Examen de Programación – Curso 2012/13 – 3er Trimestre

1. Escribe un programa *GraficoAsteriscos* que lee enteros de un fichero de texto (yo os proporciono el fichero) y los clasifica en 10 segmentos distintos. En un *array* (o una lista o lo que queráis) iremos contando cuántos de esos números se encuentran entre el 0 y el 9, entre 10-19, entre 20-29, etc., hasta llegar al último que será desde el 90 al 99.

   Con los datos de ejemplo, el fichero tendría que sacar por pantalla el siguiente resultado:

```
00-09: **
10-19: ********
20-29: **********
30-39: **************
40-49: ******************
50-59: ******************
60-69: *********
70-79: **********
80-89: ***********
90-99: *
```

2. Crea la clase *Dado* que simula un dado de x caras. Se recomienda también hacer un pequeño programa donde se prueben los distintos componentes de la clase. La clase tendrá que tener los siguientes elementos:
   - Un constructor al que le pasamos un número entero. Ese número será el número de caras que tenga ése dado en concreto. Sólo se admitirán los valores: 2, 4, 6, 8, 10, 12, 20 y 100. Cualquier otro valor resultará en una excepción.
   - También tendremos un constructor vacío que nos creará automáticamente un dado de 10 caras.
   - Un método *Tirada* que nos devolverá un entero y que simulará una tirada de ese dado.
   - Un método *SumaTiradas* al que le pasamos un valor *n* y nos dirá el resultado de tirar *n* dados y sumar sus resultados.
   - Un método *TiradaMultiple* al que le pasamos un valor *n* y otro valor *tope*. Dentro de la función, simularemos la tirada de *n* dados. El método devolverá el número de aciertos, que serán las tiradas cuyo valor sea igual o superior al *tope*.
   - Una propiedad *NumeroCaras* de sólo lectura, que nos permitirá consultar las caras que tiene un dado (por si no nos acordamos o estamos usando dados de diferentes caras).
   - ***EXTRA:*** Un método *TiradaMultipleEspecial* al que le pasamos un valor *n* y otro valor *tope*. Será igual que el método anterior, pero por cada vez que saquemos un valor mínimo (el 1), restaremos un acierto del total y cada vez que saquemos un valor máximo (por ejemplo, un 10 en un dado de 10 caras), anotaremos un acierto y además volveremos a tirar ese dado (si en esa tirada sale un 1, no lo tendremos en cuenta, pero si sale un 10 volveremos a tirar una vez más).

3. Escribe un programa que te pide un valor numérico por teclado y te lo convierte a binario y a hexadecimal.

4. Escribe el programa *SuperCodificadorPlus*. El programa nos preguntará un par de nombres de fichero, leerá el contenido del fichero1, lo codificará y lo escribirá en el fichero2. El método de codificación será el siguiente: el fichero lo leeremos como un fichero binario (independientemente de si el fichero es de binario o de texto), e iremos leyendo los caracteres como valores numéricos de tipo *Byte*. Para codificar cada valor, si ese número es menos o igual que 127, le sumaremos 127, mientras que si es mayor o igual que 128, le restaremos 127. El valor codificado lo copiaremos en el segundo fichero.

   Os proporciono los ficheros "Quijote.txt" y "Quijote Codificado.txt" para que comprobéis si funciona correctamente.

   NOTA: Para poder sumarle cosas a un *byte*, hay que escribirlo de la siguiente forma:
   ```
   byte z = (byte)(x + y); // no preguntéis por qué
   ```

5. Escribe el programa *AutoBackup*. El programa hará lo siguiente: comprobará si en el directorio actual se encuentra el archivo "survival_1".
   - Si lo encuentra, hará una copia de seguridad de ese archivo en un archivo que se llamará "backup_0001". Si ya existe el archivo "backup_0001", la copia se hará en el archivo "backup_0002", y así sucesivamente.
   - Si no lo encuentra, buscará el archivo de *backup* más reciente (el que tiene el número más grande) y restaurará esa copia con el nombre del archivo original.
   - Si no se encuentra ni el archivo "survival_1" ni ningún "backup", el programa escribirá un mensaje de error (no hace falta lanzar una excepción).

   En cualquiera de los tres casos, el programa explicará si se ha encontrado o no el fichero y qué acción se ha realizado.

   NOTA: ¿No he explicado cómo se copian ficheros? Es con *File.Copy()*.

   NOTA2: Como alguno seguro que me lo pregunta: ¿qué pasaría si ya existe el fichero "backup_9999"? Pues que se borra y se machaca con la nueva copia de seguridad.

   NOTA3: Una mejor manera de comprobar cuántos ficheros de *backup* existen que comprobándolos uno a uno es usar la función *Directory.GetFiles()* que te devuelve un *array* de cadenas con los nombres de todos los ficheros que hay en el directorio actual. Si queréis la usáis, si no, no.