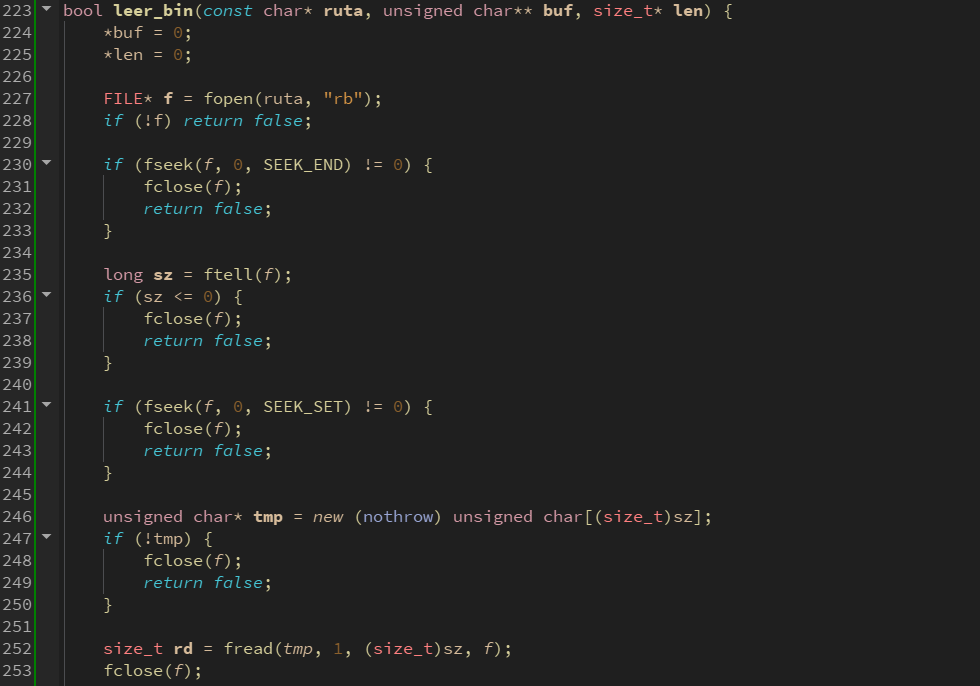
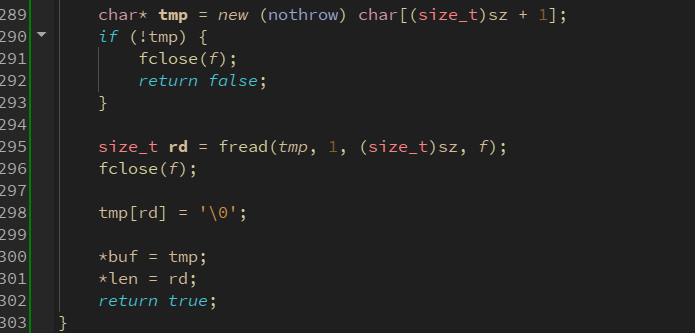
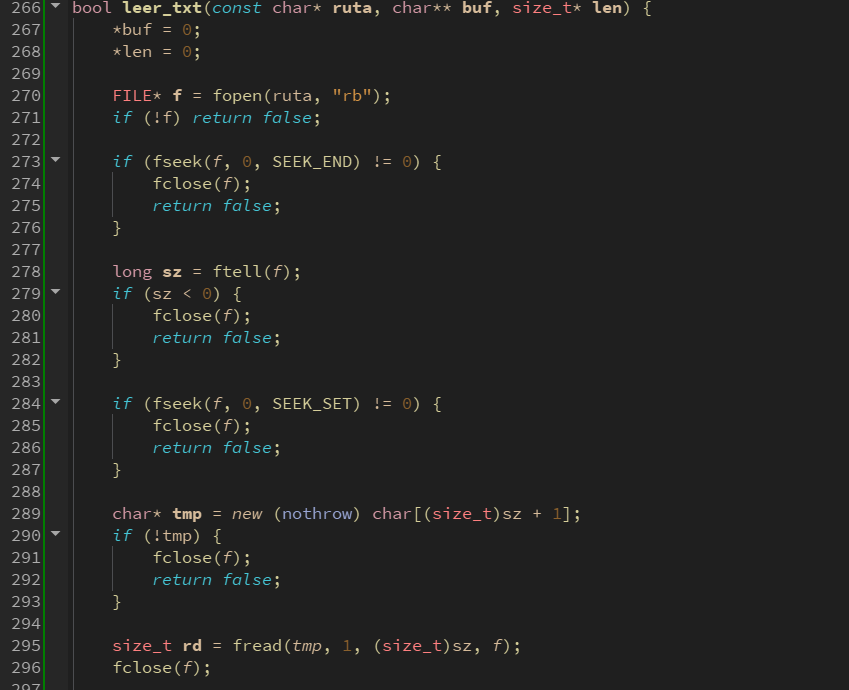
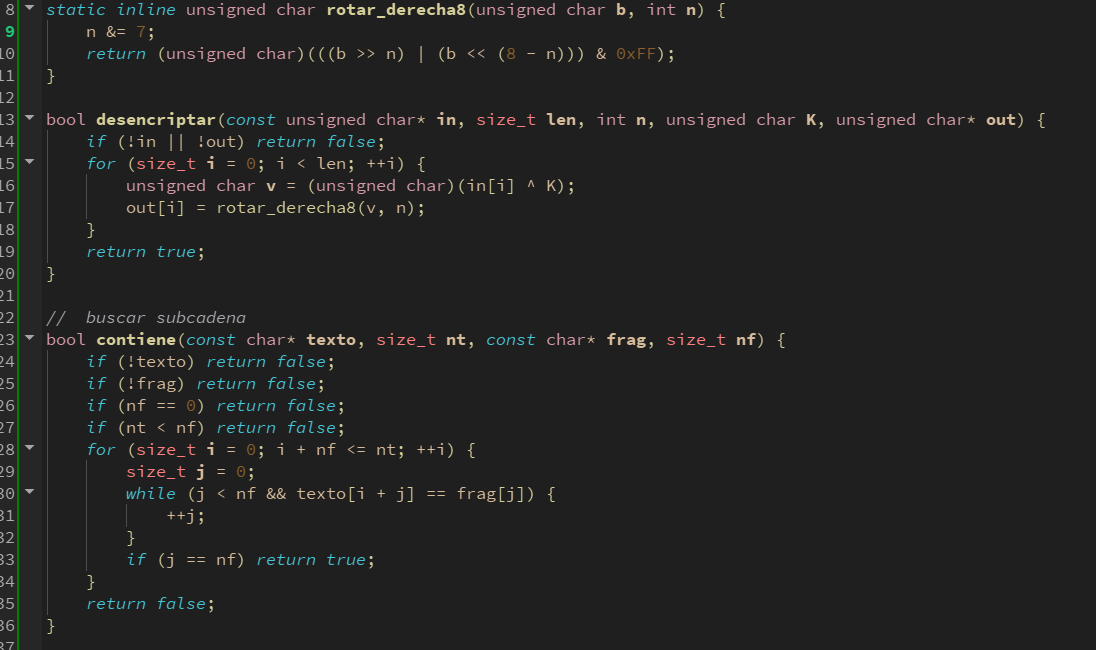
INFORME FINAL

