Final Proficiency Prep – New Beginnings Summer 2018

Here are some “number” problems that may help prepare you for the final proficiency exam. (I’ll send out List problems later today.)

1. From CodeWars(https://www.codewars.com/kata/sum-of-odd-numbers)
(Look for an algorithmic way of solving the problem – without generating an array of odd numbers. What patterns in the triangle can you find?)

   Given the triangle of consecutive odd numbers:

   ```
                1
             3     5
          7     9    11
      13    15    17    19
   21    23    25    27    29
   ...
   ```

   Calculate the row sums of this triangle from the row index (starting at index 1) e.g.:

   ```
   rowSumOddNumbers(1); // 1
   rowSumOddNumbers(2); // 3 + 5 = 8
   ```

2. Write a function that outputs the triangle shown above, but instead of odd numbers, only include prime numbers (in a pyramid format!) The function prototype is:

   ```
   void pyramid_of_primes(int h);  // where h is the height in rows of primes.
   ```

   Be sure it is a symmetric pyramid.

3. Write a function that outputs the first n Fibonacci numbers. (https://en.wikipedia.org/wiki/Fibonacci_number) The function prototype is:

   ```
   void fibonacci(int n);  // where n is the number of Fibonacci’s to print.
   ```

   To remove any ambiguity, the first two Fibonacci numbers are 0 and 1.

4. Write a function to algorithmically generate all the “happy numbers” between 1 and 1000. (https://en.wikipedia.org/wiki/Happy_number). How efficient can you make it? Place a counter in your function and output the number of comparisons you had to execute. As we talked about in class, comparisons are “one” the most expensive instructions and can be used to evaluate the “cost” of an algorithm. (I’ll have a solution for your to compare against on Tuesday.