



Universidad Simón Bolívar

Departamento de Computación y Tecnología de la Información

CI3641 – Lenguajes de Programación 1

Trimestre: Septiembre - Diciembre 2023

Profesor: Ricardo Monascal

Estudiante: Junior Miguel Lara Torres, Carnet: 17-10303

Parcial 2 (30 pts)

- “En algunas preguntas, se usarán las constantes X , Y y Z . Estas constantes debe obtenerlas de los últimos tres números de su carnet.”

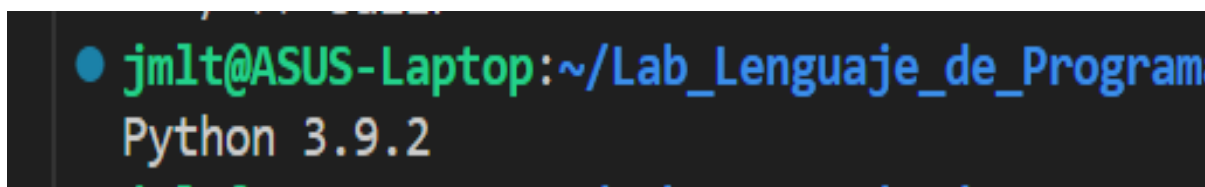
Caso particular, 17-10303 entonces $X = 3$, $Y = 0$, $Z = 3$.

- “En aquellas preguntas donde se le pida implementar un programa, mantenga su código en un repositorio git remoto (preferiblemente Github) y coloque un enlace al mismo en lugar de su respuesta. Todo su código debe ser legible y estar debidamente documentado.”

Todos los códigos, este propio documento, documento oficial del enunciado de examen 2 y otras evaluaciones de la materia serán cargados en:

https://github.com/JMLTUnderCode/Programming/tree/main/USB_Language_Programation

En todas las respuestas dadas en este parcial se usó Python cuya versión es 3.9.2.



- **1er Pregunta:**

Sin tiempo. 😞

- **2er Pregunta:**

Se escoge Python3 como lenguaje de programación.

En el siguiente link encontrara los siguientes archivos esenciales:

- ➔ Pregunta_2_respuesta.py
- ➔ AritmeticExprCalculator.py
- ➔ test_AritmeticExprCalculator.py

https://github.com/JMLTUnderCode/Programming/tree/main/USB_Language_Programation/Parcial_2/Source_Pregunta2

La ejecución del archivo fuente principal se realiza por consola con:

```
[py o python3] Pregunta_2_respuesta.py
```

En el caso de la cobertura se requiere que haga la instalación **pytest** y **coverage** en su sistema operativo basado en Linux. Es decir, realice los siguientes pasos:

```
pip install pytest  
pip install coverage
```

Luego debe realizar la siguiente corrida en consola:

```
coverage run -m pytest test_AritmeticExprCalculator.py
```

Mostrándole en pantalla la ejecución correcta de las pruebas unitarias. Luego para ver la información detallada de la cobertura debe escribir en consola:

```
coverage report -m
```

Cuyos resultados corresponden a un 100% de cobertura.

- **3er Pregunta:**

Sin tiempo. 😞

- **4er Pregunta:**

NOTA IMPORTANTE: Para la versión recursiva de cola, se tuvo muchos problemas de implementación, realmente la parte más difícil hasta el momento del parcial para mi. En este sentido, quiero comentarle de ante mano que el código escrito para la versión recursiva de cola esta discutida en equipo con Astrid Alvarado(18-10938) de la cual llegamos a muchas formulas, muchas conclusiones, todas erradas claro, pero al final logramos de cierta forma hacer una “recursión de cola” razonable en tiempo, quizás no en implementación, pues se ve engorroso.

Para esta pregunta tenemos que verificar que miembro de la familia de funciones nos tocó, como $X = 3$, $Y = 0$, $Z = 3$, entonces tenemos la siguiente función:

