Homework 2

1. Write regular expressions to capture the following regular languages:

   (a) The set of binary strings which have an odd number of 1's or which contain three consecutive 0's somewhere in the string (or both). Hint: this is an OR, so build the two regular expressions separately and then put an OR between them!

   (b) All strings of lower case letters where all five vowels appear at least once and must be in alphabetical order (a-e-i-o-u). (No, I don't believe that y is a vowel.) Feel free to use ranges to simplify, i.e. [b-d] means any of the letters b,c, or d will be accepted, so it is an OR of them.

   (c) Comments as in C++, which either consist of a string which begins with // and ends with a newline, with any text (other than newline) allowed in between, or which are any string surrounded by /* and */, with no intervening */ allowed. Again, feel free to use shortcuts for letters or sets of characters if it will simplify - just define them carefully so I know what you mean.

2. Write a DFA or NFA to recognize each of the languages described in each part of problem 1.

3. (a) Show the NFA that results from applying the standard construction we saw in class (or you can find in the book in Figure 2.7) to the regular expression: $letter\ digit(letter|digit)*$.

   (b) Convert your NFA to a DFA (see Example 2.14 in the book).