

Práctica 1. Diseño e implementación en código C/C++ de un reconocedor léxico de números decimales con signo y de números hexadecimales mediante el método de la tabla compacta. Ejemplos de números decimales que debe reconocer el programa: 25, -25, 25.50, -25.50, +25, +25.50, etc. Los números en formato hexadecimal seguirán la nomenclatura 0xNNNN de tal modo que se deben reconocer valores como: 0x000, 0xAA, 0x000aF, etc. Los datos de entrada reconocidos correctamente se deben almacenar en una tabla de símbolos para los números decimales (con un formato fijo formado por parte entera y parte decimal) y una tabla de símbolos para los números hexadecimales, que se puede implementar mediante una simple matriz o cualquier otra estructura que se considere apropiada.

Práctica 2. Diseño e implementación en código C/C++ de un reconocedor léxico de números decimales con signo y de números hexadecimales mediante el método del autómatas programado. Ejemplos de números decimales que debe reconocer el programa: 25, -25, 25.50, -25.50, +25, +25.50, etc. Los números en formato hexadecimal seguirán la nomenclatura 0xNNNN de tal modo que se deben reconocer valores como: 0x000, 0xAA, 0x000aF, etc. Los datos de entrada reconocidos correctamente se deben almacenar en una tabla de símbolos para los números decimales (con un formato fijo formado por parte entera y parte decimal) y una tabla de símbolos para los números hexadecimales, que se puede implementar mediante una simple matriz o cualquier otra estructura que se considere apropiada.

Práctica 3. Diseño e implementación mediante LEX/FLEX de un analizador estadístico de ficheros de código en lenguaje C, de forma que dado un fichero de entrada muestre el número total de caracteres que lo componen (sin contar los espacios en blanco ni los tabuladores), el número total de palabras (secuencias de caracteres que no posean ni espacios ni tabuladores), número de líneas sin tener en cuenta las líneas de comentarios (“//”, “/* ... */”). Para simplificar la realización de la práctica se considerará que el fichero de entrada es correcto.

COMPILADORES – PRÁCTICAS BÁSICAS BLOQUE 1

4º INGENIERÍA INFORMÁTICA/ 1º MÁSTER EN INFORMÁTICA

Práctica 4. Diseño e implementación mediante LEX/FLEX de un analizador léxico para el tratamiento de un listado de alumnos de una asignatura. La entrada del analizador será mediante un fichero en formato texto en el que cada línea presenta los siguientes campos separados por tabuladores: D.N.I. del alumno (con el formato *11111111-X*), apellidos y nombre (con el formato *Apellido1 Apellido2, Nombre*), grupo (formato *1X*), calificación numérica del primer cuatrimestre con dos decimales como máximo (formato 11.11), y calificación numérica del segundo cuatrimestre con dos decimales como máximo (formato 11.11). Se tendrá que efectuar la lectura completa del fichero para obtener: el nombre de la asignatura y el curso académico en formato numérico, listado de los alumnos cuya *nota1* ó *nota2* sea inferior a 5.00 (en este caso sólo se mostrará en el listado el *DNI* y el campo *apellido1 apellido2, nombre* separados por un carácter “;”), listado de alumnos cuya *nota1* y *nota2* sea superior a 5.00 (se mostrará del mismo modo que el anterior listado pero incluyendo también el valor correspondiente a la nota media con dos decimales). Además, será preciso informar de los errores de formato que se han encontrado en el procesamiento del fichero, como por ejemplo líneas en las que falten campos, valores incorrectos de los mismos, etc. Ejemplo de fichero de entrada a reconocer:

```
Nombre de la asignatura - Curso 2011/2012
DNI  Apellido1 Apellido2, Nombre  Grupo  Notal  Nota2
11345678-X  Alonso Barreiro, Antonio  1A  5.68  4.8
AAAAAAA-X  Alonso Barros, Antonio  1A  5.68  4.8
12345678-X  Alonso Calvo, Andrés  1B  8.8  10.00
13345678-X  Barreiro Calvo, Luis  2A  35.68  4.8
```

Ejemplo de salida para el ejemplo anterior:

```
Asignatura: Nombre de la asignatura
Curso: 2011/2012
Alumnos suspensos:
11345678-X;Alonso Barreiro, Antonio
Alumnos aprobados:
12345678-X;Alonso Calvo, Andrés;9.40
Errores:
Línea 4: DNI incorrecto
Línea 6: Notal incorrecta
```

Práctica 5. Diseño e implementación mediante LEX/FLEX de un conversor de formatos bibliográficos XML a BibTeX. El programa debe detectar las distintas palabras clave para poder realizar la conversión de formatos. Además tendrá que detectar errores como palabras clave no reconocidas, errores de formato, etc.

