**Alex Aiverson Palacios Mosquera - 1010108842**

**a. Análisis del problema**

Recibimos un mensaje que primero lo comprimieron (usando RLE o LZ78) y luego lo encriptaron (rotación de bits + XOR con una clave).

El problema es que no me dicen qué método usaron ni con qué parámetros, solo me dieron el mensaje ya enredado y un pedacito del mensaje original, entonces como no sé si usaron RLE o LZ78, tengo que probar los dos.

En la parte de encriptación, solo hay 7 rotaciones posibles (de 1 a 7 bits) y 256 claves (de 0 a 255). Eso significa que puedo probar todas las combinaciones hasta que una me dé el fragmento conocido.

Cuando encuentre los parámetros correctos, ya sé qué método usaron y con qué valores. Luego solo queda desencriptar y descomprimir para sacar el mensaje completo.

**b. Tareas que definí**

Dividí el trabajo en partes:

Encriptación/Desencriptación:

Hice una función para rotar bits a izquierda y a derecha.

Una función para hacer XOR con una clave.

Y otra que junta esas dos para desencriptar.

RLE:

Función para comprimir: cuenta letras repetidas y guarda “número + letra”.

Función para descomprimir: lee el número y repite la letra esa cantidad de veces.

LZ78:

Función para comprimir: va guardando subcadenas en un diccionario y genera pares (número, letra).

Función para descomprimir: reconstruye el texto usando el diccionario.

Módulo “adivino”:

Probar todas las combinaciones de n y K.

Revisar si en el resultado aparece el pedacito conocido.

Programa principal:

Unir todo: detectar método y parámetros → desencriptar → descomprimir → mostrar el mensaje original.

**c. Algoritmos que implementé**

Encriptación/Desencriptación:

Para rotar bits usé operaciones de corrimiento y máscaras.

Para XOR solo apliqué el operador con la clave.

Para desencriptar, primero hago XOR y después la rotación inversa (a la derecha).

RLE:

Ejemplo: “AAAAB” → “4A1B”.

Al descomprimir: “4A1B” → “AAAAB”.

LZ78:

Ejemplo: “ABAB” → (0,A), (0,B), (1,B).

Para descomprimir, voy reconstruyendo el diccionario con esos pares hasta armar el texto.

d. Problemas que me tocó enfrentar

Memoria dinámica, coincidencias que no eran, manejar diccionarios con arreglos dinámicos sin pasarme de tamaño

**e. Cómo fue evolucionando mi código**

Al principio: probé cada función por separado con ejemplos sencillos, como “AAAAB” en RLE o rotar un solo byte, después junté todo en un programa y ahí salieron muchos errores, revisé línea por línea, arreglé fugas de memoria y errores de lógica, liberar memoria apenas ya no se use.