1. **IDENTIFICACIÓN:**

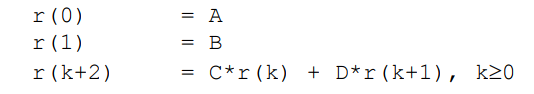
Camilo Anzola (201529838)

Santiago Rodríguez González (201512341)

1. **ALGORITMO DE SOLUCIÓN:**

Básicamente lo que hace el programa es, luego de recopilar los datos de entrada (n, A, B, C y D), poner a trabajar a un invariante que llena un arreglo con todos los resultados a la recursión y finalmente consulta de ese arreglo los resultados que necesita para calcular la solución. Esto se logra de la siguiente manera:

El método llenarInvariante() llena un arreglo (utilizando un for desde cero hasta n) con datos equivalentes a la recursión dada por la función r del enunciado



En este punto se utilizó la técnica de programación dinámica, la cual hace que el algoritmo sea más rápido pues los cálculos de la función r no se hacen de manera recursiva, sino que se van guardando en un arreglo de tamaño n utilizando los datos de las posiciones anteriores y los casos base.

La siguiente función describe la manera en que se llena el arreglo:

Después, teniendo este arreglo ya completo, el programa procede a calcular la convolución ponderada utilizando un ciclo que aplica la fórmula del enunciado con la salvedad de que la función r lo que hace es retornar el valor del arreglo en la posición n en vez de hacer la recursión.

Finalmente, el programa redondea la respuesta de acuerdo a las especificaciones del enunciado utilizando un algoritmo recursivo. Este algoritmo empieza desde el dígito número cuatro después del punto decimal (en caso de que exista, y si no existe completa con ceros hasta llegar a la posición cuatro) y se llama recursivamente hasta que llegue al penúltimo dígito y suma una unidad a ese dígito en caso de que el siguiente dígito sea mayor que cinco. De esta manera se va devolviendo en la recursión modificando los dígitos si es necesario. La respuesta es entonces el resultado redondeado.

1. **ANÁLISIS DE COMPLEJIDADES:**

COMPLEJIDAD ESPACIAL:

El programa utiliza un arreglo de tamaño n y unas cuantas variables necesarias para algunos cálculos. Estas variables ocupan memoria, pero el procedimiento esencial del programa consiste en llenar el arreglo de tamaño n para luego consultar de él los elementos necesarios para el cálculo de la respuesta.

COMPLEJIDAD TEMPORAL:

El programa ejecuta varios ciclos (no anidados) de n iteraciones y aunque tiene también una función recursiva para redondear el resultado, la complejidad de esta es menor a la complejidad de llenar el arreglo ya que el método que redondea solo efectúa la recursión si hay menos de cinco dígitos por redondear, por lo tanto, en realidad esta recursión tiene una complejidad muy baja y por lo tanto despreciable en comparación con la complejidad de llenar el arreglo.

1. **COMENTARIOS FINALES:**

Al correr el programa, se puede observar un desempeño bastante bueno ya que calcula la respuesta de manera rápida, incluso probando con casos en los que n es grande (n = 100). Algo interesante a resaltar es que la primera instancia del programa no usaba programación dinámica y por lo tanto, al probar con casos en los que n era mayor a 10000 ocurría un StackOverflow, pero con la técnica de programación dinámica calcula la respuesta, aunque se demora un poco.