COMPILADORES – PRÁCTICAS BÁSICAS BLOQUE 1

4º INGENIERÍA INFORMÁTICA/ 1º MÁSTER EN INFORMÁTICA

Ejemplo de fichero de entrada en formato XML:

```
<?xml version="1.0"?>
<dblp>
  <article key="journals/kais/LiuLH11" mdate="2011-01-26">
    <author>Hongyan Liu</author>
    <author>Yuan Lin</author>
    <author>Jiawei Han</author>
    <title>Methods for mining frequent items in data
streams: an overview.</title>
    <pages>1-30</pages>
    <year>2011</year>
    <volume>26</volume>
    <journal>Knowl. Inf. Syst.</journal>
    <number>1</number>
    <ee>http://dx.doi.org/10.1007/s10115-009-0267-2</ee>
    <url>db/journals/kais/kais26.html#LiuLH11</url>
  </article>
  <article key="journals/kais/MuhlesteinL11" mdate="2011-01-
26">
    <author>Dennis Muhlestein</author>
    <author>SeungJin Lim</author>
    <title>
      Online learning with social computing based
interest sharing.
    </title>
    <pages>31-58</pages>
    <year>2011</year>
    <volume>26</volume>
    <journal>Knowl. Inf. Syst.</journal>
    <number>1</number>
    <ee>http://dx.doi.org/10.1007/s10115-009-0265-4</ee>
    <url>db/journals/kais/kais26.html#MuhlesteinL11</url>
  </article>
</dblp>
```

Ejemplo de salida producida para el fichero de entrada anterior en formato BibTeX:

```
@article{DBLP:journals/kais/LiuLH11,
  author    = {Hongyan Liu and
               Yuan Lin and
               Jiawei Han},
  title     = {Methods for mining frequent items in data
streams: an overview},
  journal   = {Knowl. Inf. Syst.},
  volume    = {26},
  number    = {1},
  year      = {2011},
  pages     = {1-30},
  ee        = {http://dx.doi.org/10.1007/s10115-009-0267-2},
  bibsource = {DBLP, http://dblp.uni-trier.de}
}
@article{DBLP:journals/kais/MuhlesteinL11,
  author    = {Dennis Muhlestein and
```

COMPILADORES – PRÁCTICAS BÁSICAS BLOQUE 1

4º INGENIERÍA INFORMÁTICA/ 1º MÁSTER EN INFORMÁTICA

```
        SeungJin Lim},
    title      = {Online learning with social computing based
interest sharing},
    journal    = {Knowl. Inf. Syst.},
    volume     = {26},
    number     = {1},
    year       = {2011},
    pages      = {31-58},
    ee         = {http://dx.doi.org/10.1007/s10115-009-0265-4},
    bibsource  = {DBLP, http://dblp.uni-trier.de}
}
```

COMPILADORES – PRÁCTICAS BÁSICAS BLOQUE 1

4º INGENIERÍA INFORMÁTICA/ 1º MÁSTER EN INFORMÁTICA

NORMAS DE REALIZACIÓN Y ENTREGA

Las prácticas del bloque 1 son de carácter **individual** y **obligatorio**, siendo necesaria su adecuada realización y depósito mediante subversion para poder aprobar la asignatura. Deben realizarse en lenguaje **C/C++** sobre **Linux/UNIX** y utilizando únicamente los compiladores y herramientas disponibles en el laboratorio de prácticas de la asignatura (laboratorio 1.2).

La fecha límite de entrega para todas las prácticas de este bloque es el **2 de diciembre de 2011**. Cada una de ellas deberá depositarse en el directorio correspondiente del repositorio de subversion (<https://svn.fic.udc.es/ei4/com/> para ingeniería y <https://svn.fic.udc.es/mi2/com/> para el máster). Además de los archivos de código (practicaX.c, practicaX.l, etc.), se dejarán algunos ejemplos de prueba y un fichero de texto (practicaX.txt) en el que se consignará el nombre del autor y se describirá de forma breve el desarrollo, funcionamiento y peculiaridades de la práctica. También se recomienda incluir un archivo Makefile o bash para facilitar la compilación de la práctica.

En la página de la asignatura del campus virtual está habilitado el “foro de dudas en prácticas”, donde se atenderán todas las cuestiones relacionadas con el desarrollo de estas prácticas básicas (<https://campusvirtual.udc.es/moodle>). Asimismo, se puede contactar en todo momento con los profesores responsables a través del correo electrónico (angel.gomez@udc.es, diego.fustes@udc.es) y en el laboratorio 1.2 en horas de prácticas.