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

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | M.I. Marco Antonio Martínez Quintana |
| *Asignatura:* | Estructura de Datos y Algoritmos I |
| *Grupo:* | 17 |
| *No de Práctica(s):* | 12 |
| *Integrante(s):* | Alejandro Romero Ramírez |
| *No. de Equipo de cómputo empleado:* |  |
| *No. de Lista o Brigada:* | 33 |
| *Semestre:* | 2020-2 |
| *Fecha de entrega:* | Martes 21 de abril, 2020 |
| *Observaciones:* |  |
|  |  |

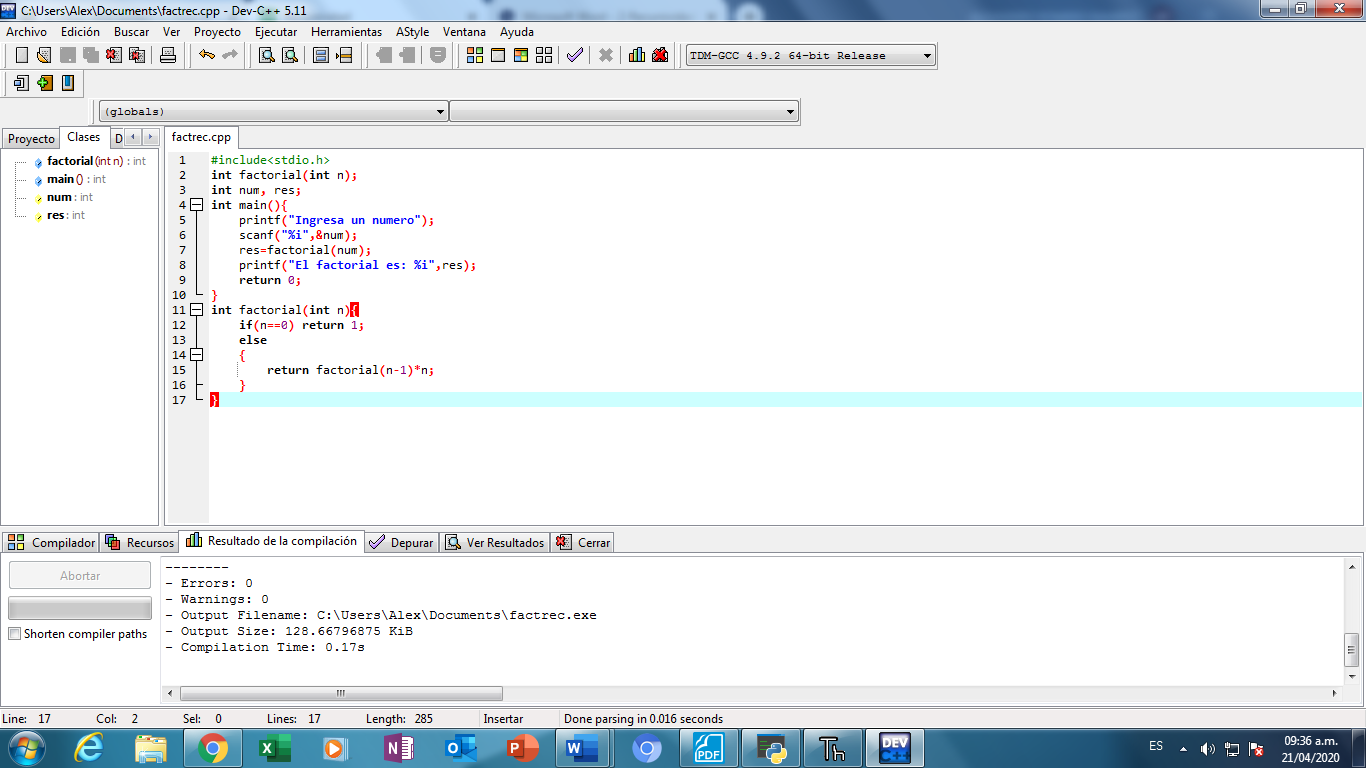
CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

