15110 Summer 2018
Problem Set 7

Name: ____

Andrew ID: ____

Instructions

- Type or neatly write the answers to the following problems.
- Save or scan this file as a pdf and submit to Gradescope

Exercises

1. (4 points) Consider the 8-bit value `10110101`. Some possible interpretations: an unsigned integer, a signed integer, or an ASCII character transmitted with odd parity.
    a. Give the decimal notation for the value if it is interpreted as an unsigned integer and show your work.

    b. Give the decimal notation for the value if it is interpreted as a signed two's complement integer and show your work.

c. If the binary value above represented a character transmitted using *odd* parity but was erroneously received as 10010001, would the receiver be able to detect an error? Why or why not?

d. If the binary value above represented a character transmitted using *odd* parity but was erroneously received as 10110001, would the receiver be able to detect an error? Why or why not?

2. (3 pts) As discussed in class, we can use the notion of *parity* to detect when an error occurs during transmission of data. International Standard Book Number (ISBN) codes use the same idea, called a *check digit* in this context. For this question we consider the older ISBN-10 standard of a 10-digit code. (Modern codes are longer.)

   An ISBN-10 code consists of nine code digits followed by a check digit, which ranges from zero to ten (but ten is represented by 'X'). The nine code digits uniquely identify a book and its publisher. Here are some example ISBN-10 codes: "080442957X", "8535902775", and "9971502100".

   The nine code digits identify a particular book, but the check digit is like a parity bit. It is computed as follows, based on the code digits:

   Set *sum* to 0

   1. Set *mult* to 10
   2. For each of the 9 digits from left to right:
      i. Multiply the digit by *mult* and add the result to *sum*
      ii. Set *mult* to *mult*-1
   3. The check digit is (11 - (*sum* mod 11)) mod 11, or `(11 - sum % 11) % 11` in Python notation.

   Using a multiplier to compute the check digit has the pleasant property that swapping two digits or changing just one digit of a valid ISBN-10 results in an invalid ISBN-10. (You might notice a certain resemblance to one of the hash functions of Lab 7.)

What is the check digit needed to complete the ISBN-10 code 432105419_? Either show your arithmetic or show the program you used to obtain the answer (if you didn't write the program, explain where you got it).

Give an example of an invalid ISBN-10 code and explain why it is invalid.

What is the purpose of having check digits in ISBN codes?

3. (2 pts) For this question, use the Huffman tree for the Hawaiian alphabet shown in lecture on Data Representation, Data Compression, Image and Sound Representation".

   The Hawaiian alphabet has 13 characters (5 vowels, 7 consonants and 1 apostrophe). If we used a fixed-width encoding for characters (i.e. every character is encoded using the same number of bits), we would need a 4-bit code for every character so we could get at least 13 unique codes for the 13 characters of the Hawaiian alphabet:

`	0000
A	0001
E	0010
H	0011
I	0100
K	0101
L	0110
M	0111
N	1000
O	1001
P	1010
U	1011
W	1100

   Go to this list of Hawaiian words and expressions (http://www.mauimapp.com/moolelo/hwnwdshw.htm). Find one word (**not "ALOHA"**) that is shorter when encoded with the Huffman tree than with the 4-bit fixed-width code above, and one that will be longer if encoded with the Huffman tree. (Hint: consider the letter frequencies of the words.) Give the encodings using the Huffman tree and the 4-bit fixed-width codes above to justify your answers.

4. (1 pt) Sound encodings:

   Suppose a sound compression algorithm takes a stream of samples, discards alternate samples (discards every other sample), and stores the result. To decompress and play the compressed data, an algorithm plays each sample in the compressed file twice. Is this compression/decompression technique a *lossless* or *lossy* compression technique? Explain.