

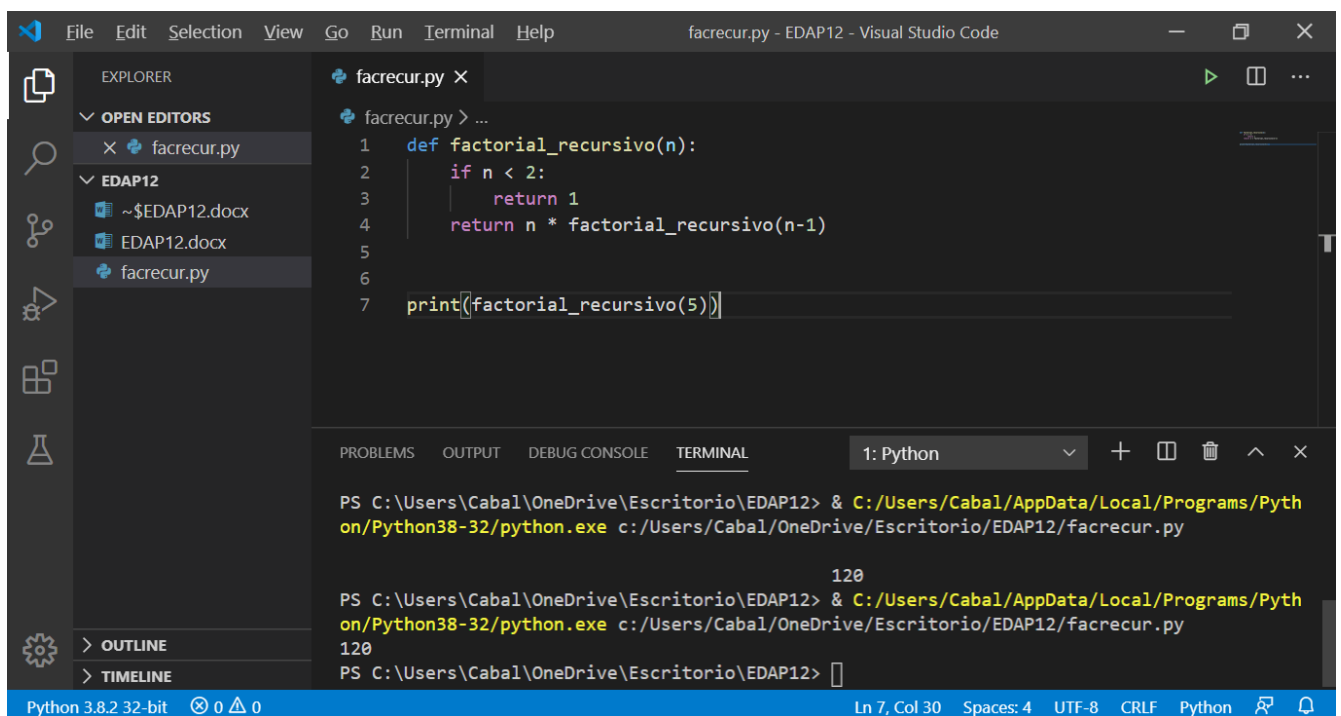
Práctica 12

Introducción

En esta práctica se presenta el termino de recursividad de las funciones que resulta muy útil en la programación que consiste básicamente en que una función se mande llamar a sí misma al menos una vez dentro de las instrucciones de esta

➤ Desarrollo (con ejercicios)

- Para el primer caso se ve el factorial recursivo en el cual hay 2 casos peculiares, cuando $n = 1$ ó $n = 0$, en el cual lo que regresa la función es el valor de uno