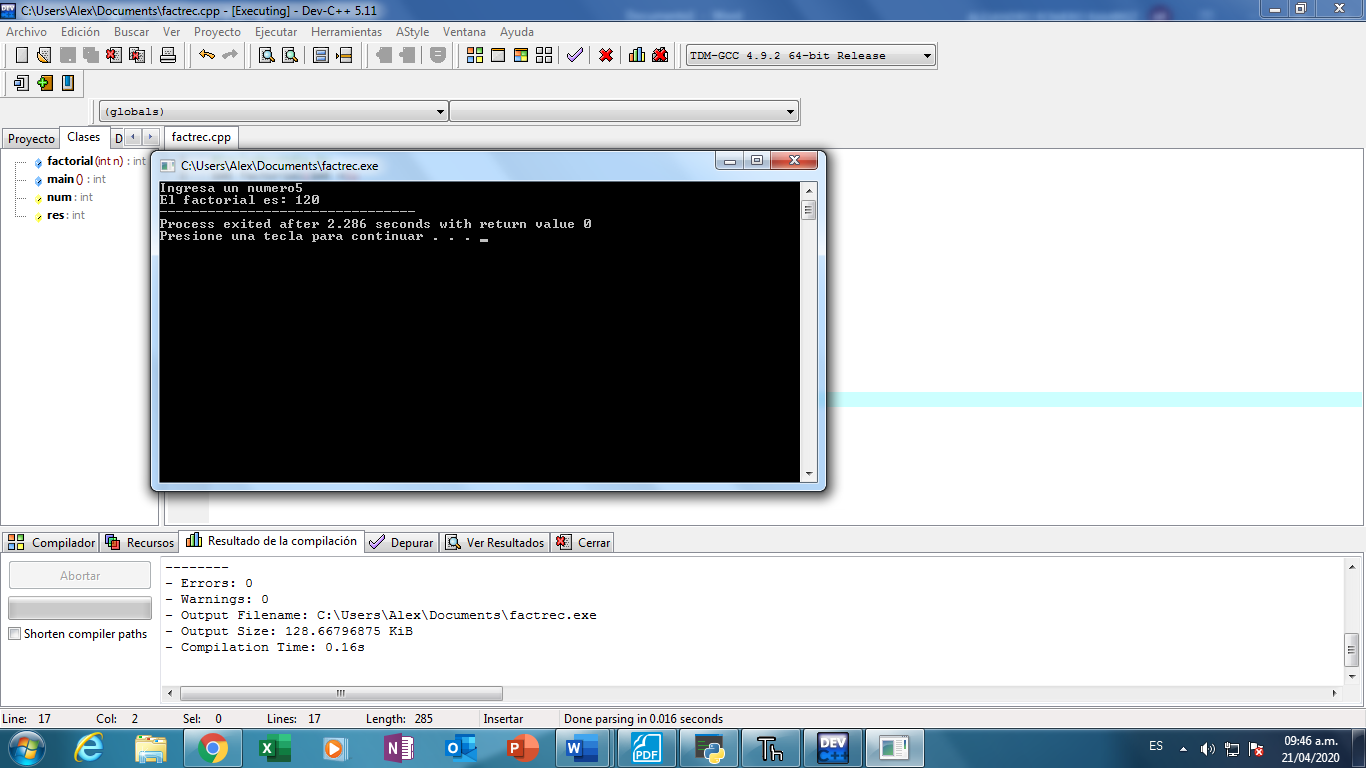
**INTRODUCCIÓN**

La recursividad es el concepto por medio del cual el diseño del algoritmo se utiliza en su propio diseño. Para diseñar un programa con recursividad, el procedimiento es el siguiente:

1. La función procedimiento se llama a sí misma.
2. El problema computacional se resuelve tratando el mismo problema, pero de tamaño menor.
3. El caso base se alcanza eventualmente a través de la disminución del tamaño del problema.

