PROGRAMACIÓN AVANZADA
EXAMEN FINAL

CAMPUS QUERÉTARO

Apegándome al Código de Ética de los Estudiantes del Tecnológico de Monterrey, me comprometo a que mi actuación en esta actividad de evaluación esté regida por la honestidad académica. En congruencia con el mismo, realizaré esta actividad de forma honesta y personal, para reflejar a través de ella mis conocimientos y competencias.

Nombre: _____ Matrícula: _____

noviembre, 2020

Criptografía (del griego *krypto*, «oculto», y *graphos*, «escribir», literalmente «escritura oculta») tradicionalmente se ha definido como la parte de la criptología que se ocupa de las técnicas, bien sea aplicadas al arte o la ciencia, que alteran las representaciones lingüísticas de mensajes, mediante técnicas de cifrado y/o codificado, para hacerlos ininteligibles a intrusos (lectores no autorizados) que intercepten esos mensajes. Por tanto el único objetivo de la criptografía era conseguir la confidencialidad de los mensajes. Para ello se diseñaban sistemas de cifrado y códigos. En esos tiempos la única criptografía que había era la llamada criptografía clásica.
Un esquema de cifrado digital muy sencillo de es cambiar la posición que los bits tienen en un byte. Por ejemplo, tenemos un byte que tiene la secuencia de bits:

```
01011111
```

Y decimos intercambiar la mitad inferior por la superior, el resultado sería:

```
11110101
```

Escribir un programa llamado *decripta.c* que se encarga de decriptar un archivo que fue encriptado usando la técnica explicada arriba. La forma de invocar el programa es:

```
$decrypt origin destination
```

El programa recibe como parámetros el nombre del archivo origen (archivo criptado) y el nombre del archivo destino (archivo decriptado). Ejemplos de uso:

```
./decrypt
usage: ./decrypt origen destination

----------------------------------------
./decrypt no_exist out.txt
./decrypt: No such file or directory

----------------------------------------
./decrypt cripto.txt out.txt
done
----------------------------------------
```

Rúbrica de evaluación:

Ponderación	
10 puntos	Verifica que el programa la cantidad correcta de parámetros (nombre de los archivo).
10 puntos	Verifica que el archivo `origin` existe (archivo cifrado).
10 puntos	Verifica que el archivo `destination` se pueda crear (archivo descifrado).
70 puntos	Descripta correctamente el archivo.

TIPS:

1. Usa “unsigned char”.
2. Operaciones sobre bits: `https://www.cprogramming.com/tutorial/bitwise_operators.html`