Nicolás Lonodoño- 201821364 – n.londonoc

Camila Parra – 201819464- mc.parrad

María Gabriela Torres. 201820251. mg.torres

**Particularidades del proyecto**

**Longitud en bits del divisor que ingresa el usuario (método calcularCRC):**

Al ingresar los argumentos para correr el programa el divisor que se ingresa debe tener la longitud de un byte, es decir 8 bits. Para que esto ocurra se debe completar el divisor con ceros a la izquierda si la cantidad de caracteres de este es menor a 8 para que funcione. Esto se puede atribuir a que asumimos que el dato que entra por parámetro siempre llega como un char que representa un byte, que tiene 8 como longitud. Teniendo en cuenta lo anterior se puede afirmar que si entra un dato con menos de ocho caracteres no va a ser reconocido por el programa. Sin embargo, cuando el dato que entra tiene 8 bits el programa se ejecuta, lo pasa a binario y calcula el crc con la longitud del divisor real y queda el divisor que debe ser por lo que termina la ejecución.

**Resultado que se imprime en consola:**

Otra particularidad que tiene el programa es que el resultado que se imprime no concuerda con el que se debería mostrar en la consola. En el caso del caso de prueba 1, el resultado debería ser 78 en hexadecimal. El número que se imprime en pantalla es extremadamente largo y tiene muchos caracteres. Se debería mostrar una cadena de ceros seguidos por el 78 sin embargo, esto no ocurre. Una de las posibles causas de que el resultado no se muestre correctamente es la función strrev(), que se encarga de invertir el resultado binario del dato ingresado. Esta función se utilizó para corregir el resultado de pasar el valor del divisor a binario para poder hacer la división del dato de interés y posteriormente calcular el CRC. Otra posible causa es que al hacer la división para calcular el crc haya un error en los datos, ya sea en el byte calculado en la función calcularByte o en el polinimio generador.