La recursividad se puede aplicar, por ejemplo, para calcular el factorial de un número en lenguaje C, llamando la función a sí misma como se muestra aquí:





En la vida cotidiana, la recursividad puede aplicarse en una artesanía tradicional rusa llamada matroshka, que consiste en muñecas que esconden en su interior otra muñeca de manera sucesiva a medida que su tamaño va disminuyendo hasta que ya no se esconde nada.



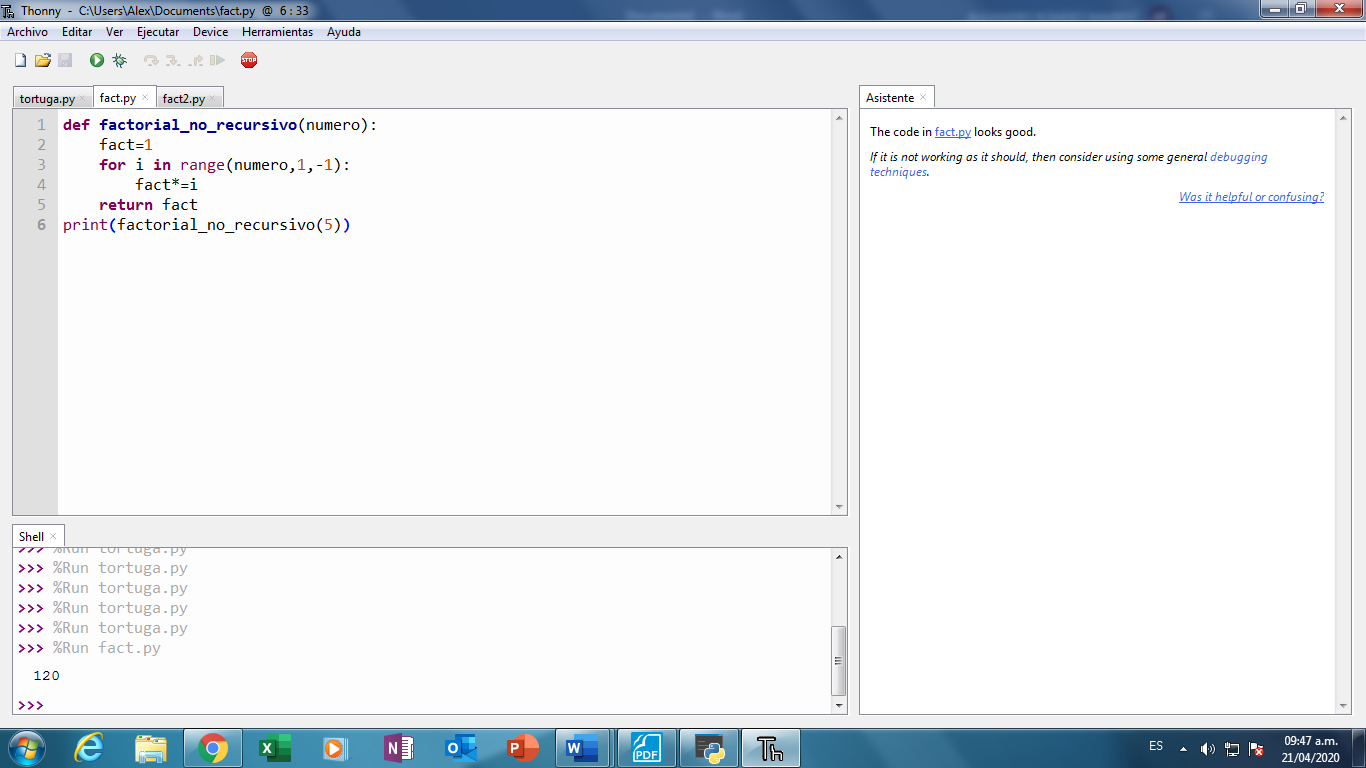
**OBJETIVO**

Aplicar el concepto de recursividad en la resolución de problemas computacionales.

