# CSE116 Problem Set #2

For each of the following problems you should start by writing several JUnit tests that will verify correct behavior. Once you have written at least four distinct tests for each of the problems, begin to implement solutions to each of the problems. Be sure to run your unit tests often. For all these methods, when it comes time to implement them, the only methods you may call on a String are charAt(int) and length().

1. Define a method named x2ten that accepts a String and an int as input, and returns (as an int) the base 10 interpretation of the String interpreted as a number in the base given by the int input. For example, x2ten("1101", 2) must return 13. For bases beyond 10 use upper-case letters starting with 'A' as digits. In base 16 the digits used are therefore '0' through 'F'. Your method must work with bases in the range 1 through 36. For illegal inputs (Strings with illegal characters or bases outside the allowable range) the method must return -1.

2. Define a method named ten2x that accepts two ints as input, and returns a String representing the first int (a base 10 number) expressed in the base given by the second int. For example, ten2x(13,2) must return "1101". Your method must work with the same range of bases as in question 1. For illegal inputs the method must return null.

3. Define a method named sumOfDigits that accepts a String and returns an int which is the sum of the digits contained in the String. For example, sumOfDigits("1234") must return 10, and sumOfDigits("338 Davis Hall, Buffalo NY 14260") must return 27.

4. Define a method named x2y that accepts a String and two ints as input, and returns (as a String) the base y interpretation of the input String, assuming it is in base x. Thus, x2y("1101", 2, 10) must return "13", and x2y("13",10,2) must return "1101".

5. Define a method named largest that accepts an array of int and returns the largest value from the array.

6. Define a method named indexOfLargest that accepts an array of int and returns the index of the largest value from the array.

7. Define a method named countAll that accepts a String as an argument. The method must return an array of int of size 27, such that the value in position 0 is a count of the number of 'a' and 'A' characters in the input, the value in position 1 is a count of the number of 'b' and 'B' characters in the input, ... the value in position 25 is a count of the number of 'z' and 'Z' characters in the input, and the value in position 26 is a count of all the non-alphabetic characters in the input. For example,
   - if s is "", then all entries in the array must be 0.
   - if s is "a", then all entries in the array must be 0 except for entry 0, which must be 1.
   - if s is "Baaa!", then all entries in the array must be 0 except for:
     - entry 0, which must be 3,
     - entry 1, which must be 1, and
     - entry 26, which must be 1.

   The only methods you may call on the String s are charAt(int) and length(). You may use the static toLowerCase method defined in the Character class, which maps a char to its lower-case equivalent. For example,
   - Character.toLowerCase('a') returns 'a'
   - Character.toLowerCase('A') returns 'a'
   - Character.toLowerCase('%') returns '%'

   You may, if you wish, define private helper method that you call from your character counting method.