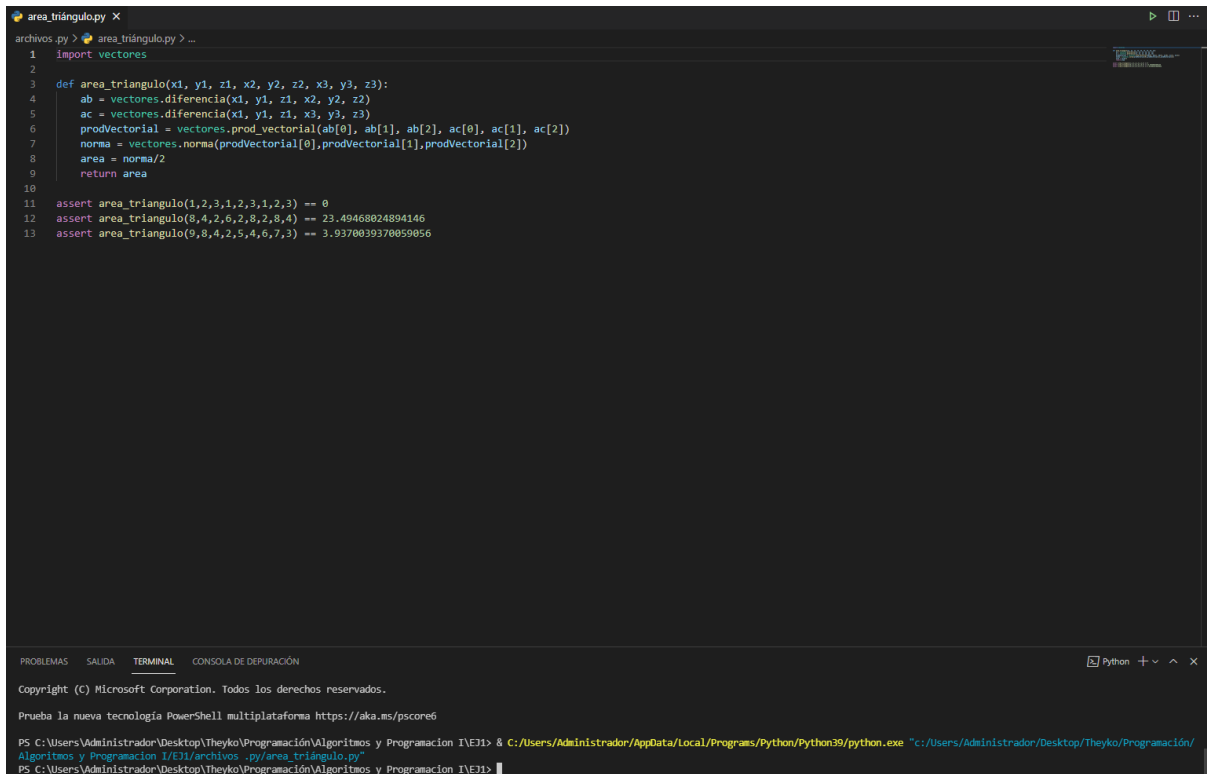
Es importante definir las funciones y variables de forma representativa para así hacer mucho más entendible nuestro código, es una buena práctica para lograr que cualquier persona que lo lea pueda entender sin necesidad de consultar o revisar arduamente que hace cada función o que significa cada variable. Además se suelen hacer comentarios en ciertas partes del código para ayudar a este entendimiento. (ejemplo: las pre y post condiciones de las funciones)

7) se puede escribir el cuerpo de la función en una sola línea realizando, para este caso se puede definir una variable en forma de vector, donde cada componente del vector será las operaciones lógicas que antes teníamos definidas como variables. De esta forma acortamos la longitud del código, aunque no suele ser una buena práctica ya que se tiende a tener más errores y vuelve el código más confuso de leer.

Parte 5:

4) Es importante reutilizar funciones para acortar y simplificar nuestro código. La idea es generar funciones que son bloques de código reutilizables y modularizar sistema de tal forma que podamos simplemente tener un código “main” como se lo conoce comúnmente corto, limpio y entendible.

Ejecución del programa:



```
area_triángulo.py X
archivos.py > area_triángulo.py > ...
1 import vectores
2
3 def area_triángulo(x1, y1, z1, x2, y2, z2, x3, y3, z3):
4     ab = vectores.diferencia(x1, y1, z1, x2, y2, z2)
5     ac = vectores.diferencia(x1, y1, z1, x3, y3, z3)
6     prodVectorial = vectores.prod_vectorial(ab[0], ab[1], ab[2], ac[0], ac[1], ac[2])
7     norma = vectores.norma(prodVectorial[0], prodVectorial[1], prodVectorial[2])
8     area = norma/2
9     return area
10
11 assert area_triángulo(1,2,3,1,2,3,1,2,3) == 0
12 assert area_triángulo(8,4,2,6,2,8,2,8,4) == 23.49468024894146
13 assert area_triángulo(9,8,4,2,5,4,6,7,3) == 3.9370039370059056
```

PROBLEMAS SALIDA TERMINAL CONSOLA DE DEPURACIÓN Python + - ^ X

Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E31> & C:\Users\Administrador\AppData\Local\Programs\Python\Python39\python.exe "c:/Users/Administrador/Desktop/Theyko/Programación/Algoritmos y Programación I/E31/archivos .py/area_triángulo.py"

PS C:\Users\Administrador\Desktop\Theyko\Programación\Algoritmos y Programación I\E31>