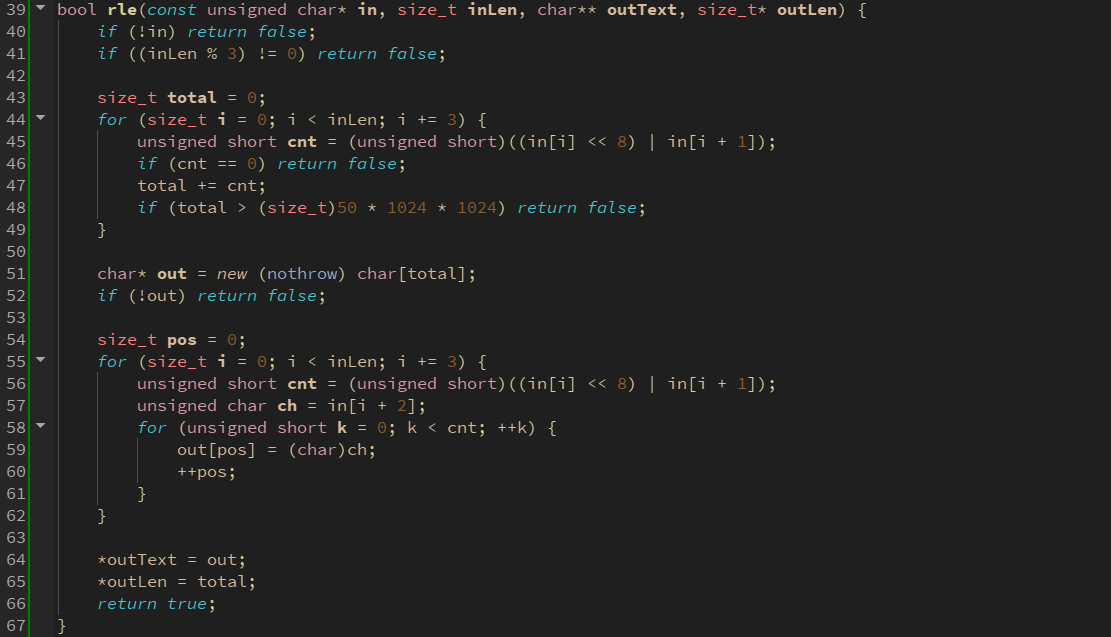
Como ya muy bien sabemos el problema que resolvemos consiste en que se nos entrega un texto el cual ha sido sometido a un proceso de compresión (RLE o LZ78) y posteriormente una encriptación que consiste en rotar los bits n veces a la izquierda y a cada bit se le aplican operaciones lógicas XOR con cierta clave K. Se nos pide generar un código de ingeniería inversa donde logremos entregar el texto original indicando la clave XOR, la cantidad n de rotación de bits y la compresión utilizada.  
como ya indique en anteriores informes lo que vamos a utilizar es un código de fuerza bruta que evalué todas las posibilidades hasta encontrar una que coincida con la pista, en otras palabras, seria como nadar en contra de la corriente en todas las direcciones posibles hasta encontrar el camino correcto, por lo tanto, tenemos que empezar desde el final hasta el principio del problema, ósea, primero desencriptar y luego descomprimir.

