What is a Scanner?

The scanner performs lexical analysis of a certain program. It reads the source program as a sequence of characters and recognizes "larger" textual units called tokens. For example, if the source programs contains the characters

VAR ics142: INTEGER; // variable declaration

the scanner would produce the tokens

VAR  ID(ics142)  COLON  ID(INTEGER)  SEMICOLON

to be processed in later phases of the compiler. Note that the scanner discards white space and comments between the tokens, i.e. they are "filtered" and not passed on to later phases. Examples of nontokens are tabs, line feeds, carriage returns, etc.

How to use FLEX

FLEX (Fast LEXical analyzer generator) is a tool for generating scanners. In stead of writing a scanner from scratch, you only need to identify the vocabulary of a certain language (e.g. Simple), write a specification of patterns using regular expressions (e.g. DIGIT [0-9]), and FLEX will construct a scanner for you. FLEX is generally used in the manner depicted here:

First, FLEX reads a specification of a scanner either from an input file \*.lex. It generates as output a C source file. Then, the source is compiled and linked with the "-lfl" library to produce an executable. Finally, a.out analyzes its input stream and transforms it into a sequence of tokens.  
  
http://alumni.cs.ucr.edu/%7Elgao/Img/dot.gif\*.lex is in the form of pairs of regular expressions and C code.  
http://alumni.cs.ucr.edu/%7Elgao/Img/dot.giflex.yy.c defines a routine yylex() that uses the specification to recognize tokens.  
http://alumni.cs.ucr.edu/%7Elgao/Img/dot.gifa.out is actually the scanner!

1. Run flex on the included programs (\*.lex)  
   Flex –o <prog>.c <prog>.lex  
   Make <prog>  
   <prog> and describe the results where <prog> is the name of the lex source file you are working with.  
   The lex source files are in:  
   Sample1.lex, sample2.lex
2. Write a scanner for recognizing 32-bit hexadecimal numbers
3. Implement the Caesar cipher: it replaces every letter with the one three letters after in alphabetical order, wrapping around at Z. i.e. A is replaced by D, B by E, C by F, etc.