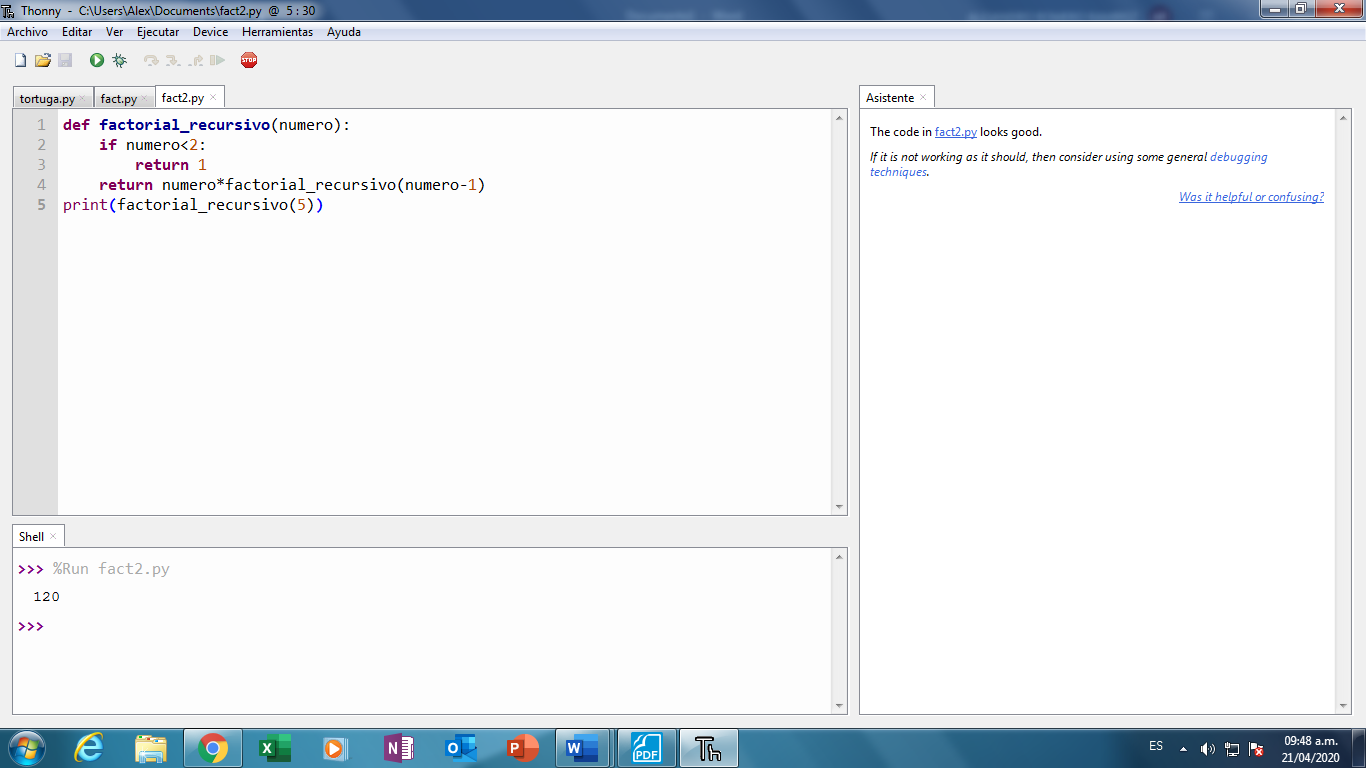
**DESARROLLO**

En Python, como en lenguaje C, el problema de cálculo del factorial puede resolverse de dos maneras distintas:

De forma iterativa con un ciclo for, como muestra el siguiente código:

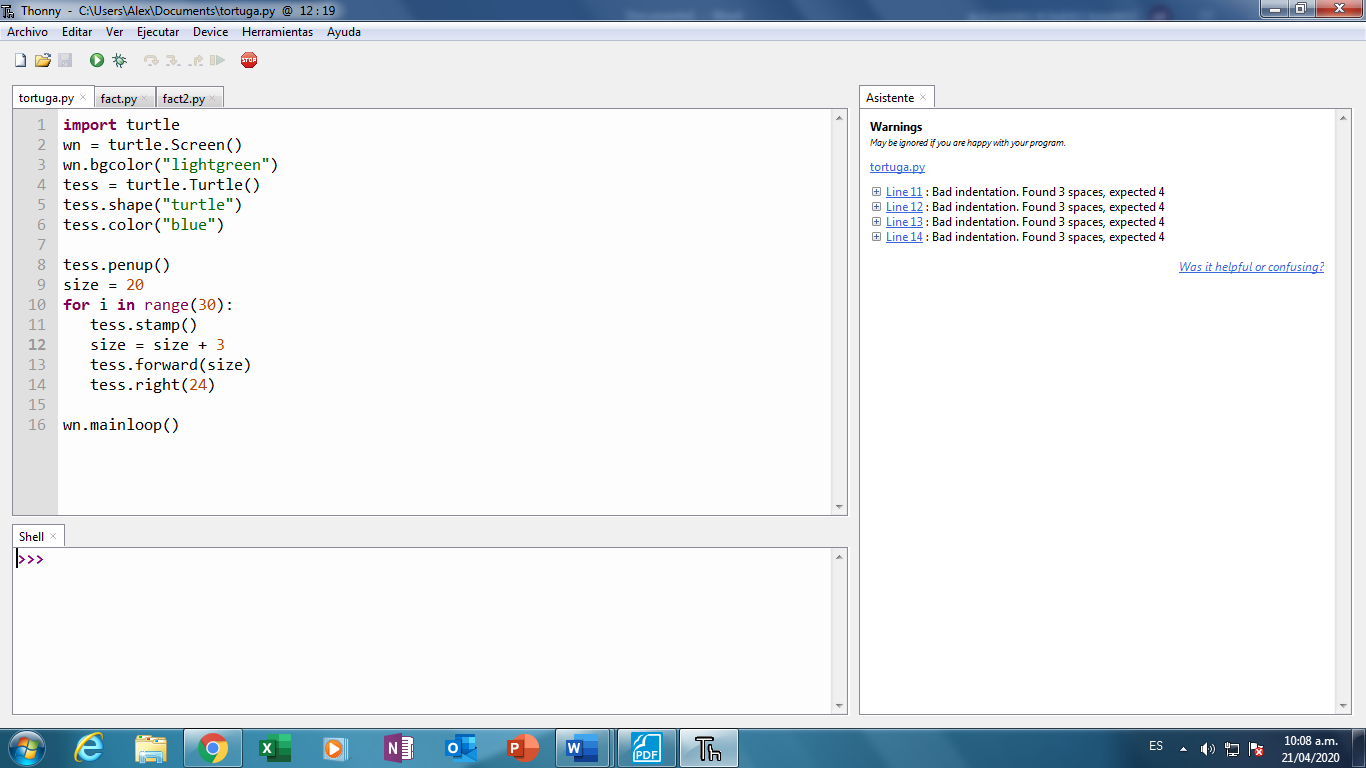


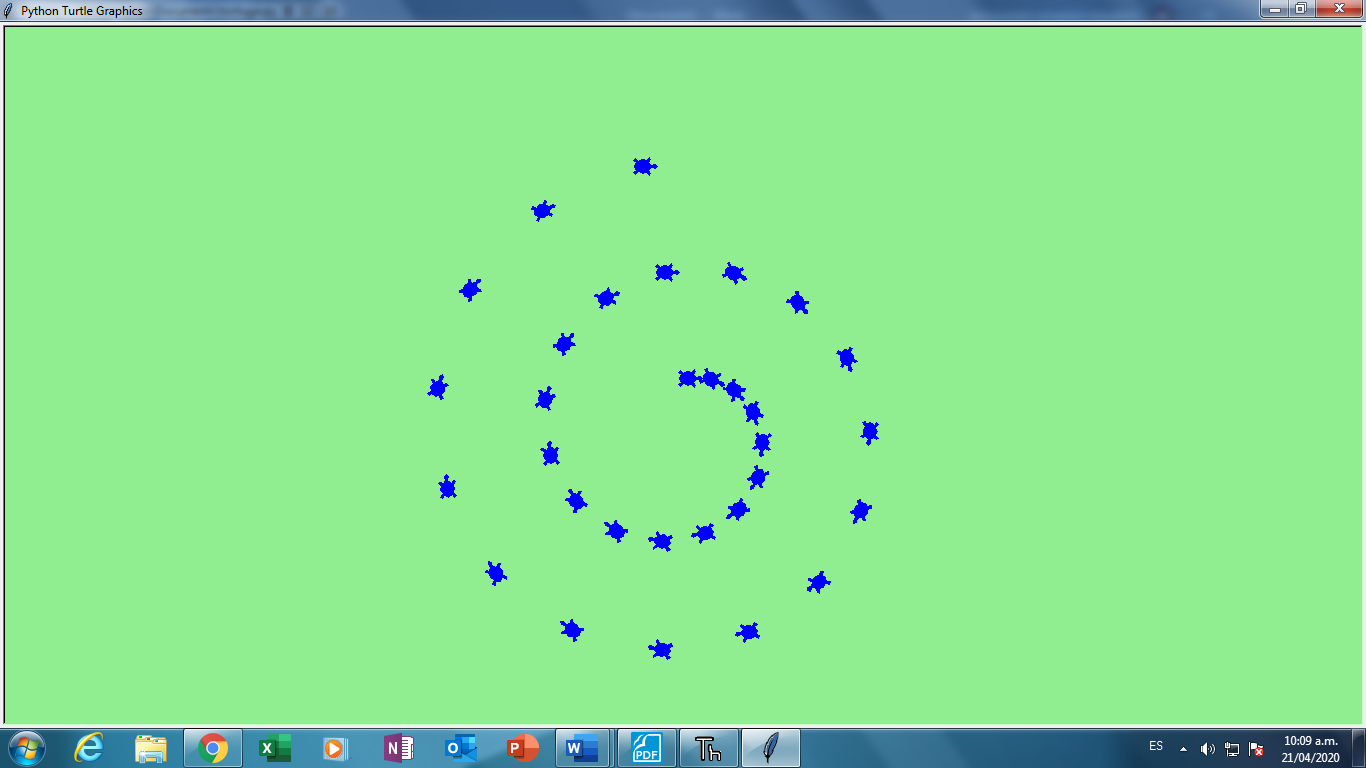
O bien, con recursividad llamando a la función a sí misma y haciendo el problema cada vez más pequeño de la siguiente manera:



El siguiente ejercicio muestra el desarrollo de las llamadas “Huellas de Tortuga” donde la tortuga se mueve a través de una función recursiva, es decir, que se llama a sí misma. Como se observa, la función recursiva indica las vueltas que hace la tortuga durante su recorrido.

El código y la ejecución se muestran aquí:





**CONCLUSIONES**

* La recursividad descompone, a medida que va ejecutándose, la resolución del problema en partes más pequeñas; sin embargo, las veces en que se puede ejecutar la recursión son limitadas, pues consume espacio de memoria y tiempo de ejecución en un ordenador.
* La ejecución mediante recursión implica cálculos lógicos en el algoritmo.
* Al generar recursión demasiadas veces, pueden producirse errores de compilación.

**BIBLIOGRAFÍA**

Gimeno, J. M., & González, J. L. (s.f.). *Recursividad*. Obtenido de Universitat de Lleida: http://ocw.udl.cat/enginyeria-i-arquitectura/programacio-2/continguts-1/2-recursividad.pdf

*Recursividad. Análisis de Algoritmos*. (Abril de 2011). Obtenido de Universidad Veracruzana: https://www.uv.mx/personal/ocastillo/files/2011/04/Recursividad.pdf