1. Problem 6.5 in Louden, page 251.

x and y are static arrays. In other words, x and y really are arrays, not pointers. C does automatic type conversion between arrays and pointers when necessary. However, in this case, x is not a pointer and cannot be viewed as one. Assignments between arrays are not legal in C. If you want the assignment to work, you need either x to be a pointer (int *x) or to run a loop explicitly copying y to x one int at a time. (Or, you could use the memcpy function, which can copy whole arrays.)


In Java, just as in C/C++, all floating-point constants are of type double. In C/C++ copying a double to a float results in a warning (if even that), but in Java it is illegal. To fix it, you could redefine x as a double: double x = 2.1; cast x as a float: float x = (float) 2.1; or use floating-point constant notation (that also exists in C/C++): float x = 2.1f.


Allowing any non-zero value to signify “true” allows the programmer flexibility to write slick code taking advantage of this. For example, in C rather than type “if (x==0)” a programmer will frequently type “if (!x)” Such coding would not be possible if 1 were the only acceptable value for “true.” However, explicitly coding internal logical values to be 0 or 1 (such as in a <= b) also gives the programmer flexibility. If the programmer knows EXACTLY what value is going to be returned s/he can write code expecting it, rather have to allow for a number of unpredictable values.

4. I have a C-like language, except that it has no floating-point numbers. Describe how I can emulate floating-point numbers using only integers.

There are many ways to do this. I would just use two integers to represent a numerator and denominator, and just use fractional arithmetic for floating-point operations.

5. I have two C-like languages. Both of them allow for unsigned integers to be arbitrarily large. However, one of them has arrays and the other doesn't. Show that these two languages are equally powerful by demonstrating how to emulate an array of arbitrarily large unsigned integers with a single unsigned integer.
Encode each array into a single integer, just as shown in class. Then, combine that integer with the size of the array. When the integer is converted to an array, the size is pulled out first, and then the other integer is converted into an array of that size.