|  |  |
| --- | --- |
|  | **Carátula para entrega de prácticas** |
| Facultad de Ingeniería | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | M.I. Marco Antonio Martinez Quintana |
| *Asignatura:* | estructuras de datos y algoritmos |
| *Grupo:* | 17 |
| *No de Práctica(s):* | 12 |
| *Integrante(s):* | Villanueva Corona Miguel Angel |
| *No. de Equipo de cómputo empleado:* | 36 |
| *No. de Lista o Brigada:* | 41 |
| *Semestre:* | 2020-2 |
| *Fecha de entrega:* | 28 de abril del 2020 |
| *Observaciones:* |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

***Objetivos***

El objetivo de esta guía es aplicar el concepto de recursividad para la solución de problemas.

***Introducción***

***Recursividad***

La recursividad se basa en dividir un problema en sub problemas mas pequeños los cuales llegan a ser fácilmente resueltos por su forma trivial de verlos.

Para aplicar la recursividad se tienen que seguir estas reglas:

1. Debe de haber uno o más casos base.
2. La expansión debe terminar en un caso base.
3. La función se debe llamar a sí misma.

Algunos ejemplos de recursividad son

1. Factorial
2. Huellas de tortuga
3. Serie de Fibonacci

***Desarrollo***

*Factorial*

El factorial es un claro ejemplo de recursividad ya que divide por factores los números que se van a multiplicar primero empieza con un numero y después lo multiplica por su antecesor asi consecutivamente hasta llagar al 1.

CODIGO

#Factorial no recursivo

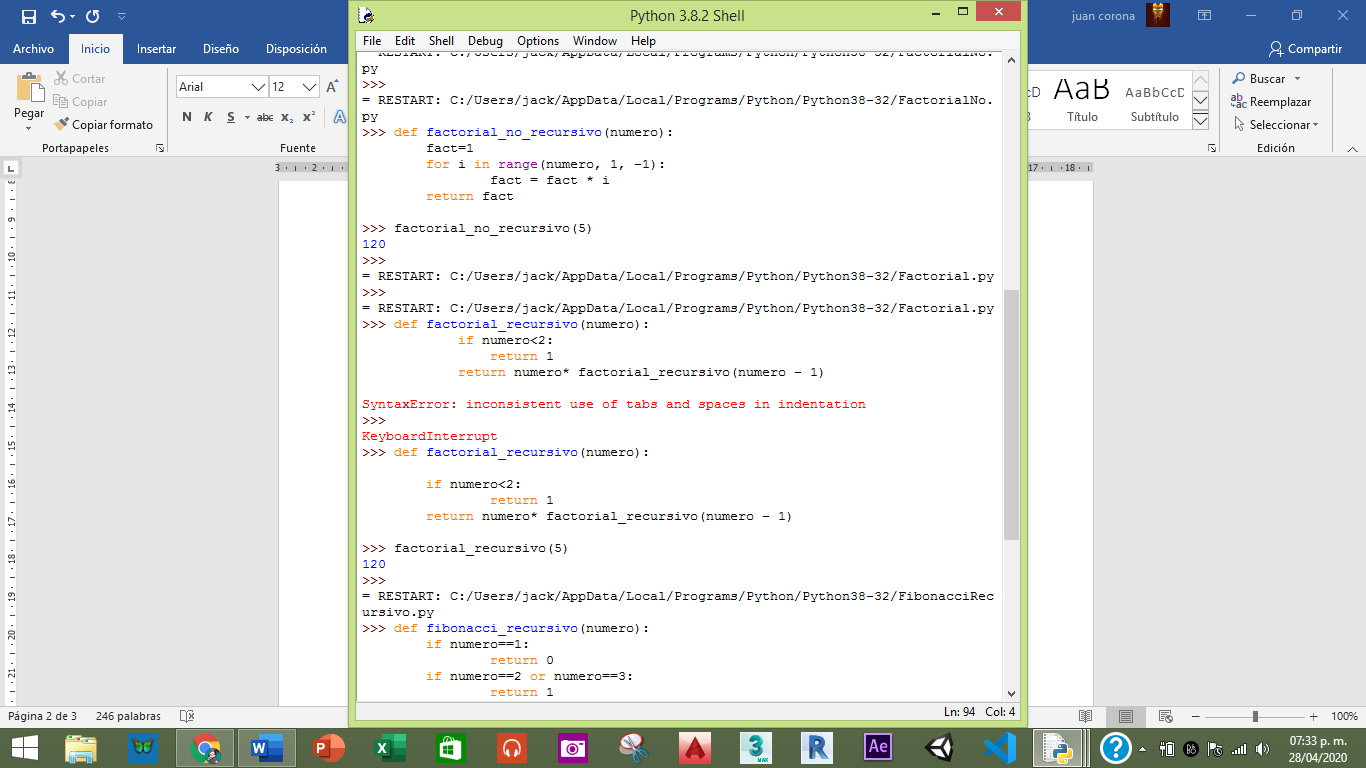
def factorial\_no\_recursivo(numero):

