1. List the members of your group below:

2. Depict the bitwise memory organization of an array containing the first 10 digits of π as 10 4-bit unsigned integers (i.e., as 10 *nybbles*).

3. We write $z = (z_{k-1} \cdots z_2 z_1 z_0)_b$ to denote that the base-b representation of z is $z_{k-1} \cdots z_2 z_1 z_0$. For example, $(0110)_2 = (6)_{16} = (6)_{10} = 6$.

   Exhibit X, Y, Z, and W such that $42 = (X)_1 = (Y)_2 = (Z)_5 = (W)_8$. Justify your answers briefly.

4. Given an array $a[1], \ldots, a[n]$ of n unsigned w-bit integers, and some $k \in 0 \ldots n-1$, we say the array is *bit-k sorted* if $a[i]_k \leq a[j]_k$ for all $1 \leq i < j \leq n$, where $a[i]_k$ denotes the kth bit of $a[i]$ using a right-to-left numbering of bits starting with 0.

   Depict the results of progressively bit-k sorting the array of of Question 2, for $k = 0, 1, 2, 3$. Use a stable sort.

5. Suppose you are given efficient parallel implementations of the *scan* and *prescan* operators discussed in earlier classes. Provide pseudocode for a bit-splitting radix-sort of an n-element array of b-bit unsigned integers that runs in time $O(bn/p + b \log p)$, given $p \leq n$ processors.

6. Depict the action of the algorithm of Question 5 on the array of Question 2.

7. Provide a Lex program that yields a filter that converts all hexadecimal and octal representations in its input to the equivalent decimal form. Hexadecimal representations are indicated by the prefix `0x` followed by a string of hexadecimal digits while octal representations are indicated by the prefix `\0` followed by a string of octal digits.

8. Suppose we wish to implement a version of the popular gimmick *sucks/rules-o-meter*, which counts occurrences of the words "sucks" and "rules" that occur in one of the contexts of a few selected keywords and tabulates the results. In more detail, we define the context of an occurrence of 'sucks' or 'rocks' as the line (newline-terminated) in which it occurs. Each such occurrence is credited to all keywords that occur in its context. The desired output has one line for each keyword. Each line lists the keyword followed by two integers, separated by tabs: the number of times the keyword is in the context of 'sucks' and 'rocks' in the standard input.

   Provide a *Lex* program with the above behavior for the following set of keywords: vi, vim, ed, sed, emacs.