CSCI 2824 - Discrete Structures
Homework 4

You MUST show your work. If you only present answers you will receive minimal credit. This homework is worth 100pts.

Due: Wednesday July 5

For all of the following questions you are expected to write a program to solve this. If you write a simple, straight-forward program to solve these, you will *eventually* solve these. Thus you'll need to think at least a little bit about optimization, you should definitely ask questions if you don't know how to optimize some of these (it is not *expected* that you know).

All of these questions come from Project Euler (though I've modified some), a website which assigns various math problems to be solved by a computer. If you like these problems you should check it out!

For this assignment, you should submit your code for each problem, so at least 5 files. However the Moodle submission will only allow one file, thus you'll need to make a *tarball* again. The files that contain your main method should be named Problem$k$.[py,cpp,...] where k is the number of the problem.

For this assignment I will require that you include a Makefile, if your code requires any compilation. Thus languages such as `C/C++, Go, Rust, Java,...` will require one additional file in *tarball*. A Makefile is a file that handles compilation for the user. There are many examples of Makefiles on-line. But if you have questions on how to make one please ask. When grading this, the grader should only need to un-tar your file, run `make` (to compile all files), and then run each file on its own.

Since this is a code-writing assignment, there is no opportunity for LaTeXextra credit. However if all 5 of your solutions are in a language other than `Python, C/C++` then you will get 5 extra credit points.

In the comments for each problem you should have the answer, so I can verify by reading your comments, in case your code is very slow.

None of these problems is particularly difficult, so I would encourage you to use this time to try a new language!

1. (10 points) Work out the last 10 (10 least significant) digits of the sum of the 100, 50-digit numbers in `nums.txt`

2. (10 points) The Collatz function is defined as followed:

$$C(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ 3n+1 & \text{if } n \text{ is odd} \end{cases}$$

   Though it has not been proven, every number that you will likely choose to plug into this function will eventually reach 1, though iteration, for example starting with 13:

$$13 \to 40 \to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$$

   We can see that this sequence (from 13 to 1) has 10 terms. Your goal is to figure out which number, under 1 million has the longest chain, between starting value and 1.

   Note: once the chain starts, it may go above 1 million, this is fine.

3. (25 points) In class we discussed the Fibonacci sequence defined as:

$$f_n = f_{n-1} + f_{n-2}; \qquad f_1 = 1, f_2 = 1$$

   Consider only the values of the Fibonacci sequence which do not exceed 4 million, what is the sum of all the even terms? Even terms meaning terms which are even, not that their index is even.

4. (25 points) 2520 is the smallest number which can be divided evenly by each of the numbers between 1 and 10. What is the smallest positive integer that is evenly divisible by all of the integers from 1 to 20?

5. (30 points) Using `names.txt` a 46K text file containing over five-thousand first names, begin by sorting it into alphabetical order. Then working out the alphabetical value for each name, multiply this value by its alphabetical position in the list to obtain a name score. For example, when the list is sorted into alphabetical order, COLIN, which is worth $3 + 15 + 12 + 9 + 14 = 53$, is the 938th name in the list. So, COLIN would obtain a score of $938 \times 53 = 49714$. What is the total of all the name scores in the file?

   Note: for this problem we are using the the order as the multiplier, not the index. Thus the first name will be multiplied by 1, not by 0, etc.