fact=1

for i in range(numero, 1, -1):

fact = fact \* i

return fact

factorial\_no\_recursivo(5)

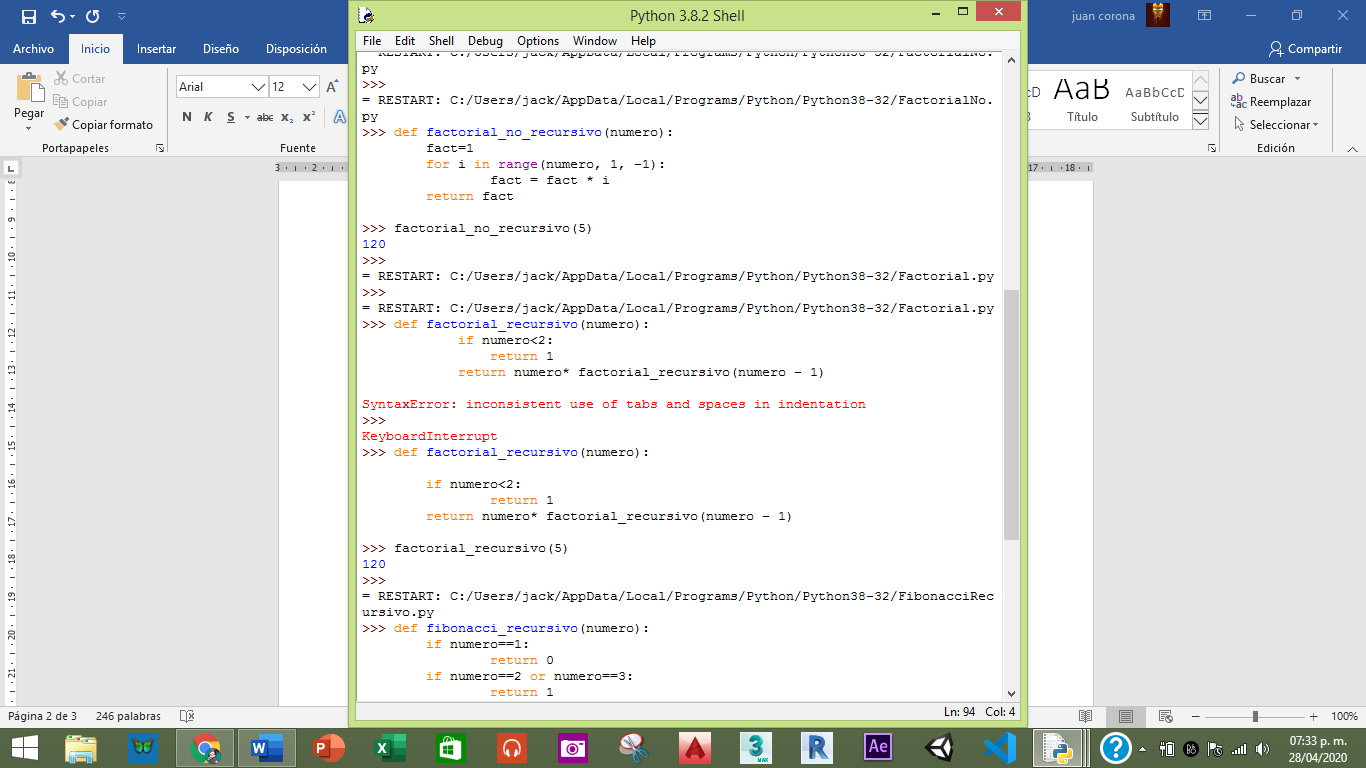
#Factorial Recursivo

def factorial\_recursivo(numero):

if numero<2:

return 1

return numero\* factorial\_recursivo(numero - 1)

factorial\_recursivo(5)

*Huellas de tortuga*

En este ejemplo se hace referencia al supuesto rastro que dejaría una tortuga si esta se moviera haciendo espirales

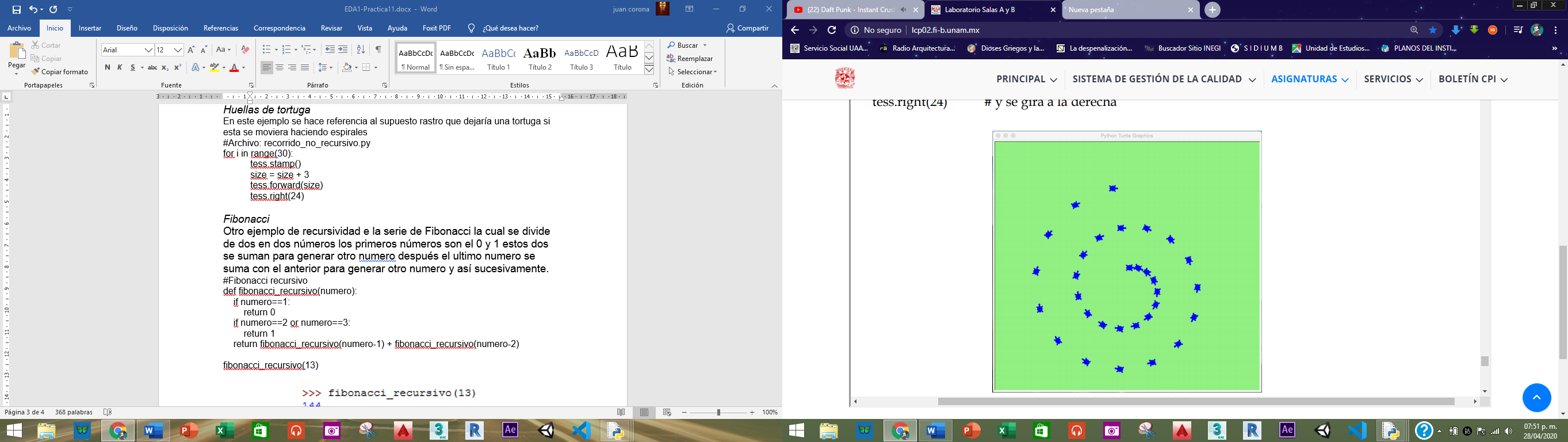
# recorrido\_no\_recursivo

for i in range(30):

tess.stamp()

size = size + 3

tess.forward(size)

tess.right(24)

#recorrido \_recursivo

def recorrido\_recursivo(tortuga, espacio, huellas):

if huellas > 0:

tortuga.stamp()

espacio = espacio + 3

tortuga.forward(espacio)

tortuga.right(24)

recorrido\_recursivo(tortuga, espacio, huellas-1)

*Fibonacci*

Otro ejemplo de recursividad e la serie de Fibonacci la cual se divide de dos en dos números los primeros números son el 0 y 1 estos dos se suman para generar otro numero después el ultimo numero se suma con el anterior para generar otro numero y así sucesivamente.