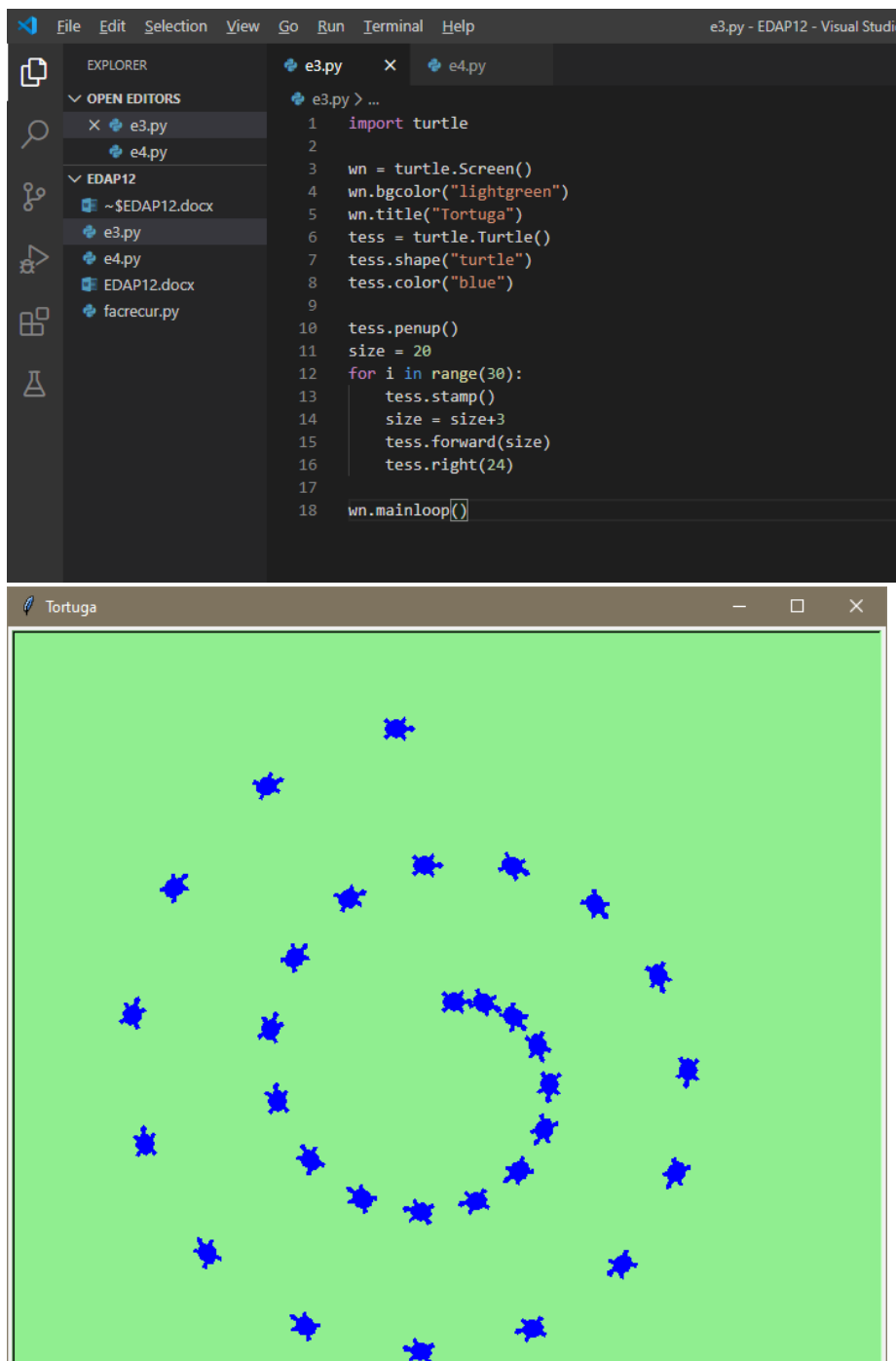


The screenshot shows the Visual Studio Code interface with a file named `facrecur.py` open. The code defines a recursive function `factorial_recursivo(n)` and calls it with `5`. The terminal output shows the execution of the script, resulting in the value `120`.

```
def factorial_recursivo(n):  
    if n < 2:  
        return 1  
    return n * factorial_recursivo(n-1)  
  
print(factorial_recursivo(5))
```

```
PS C:\Users\Cabal\OneDrive\Escritorio\EDAP12> & C:/Users/Cabal/AppData/Local/Programs/Python/Python38-32/python.exe c:/Users/Cabal/OneDrive/Escritorio/EDAP12/facrecur.py  
120  
PS C:\Users\Cabal\OneDrive\Escritorio\EDAP12> & C:/Users/Cabal/AppData/Local/Programs/Python/Python38-32/python.exe c:/Users/Cabal/OneDrive/Escritorio/EDAP12/facrecur.py  
120  
PS C:\Users\Cabal\OneDrive\Escritorio\EDAP12>
```