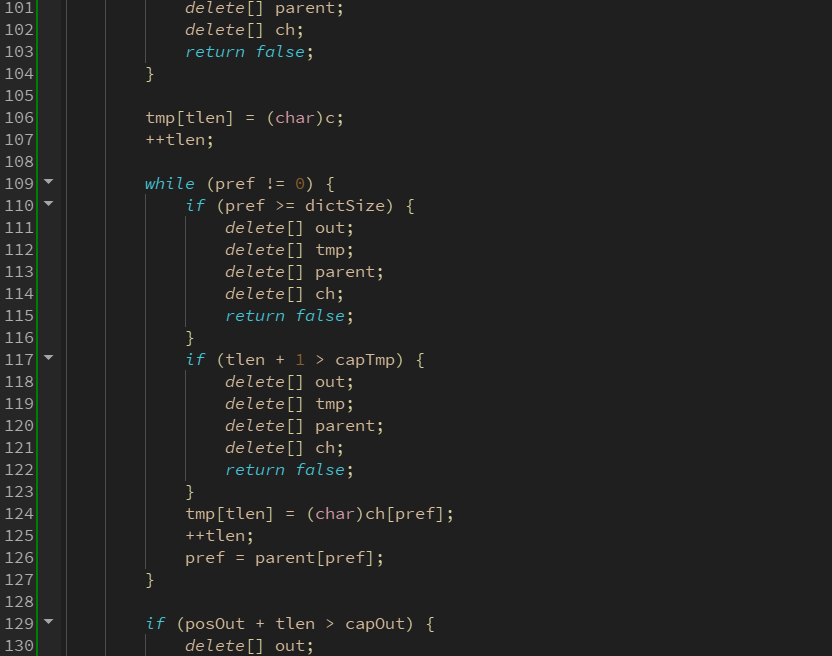
Para el desarrollo del código primero necesitamos ciertas utilidades para poder leer el archivo Encriptado.txt:  