#Fibonacci recursivo

def fibonacci\_recursivo(numero):

if numero==1:

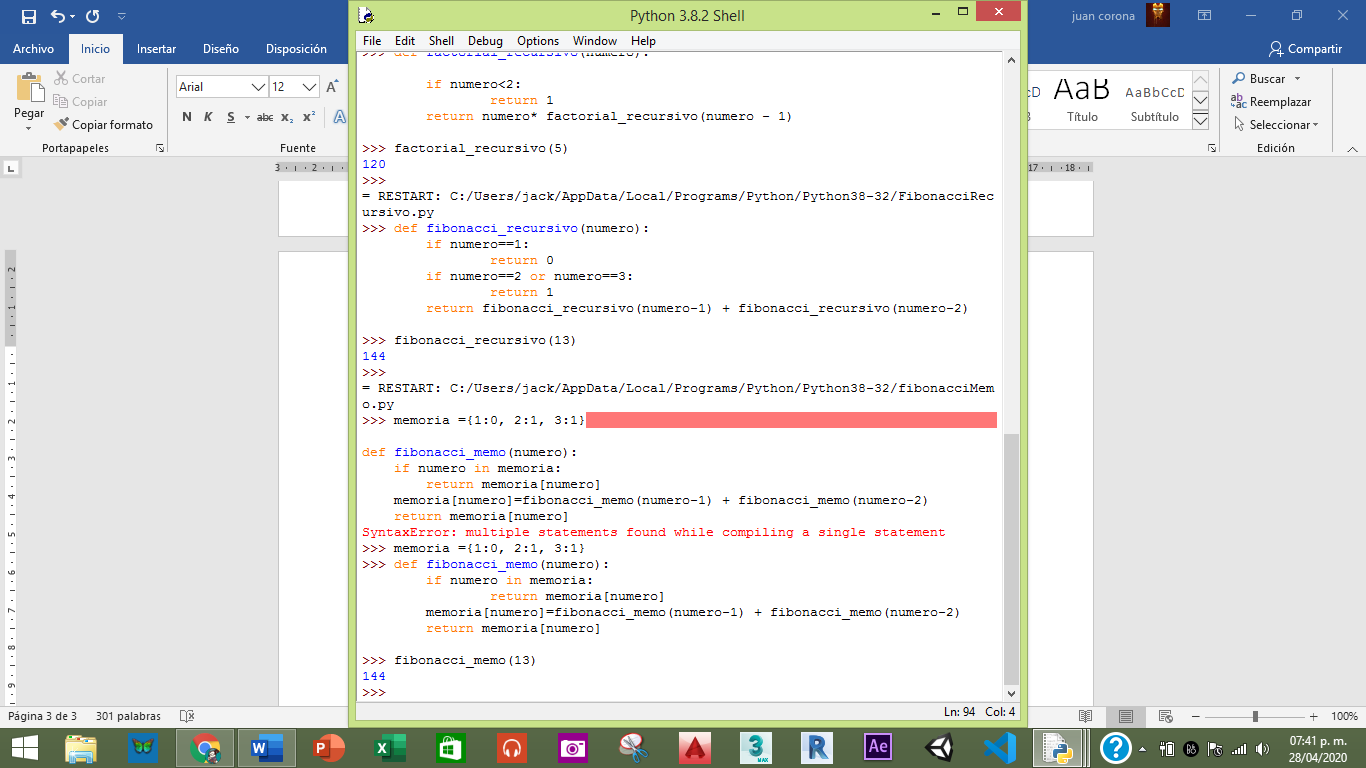
return 0

if numero==2 or numero==3:

return 1

return fibonacci\_recursivo(numero-1) + fibonacci\_recursivo(numero-2)

fibonacci\_recursivo(13)



#Fibonacci memoria

memoria ={1:0, 2:1, 3:1}

def fibonacci\_memo(numero):

