This document is an overview of the concepts explored in the GS Worksheet 1(Num Rep), and meant as an accompaniment to the solutions document to provide another perspective on the material.

1. Number Representation
   a. This question tests your understanding of the core ideas of number representation by working in base-5 format. For any base (x), each digit provides x possible values. Taken in combination with (n) digits there are x^n possible distinct assignments. As an unsigned number representation needs to use one of those assignments for 0, for x = 5 we get 5^n -1 as the largest number.

   b. The first five powers of 5 are: 1, 5, 25, 125, 625. The easy way to solve this problem is to notice 623 is only 2 less than 625. $10000_5 - 2_{10} = 4443_5$. The more general way to solve it is to work down the powers of 5 from largest to smallest and see how many of each power we can fit into 623: 4 * 125 = 500 leaving 123. 4 * 25 = 100 leaving 23. 4*5 = 20 leaving 3. Again the result is 4443_5.

2. Another Number Representation
   a. This question is a straightforward number representation translation question with the added wrinkle of working with two's complement numbers. First express each hex digit in binary. If the first bit is 0, we can convert the number to decimal as if it was unsigned. If the first bit is 1, we know its negative so remember that and take it's complement (flip all the bits and add 1). Its decimal value is the negative of the unsigned interpretation of the complement.

   b. This part of the question helps to develop your intuition for working in different number representations. You can use the ideas here as a sanity check on your answers to other translation problems. See the solution document for detailed explanation.

3. Back to the Base-ics
   a. Another translation question. If you don't have the 4 digit binary pattern for hex numbers memorized I recommend making a conversion table counting from 0 to 15 with hex on one side and binary on the other. Decimal representations are just the sum of the powers of 2 for each binary digit. Two's complement decimal can be obtained by flipping the bits, adding 1, and taking the negative of the resulting unsigned binary representation when converted back to decimal.

   b. I recommend solving this question by working in decimal. In general n digits in base 7 can represent up to 7^n distinct values. We have 3 digits so we can encode 7^3 = 343_{10} values. Similarly n digits in base 2 can represent 2^n distinct values. So just find the smallest n satisfying 2^n >= 343_{10}.

c. Again 3 base 7 digits can represent 343_{10} values. That means the largest representable number is 343_{10} -1. Half of 342_{10} is 171. That means if we add a translation of -171, 0_{10} is translated to -171, 171_{10} is translated to 0, and 342_{10} is translated to 171. There are 171 numbers in the interval [-171, -1] and 171 numbers in the interval [1, 171].

d. Convert the hex representation to base 10, then convert it to base 7 by the same method as in Problem 1b.

4. Num Rep!
   a. This question explores the range of expressible numbers in different representations.
      i. Any unsigned representation is going to have a lower bound of 0_{10}. The upper bound is given by taking the maximum value for each digit (here 3333_4). You can either convert 3333_4 to decimal or know that the upper bound for n digits for base x is going to be x^n -1 (4^4 -1 = 255).

      ii. Solve this question the same way as problem 1b

   b. This question explores representation of negative numbers
      i. Solve this question like problem 3c. The only difference is we now have an even number of representable numbers, so we use -128 to favor negative numbers.

      ii. Note that using a base 4 digit to represent sign is more wasteful than using a base 2 digit! There are now only two valid possibilities for the most significant digit. The remaining 3 digits have 4^3 representations, in total resulting in 2 * 4^3 = 128.