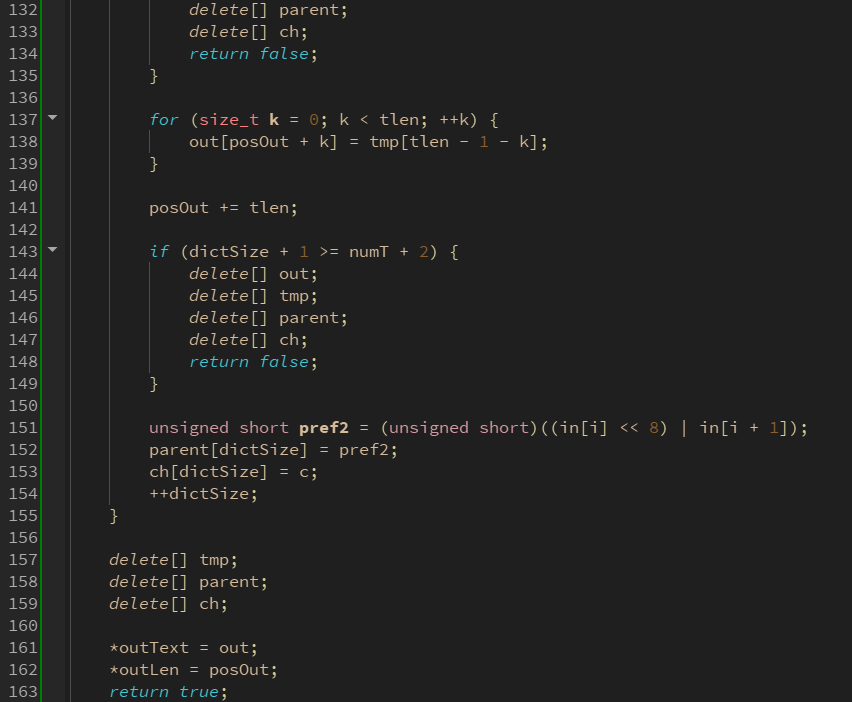
También cree otra función para leer Pista.txt:  


Ahora las funciones de desencriptado y búsqueda de la cadena pista en el texto que hallemos:  


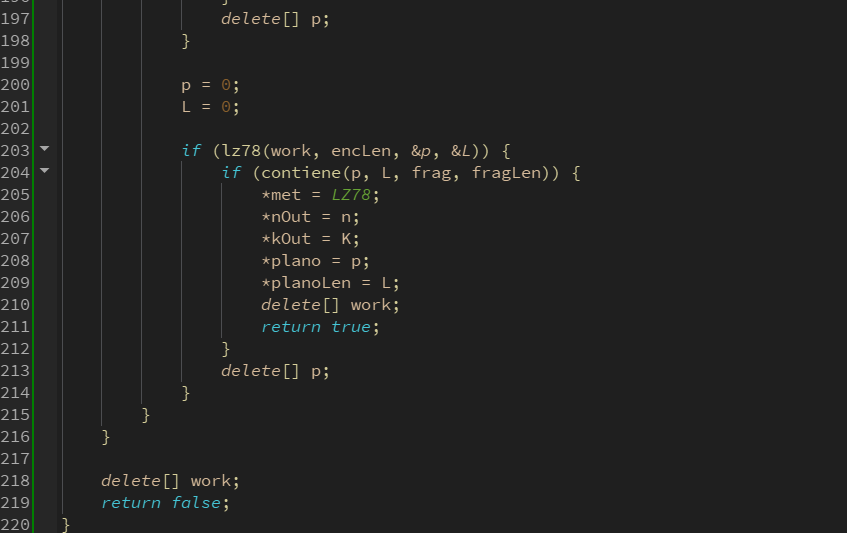
Función de compresión RLE:



Función de compresión LZ78: 

