CENTRO SUPERIOR DE INFORMÁTICA
Departamento de Estadística, I.O. y Computación
Teoría de Autómatas y Lenguajes Formales
Curso 2002-2003

PRACTICA 4: Manipulación de ficheros de texto: Criptografía

Durante la segunda guerra mundial, el ejército alemán utilizó la máquina conocida como *Enigma* para codificar sus mensajes. Básicamente dada una *semilla* la máquina *Enigma* generaba una secuencia de números pseudoaleatorios que era difícil de reproducir, incluso aunque los detalles técnicos de la máquina pudieran ser descubiertos.

Los aliados habían capturado algunas de las máquinas *Enigma*, de forma que conocían la forma en que la máquina trabajaba, pero los trabajos que se realizaron para descubrir los códigos de la *Enigma* fueron los fundamentos de la informática moderna. El propio Alan Turing participó en este tipo de trabajos. Si está interesado en conocer más sobre esta historia, puede consultar la siguiente referencia:

`http://www.gl.umbc.edu/Ĩmazia1/Enigma/enigma.html`

La criptología es la rama de conocimiento que se ocupa del estudio y diseño de sistemas que permitan comunicaciones secretas entre un emisor de un mensaje y uno o varios receptores del mismo. Inicialmente las únicas aplicaciones de la criptología fueron militares, pero hoy en día están surgiendo otras aplicaciones. Por ejemplo, en los computadores multiusuarios, cada usuario prefiere mantener sus ficheros de una forma que no sean legibles para otros usuarios "indiscretos"; para conseguir esto, los ficheros se codifican (encriptan) utilizando una clave que sólo conoce su propietario. Alguna de la información que enviamos a través de internet viaja también de forma codificada para protegerla de receptores no deseados.

El objetivo de esta práctica es el diseño de un programa que se llamará cripto.c, cuya finalidad será encriptar y/o desencriptar ficheros de texto.

Si el programa se ejecuta sin pasar parámetros en la línea de comandos, debemos obtener el siguiente mensaje:

```
cripto -- file encriptor
Usage: cripto input_file output_file method operation
```

```
input_file:  the file to be processed
output_file: the resulting file
Method: the encription/decription method:
                1: Xor Method
                2: Cesar Method
operation:   operation to be carried on the file:
                +: encrypt the file
                −: decrypt the file
```

Que es una explicación al usuario del modo de funcionamiento del programa: los parámetros son los nombres de un fichero de entrada, el de salida y una operación que será un signo + o bien -. El fichero de entrada se encriptará o desencriptará (según la operación introducida (+ o -)).

Ante cualquier anomalía en los parámetros introducidos en la línea de comandos, el programa deberá presentar siempre este mismo mensaje: por ejemplo si hay 4 parámetros en lugar de 3 o si la operación especificada no es + ni -. Se indicará también un mensaje de error si el programa no consigue abrir el fichero de entrada.

Para encriptar un fichero hay muchas alternativas. Todas ellas consisten en transformar cada uno de los caracteres del fichero original c_i en otro carácter c'_i siguiendo una determinada transformación. Indicaremos dos métodos diferentes de encriptado. En el programa se implementarán al menos estos dos métodos.

4.1 Método1: Xor

Se solicitará al usuario una clave secreta de encriptado, que habrá que conocer también si se desea decodificar el fichero.

A cada uno de los caracteres del fichero se le hará una transformación, que consistirá en hacerle la operación `xor` con un carácter de la clave secreta. El carácter de la clave secreta con el que se transforma el carácter original, se variará de forma cíclica. P. ej. si suponemos que la clave secreta es la palabra *alfa*, y las primeras palabras del texto son *Teoría de autom...*, los primeros caracteres del fichero de salida serán:

```
carácter 1  T xor a
carácter 2  e xor l
carácter 3  o xor f
carácter 4  r xor a
carácter 5  í xor a
carácter 6  a xor l
```

```
carácter 7    xor f
carácter 8  d xor a
carácter 9  e xor a
carácter 9    xor l
carácter 10 a xor f
carácter 11 u xor a
carácter 12 t xor a
carácter 13 o xor l
carácter 14 m xor f
. . .
```

Es decir, se va haciendo la operación `xor` de cada uno de los caracteres del texto de entrada con cada uno de los caracteres de la clave secreta, tomando la clave de forma cíclica (cuando se acaba con el último carácter de la clave, se comienza de nuevo con el primero).

Antes de operar de este modo se procesará la clave secreta haciendo `xor` a cada uno de sus caracteres con el número 128.

Una ventaja de este método es su especial aptitud para ser utilizado en un ordenador (puesto que la operación `xor` se realiza muy eficientemente en un ordenador).

Otra ventaja del método es que la operación de desencriptado consiste en hacer exactamente lo mismo al texto que se ha encriptado (con la misma clave secreta, por supuesto).

4.2 Método 2: Cifrado de César

Como se deduce de su nombre, este método era usado ya ¡en tiempos de los romanos!. En este caso, la codificación es como sigue: si una letra en el texto a codificar es la N-ésima letra del alfabeto, sustitúyase esa letra por la (N + K)-ésima letra del alfabeto. (César utilizaba el valor K = 3). Se muestra a continuación un texto encriptado siguiendo este método y utilizando K = 1:

```
Texto original: Navidad, Navidad, dulce navidad
Texto encriptado: Obwjebe-!Obwjebe-!evmdf!obwjebe
```

Podemos optar por hacer fijo el valor de K o bien solicitarlo al usuario.

Evidentemente, el desencriptado del fichero consistirá en realizar la operación inversa, y en este caso, el valor de K a utilizar debería solicitarse al usuario para garantizar que está autorizado a leer el fichero.

El funcionamiento del programa que se pide, debería ser análogo al de los programas

`ftp://ftp.csi.ull.es/pub/asignas/AUTOMALF/p01_cripto/cripto`
`ftp://ftp.csi.ull.es/pub/asignas/AUTOMALF/p01_cripto/cripto.exe`

Posibles mejoras del programa:

1. Para estudiar el comportamiento del método Xor, diseñar una función que imprima en un fichero un carácter, su código ASCII y su representación binaria. La función se utilizará para imprimir toda la tabla ASCII, estudiando las configuraciones binarias correspondientes a cada carácter.

2. El método de César se puede hacer más potente utilizando una tabla en la que para cada posible carácter en el texto fuente, se indica cuál es el carácter transformado. Evidentemente, esta tabla deberá ser conocida por quien desee decodificar el fichero. Implementar esta técnica.

3. Buscar información sobre otros esquemas de encriptado, e implementarlos.

4. Buscar la forma de conocer el tamaño del fichero que se desea procesar, alojar una zona de memoria suficientemente grande como para almacenar allí todo el fichero, leerlo completamente y almacenarlo en la memoria, y procesarlo desde memoria en lugar de leerlo caracter a caracter.

L@s alumnos deberán realizar una implementación orientada a objetos, tratando de seguir los pasos que ya expusimos en una práctica anterior:

1. En el ámbito de aplicación de su programa, identifique las entidades (objetos) y sus atributos (datos).

2. Determine las acciones que pueden realizarse sobre un objeto.

3. Determine las acciones que un objeto puede realizar sobre otros objetos.

4. Determine las partes de cada objeto que serán visibles a otros objetos, qué partes serán públicas y cuáles privadas.

5. Defina la interface pública de cada objeto.