EXERCISES ON ERROR-CORRECTING CODES

KEN BROWN

1. Recall that an International Standard Book Number (ISBN) is a 10-digit number $a_1a_2\cdots a_{10}$ satisfying
$$10a_1 + 9a_2 + \cdots + 2a_9 + a_{10} \equiv 0 \mod 11$$
for books published before 2007. [If this forces a_{10} to be 10, then the Roman numeral X is used.]
   (a) Find a book published before 2007 and check that its ISBN is valid.
   (b) The incorrect ISBN 0-669-03925-4 resulted from transposing two adjacent digits not involving the first or last digit. What is the correct ISBN?

2. Recall that a Universal Product Code (UPC) is a 12-digit number $a_1\cdots a_{12}$ satisfying
$$3a_1 + a_2 + 3a_3 + \cdots + 3a_{11} + a_{12} \equiv 0 \mod 10.$$
Why is this better than a scheme that requires $a_1 + a_2 + \cdots + a_{12} \equiv 0 \mod 10$?

Recall that a *word* of length n is a string of n symbols, and a *code* of length n is a collection of words of length n. The *distance* between two words is the number of positions in which they differ. For a code C, we denote by $d(C)$ the *minimal distance* between two distinct code words. The bigger $d(C)$ is, the better C is for error detection. [If you try to type a code word, you have to make at least $d(C)$ mistakes in order to get a valid code word different from the one you tried to type.] For example, if C is the binary repetition code of length 5, consisting of the two words 00000 and 11111, then $d(C) = 5$. And if C is the Hamming code of length 7, then $d(C) = 3$.

3. (a) Find $d(C)$ if C is the set of valid ISBNs.
   (b) Find $d(C)$ if C is the set of valid UPCs.

4. Suppose C is a code with $d(C) = 5$.
   (a) Show that C can *detect* 4 errors.
   (b) Show that C can *correct* 2 errors.
   (c) Prove the following statements, which generalize (a) and (b): A code with $d(C) \geq s + 1$ can detect up to s errors. A code with $d(C) \geq 2t + 1$ can correct up to t errors.

5. Let C be a code with $d(C) = 4$.
   (a) Show that C can be used to *simultaneously* correct 1 error and detect 2 errors. More precisely, give an algorithm that takes a received word y and either produces a code word x or declares "2 errors"; the algorithm should always give the right answer if the number of errors is at most 2.

Date: February 10, 2010.

(b) According to Exercise 4, C can actually detect 3 errors. Why does the algorithm in (a) only detect 2 errors? In other words, you are to explain why we cannot use C to detect 3 errors if we are simultaneously trying to correct 1 error.

6. Read the article about the hat problem at

   `http://www.math.cornell.edu/~kbrown/3360/hat.html`.

   Consider the version of the game with 7 players, and devise a strategy that wins with probability 7/8. [Hint: Use properties of the Hamming code, and note that a random 7-bit word has probability 1/8 of being a code word.]