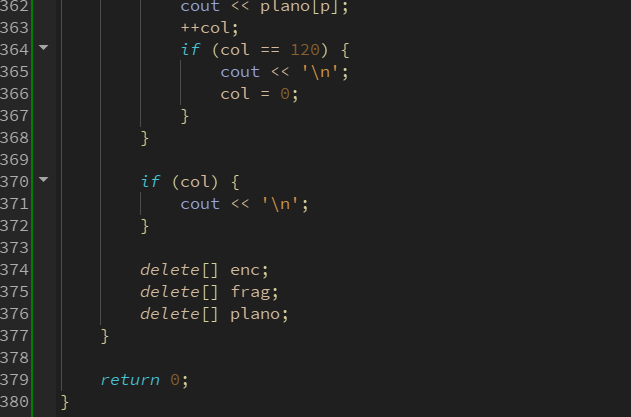
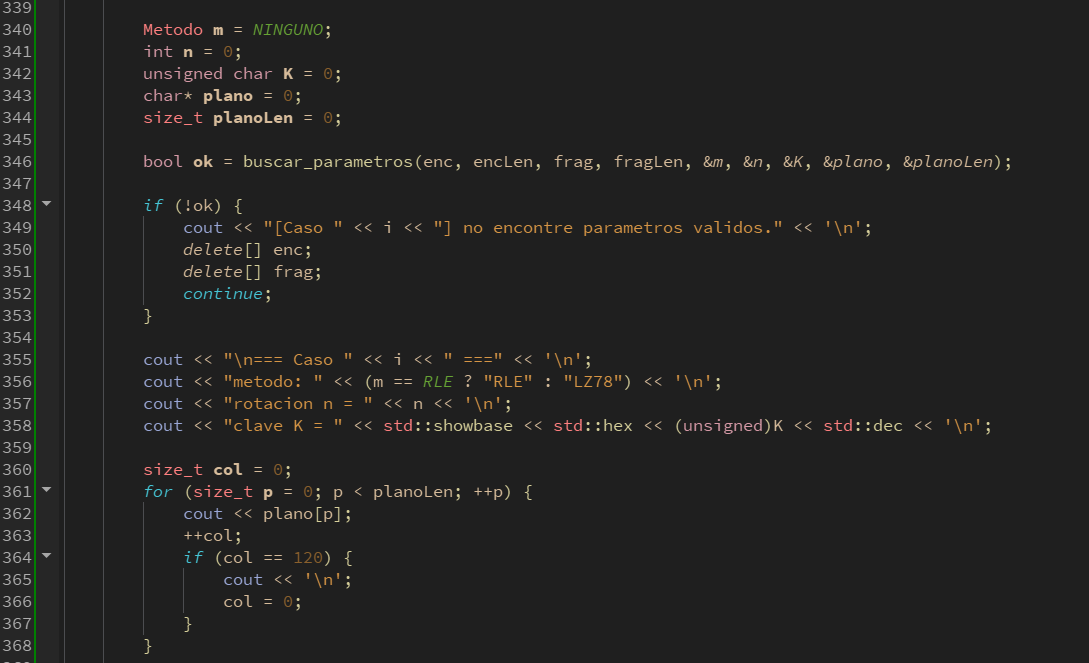


La última función es de fuerza bruta para n y para k, donde encontramos los parámetros que estamos buscando:  

Finalmente tenemos el main:





PROBLEMAS DE DESARROLLO AFRONTADOS

El mayor problema que presente a la hora de desarrollar el código fue el error en el planteamiento de estrategia que expuse en el informe preliminar, lo cual me quito mucho tiempo, sin embargo, fue una etapa necesaria para el éxito en el desarrollo del código. Ahora en la implementación de la estrategia “correcta” una de las cosas más complicadas de la implementación de la compresión LZ78 debido a su complejidad y otra cosa que se me dificulto fue la lectura de los archivos y el uso de la memoria dinámica.

EVOLUCION DE LA SOLUCION Y CONSIDERACIONES PARA TENER EN CUENTA EN LA IMPLEMENTACION.

Como ya lo hemos hablado en clase mas de la mitad del trabajo consiste en el análisis y planteamiento de la estrategia, por lo tanto, por mas que sepamos mucho de memoria dinámica, punteros, operadores de bits, lo que debemos saber es implementarlo al problema que queremos resolver. Una vez se ha encontrado una buena estrategia debemos analizar si es viable y empezar a programar lo antes posible.

En lo personal algo que aprendí en este trabajo es no aferrarse a una estrategia si desde un principio no funciono, por mas que queramos convencernos a nosotros mismos de que podemos hacer que funcione hay veces que simplemente toca aceptar que nos equivocamos, replantearse y si es necesario empezar desde cero.