- En el segundo ejercicio se hizo un ejemplo Las huellas de tortuga que va de una tortuga que realiza un camino en espiral y a su vez va dejando huellas de esta



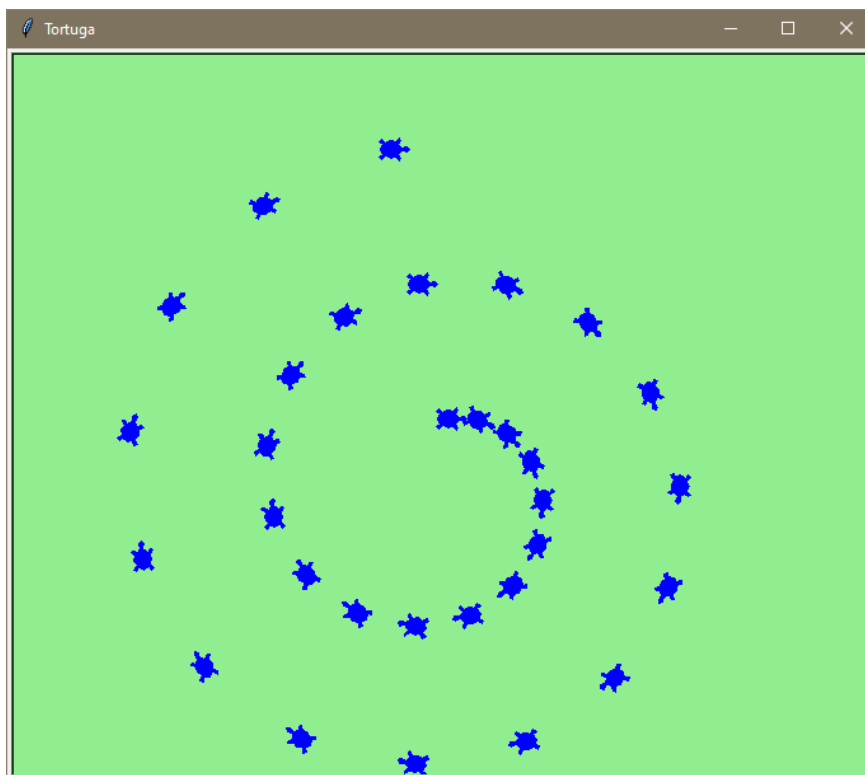
- Ahora se hace pero con un código que implica que sea de manera recursiva el recorrido que hace la tortuga

```
File Edit Selection View Go Run Terminal Help e4.py - EDAP12 - Visual Studio Code

EXPLORER
  OPEN EDITORS
    e3.py
    e4.py
  EDAP12
    ~$EDAP12.docx
    e3.py
    e4.py
    EDAP12.docx
    facrecur.py

e4.py > ...
1 import turtle
2
3 def recorrido_recursivo(tortuga, espacio, huella):
4     if huella>0:
5         tortuga.stamp()
6         espacio = espacio +3
7         tortuga.forward(espacio)
8         tortuga.right(24)
9         recorrido_recursivo(tortuga, espacio, huella-1)
10
11
12 wn = turtle.Screen()
13 wn.bgcolor("lightgreen")
14 wn.title("Tortuga")
15 tess = turtle.Turtle()
16 tess.shape("turtle")
17 tess.color("blue")
18
19 tess.penup()
20 recorrido_recursivo(tess, 20, 30)
21
22 wn.mainloop()
```

- El recorrido resulta ser el mismo pero está implementado de forma recursiva



➤ Conclusión

- **Caballero Hernandez Juan Daniel**

El objetivo de esta práctica ya lo hemos puesto en ejecución en otros problemas pero por ahora python sigue asombrándome con todo lo que se puede hacer ya que es bastante versátil sin embargo nada es tan

sorprendente para no tener algún costo, y en este caso el uso de la recursividad puede llegar a ser mas tardado debido a la simplicidad tan compleja que conlleva