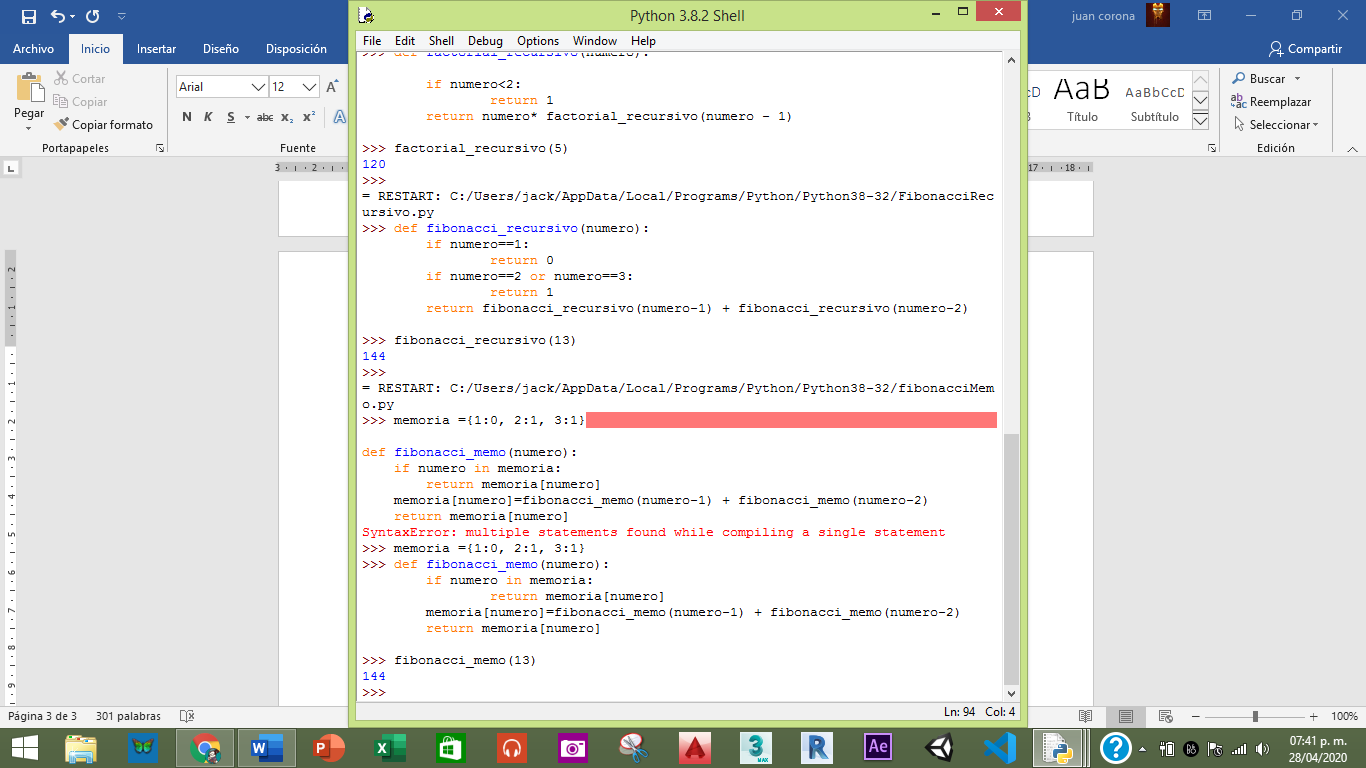
if numero in memoria:

return memoria[numero]

memoria[numero]=fibonacci\_memo(numero-1) + fibonacci\_memo(numero-2)

return memoria[numero]

fibonacci\_memo(13)



***Conclusión***

En conclusión la recursividad es muy útil para hacer que un problema sea algo muy simple de resolver hasta la magnitud de decir que es algo que se pudiese resolver solo ya que las soluciones pueden ser triviales.

***Bibliografía***

<http://lcp02.fi-b.unam.mx/>