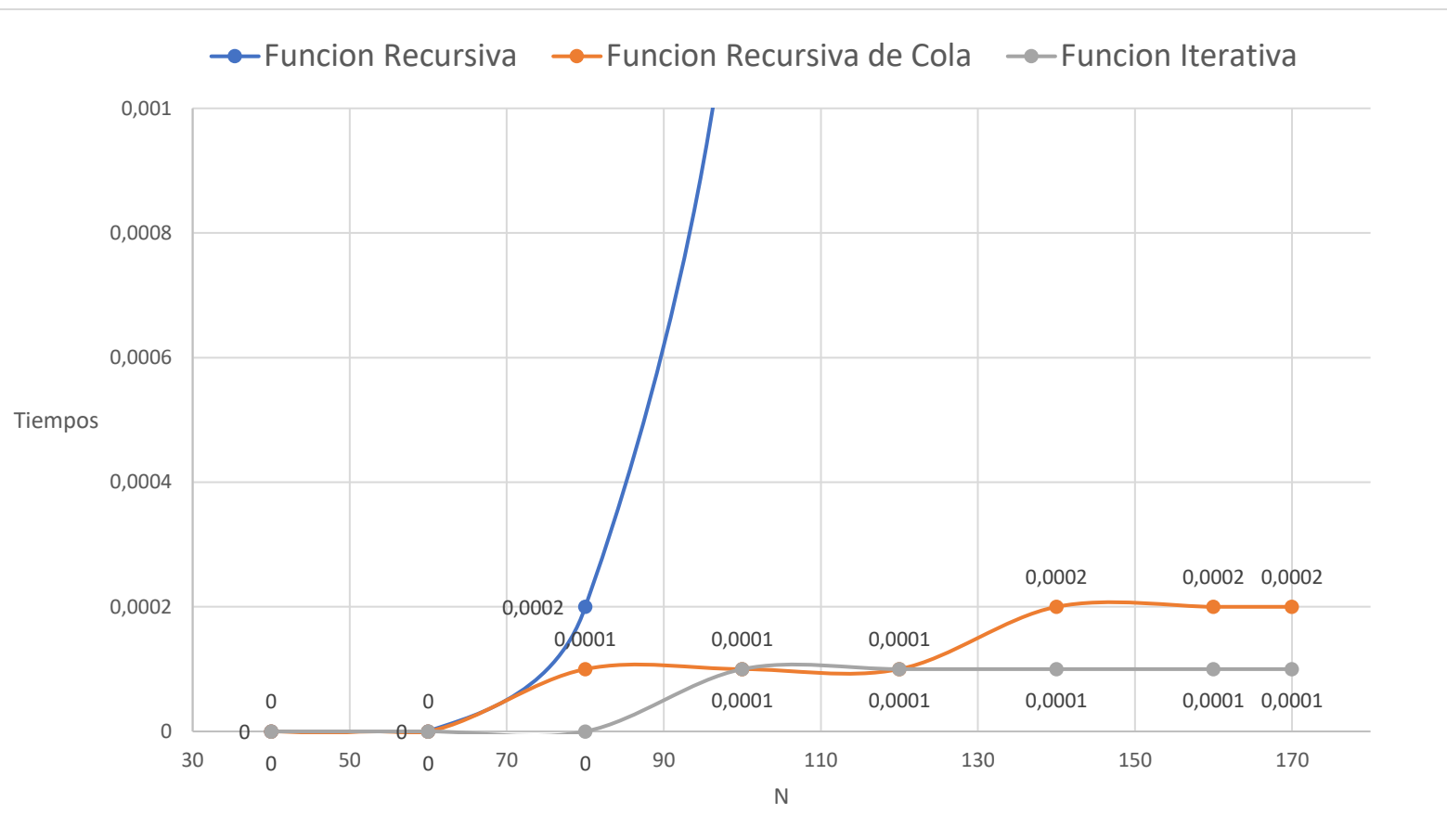
$$\alpha = 6 \wedge \beta = 6$$

$$F_{6,6}(n) = \begin{cases} n & \text{si } 0 \leq n < 36 \\ F_{6,6}(n-6) + F_{6,6}(n-12) + F_{6,6}(n-18) + F_{6,6}(n-24) + F_{6,6}(n-30) + F_{6,6}(n-36) & \text{si } n \geq 36 \end{cases}$$

El archivo fuente que realiza los cálculos para la versión recursiva, recursiva de cola e iterativa se encuentran en el link a GitHub siguiente, así como también el archivo .csv de los tiempos de ejecución para múltiples valores.

https://github.com/JMLTUnderCode/Programming/tree/main/USB_Language_Programation/Parcial_2/Source_Pregunta4

La grafica se la presento a continuación, de la cual se esperar, que la versión iterativa es más eficiente en todo caso que la propia versión recursiva de cola, sin embargo, la idea de una buena implementación de recursividad por cola debe ser tal que sea similar a una versión iterativa. Claro que se complicó como se mencionó anteriormente a nivel de implementación.



- **5er Pregunta:**

El lenguaje escogido fue Python. Se arregla el link a los archivos esenciales en GitHub:

https://github.com/JMLTUnderCode/Programming/tree/main/USB_Language_Programation/Parcial_2/Source_Pregunta5

PD: No se imprimen los resultados para la versión optimizada de DESCRIBIR, pues no está implementada.

- **Pregunta Extra:**

Sin tiempo. 😞