22C:16 Practice Problem Set 1

Morning Section: Complete before Tuesday, Jan 28

Evening Section: Complete before Monday, Jan 27, Evening

Note: This is just for practice for Quiz 1 in your first discussion sections. Nothing needs to be turned in.

1. Use the algorithm described in class to find the binary equivalent of 86. Show the working of the algorithm as in Slide 14 in the lecture notes titled "Jan 24."

2. The *greatest common divisor* (GCD) of two numbers (non-negative integers) is the largest number that divides evenly (i.e., without a remainder) into both numbers. For example, the GCD of 12 and 16 is 4. *Euclid's algorithm* computes the GCD of two numbers. It is probably the oldest algorithm still in common use, having been described by Euclid in 300 B.C. It is based on the principle that the GCD of two numbers does not change when the smaller number is subtracted from the larger number. (Convince yourself of this!) For example, the GCD of 35 and 14 is also the GCD of 21 and 14.

   This property can be applied repeatedly to find the GCD of a given pair of numbers. Again, suppose that we want to compute the GCD of 35 and 14. As mentioned above, this is the same as the GCD of 21 and 14. Furthermore, this is the same as the GCD of 7 and 14, which is the same as the GCD of 7 and 7. We know that the GCD of 7 and 7 is 7 and therefore the GCD of 35 and 14 is 7. In other words,

   $$gcd(35, 14) = gcd(21, 14) = gcd(14, 7) = gcd(7, 7) = 7. \quad (1)$$

   (a) Use this algorithm to compute the GCD of 96 and 36 and the GCD of 63 and 27. Show the working of the algorithm using a one line description as in (1).

   (b) Write down this algorithm as precisely as you can in 3-4 sentences.

3. We are all familiar with the notion of expressing numbers in the *decimal* system (i.e., base-10 system), using the 10 digits $0, 1, 2, \ldots, 9$. In class, we discussed the *binary* system (i.e., the base-2 system) that uses the 2 digits 0 and 1. The decimal and binary systems for representing numbers are not the only number representation systems that use place values. The same principles that are used to represent numbers in base-10 or base-2, can be used to represent numbers in a base-b, for any integer $b \geq 2$. In this problem, we will help you understand how numbers can be represented in the *ternary* (i.e., base-3) system.

   First of all, place values in the ternary system are powers of 3. So starting from the rightmost place, the place values are $3^0 = 1$, $3^1 = 3$, $3^2 = 9$, $3^3 = 27$, etc. As you can guess, the ternary system uses 3 digits: 0, 1, and 2. For example, consider the ternary number 20122. The value of this number is $2 \times 3^0 + 2 \times 3^1 + 1 \times 3^2 + 0 \times 3^3 + 2 \times 3^4$. You can check that this evaluates to $2 + 6 + 9 + 162 = 179$. Thus the value of the ternary number 20122 expressed in the decimal system is 179. In other words, the *ternary equivalent* of 179 is 20122.

   (a) The following is an incomplete table of the first twenty whole numbers in decimal and their ternary equivalents. Your task is to complete this table.

Decimal	Ternary
0	0
1	1
2	2
3	10
4	
5	
6	
7	
8	22
9	100
10	
11	
12	
13	
14	112
15	
16	
17	
18	200
19	

(b) Based on what you see in this table, what can you say about the last digit of the ternary equivalent of a decimal number n? In other words, make a precise observation about what the last digit of the ternary equivalent of the decimal number n is, as a function of the value of n. Answer in one sentence.

(c) The above observation helps you get the last digit of the ternary equivalent of a number n. To get the rest of the digits observe a relationship between the ternary equivalent of a decimal number n and the ternary equivalent of a number roughly one-third of n. State this observation precisely. Answer in 1-2 sentences.

(d) Use the above two observations to derive an algorithm to compute the ternary equivalent of a given, nonnegative decimal integer n. Write this algorithm in *pseudocode* similar to the one used in class to describe the algorithm for translating a decimal number into its binary equivalent.
