4330 Assignment 2 [1]

For this assignment, you need to submit a Python program for Problem 1 and a neatly handwritten proof for Problem 2. The Python file should be named:

$$\boxed{\texttt{hw2-}\textit{lastname}\texttt{.py}}$$

Please: name your file in exactly this way; lowercase 'hw', a dash (not an underscore), and **NO SPACES** in the filename! For Problem 2, you may submit a your handwritten argument in class, or a scanned PDF on Blackboard with your solution to Problem 1.

You must complete Problem 1 according to the "Structured Coding Process" outlined at the end of this document. It will be graded according to that rubric.

---

(1) (20 points) Suppose $c \geq 1$ is a real number, and consider the sequence $\{x_n\}$ defined by

$$\begin{aligned} x_0 &= 1, \\ x_n &= \frac{1}{2}\left(x_{n-1} + \frac{c}{x_{n-1}}\right), \qquad \text{for } n \geq 1. \end{aligned}$$

Write a Python program which does the following:

(i) Input a floating point number `c` greater or equal 1 from the user.

(ii) Input a positive integer `N` from the user.

(iii) Compute the term x_N of the sequence defined above, for the given values of c and N, and print it out with 8 decimal digits. Do NOT use a list or other data structure to store every term in the sequence.

---

(2) (10 points) Prove that if $\lim_{n\to\infty} x_n$ exists, then it equals $\sqrt{c}$.

Note: it is a little more difficult, but also possible, to prove that this limit does exist; however, you do not need to do that.

[1] This document is copyright 2021 Chris Monico, and may not be reproduced in any form without written permission from the author

4330 A Structured Coding Process [2]

This describes the process that you are to apply to every assignment this semester, after the first two. It will be used as a grading rubric for your assignments as well. In particular, note that:

- 60% of the available credit does not depend explcitly on any programming at all.
- A program (working or not) without (I), (II), and (III) as described below can only earn up to 40% of the available credit.

(I) Understand the problem (20%):
Read and think about the problem description, and work out an example of your own choosing by hand. It should be a 'small' enough example that working it out by hand is reasonable, yet 'large enough' to capture most things that you think might happen. Include this in comments at the top of your Python program file.

(II) Develop an approach: (20%):
Explain, in words, how to solve a general instance of the problem by hand. Include this in comments at the top of your Python program file.

(III) Test your approach by hand: (20%):
Take the method you just described, and apply it to a different example by hand. Verify that it works correctly. Include this in comments at the top of your Python program file.

(IV) Code it: (20%):
Translate your "in words" method into Python code.

(V) Test and debug: (20%):
Attempt to use your code on the examples you already worked out by hand, and several more. Try hard to think of examples where something could fail. If it fails on any of those, determine whether the problem comes from (III) or (IV), by adding print statements, if necessary, and working the example by hand.

You will be required to go through these steps for each assignment from Assignment 3 onward. You must clearly indicate what you've done for Steps (I), (II), and (III) in comments at the top of your Python program file. The code itself will be used to grade steps (IV) and (V).

[2]