1. Describe an application of programming to Cognitive and Information Science (in their broadest sense) that you think the class could spend one week working on. Explain the simplest example of what we could do, and what we would learn from it. Provide enough details (including a couple of useful references) to actually do this during the semester.

Any ideas welcome. If you are struggling, you can see some examples in the book *Think Complexity*. You can also take a look at the table of contents of this book on *The Computational Beauty of Nature*. You can also read a little more about agent-based modeling in Scholarpedia.

2. Write a function that solves the *Farmyard problem* (from the 5th grade). You have a farm with some number pigs and chickens. The function should prompt the user to input the number of heads. Then it should prompt the user to input the number of legs. (Work under the assumption there are no amputees). The program has to figure out how many chickens and how many pigs there are in the farm and print the result back to the user.

Use a brute force algorithms. That is, try exhaustively enumerating all of the possible solutions until you find one that works.

3. Change the function above in two ways. First, the program should receive the number of heads and legs as parameters to the function (as opposed to prompting the user for input interactively). Second, the program should return the number of chickens and pigs. Note you will have to return two values.

4. Write a function that computes and prints the Nth prime number. Test the program with the 1000th prime.

Divide the program into simpler subcomponents and create functions that the main function can use. For example, you can define a function isprime(x) that returns True if x is prime and False if x is not. Also, you may need to obtain the remainder after dividing two integers, a % b. Finally, to figure out whether your program is correct, you can see the first 1000 prime numbers here.

5. [Advanced optional] 145 is a curious number, as 1! + 4! + 5! = 1 + 24 + 120 = 145. Find other numbers which are equal to the sum of the factorial of their digits. Note: as 1! = 1 and 2! = 2 are not sums, so they are not included.