1. Approximation of $e$ to 4 digits of accuracy

The Maclaurin expansion of $e^x$ is given by

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

Compute a sequence of approximations of $e$ (using $x = 1$) until the approximate relative error $\varepsilon_a$ is small enough to guarantee at least four significant digits of precision. Submit a table showing each estimate along with the approximate relative errors (for all but the first estimate).

2. Programming exercises

This question is to get you thinking about programming in Python in case you’re just starting out, or it’s been a while. If you’re feeling comfortable with Python and you want to try something different you may work out solutions to any two Project Euler problems (see https://projecteuler.net/) you have not yet completed instead of doing the exercises below.

On the course website, I’ve posted two Python programs, `guess1.py` and `guess2.py`. Your goal is to modify the programs to obey these guidelines:

- Modify both programs so that the guesser is limited to a maximum of 7 guesses, and have the program print out which guess is being made (i.e. 1 to 7). Use a `for` loop to limit the guesses, and use a `break` statement to exit the loop when the number has been guessed correctly.

- Modify `guess1.py` so that the computer tells the user whether their guess is too high or too low, instead of whether it is correct or incorrect.

- Modify `guess2.py` so that the user can tell the computer whether its guess is too high, too low, or correct by entering the letters h, l, or c, and have the computer implement an intelligent strategy for guessing based on the feedback. If you need strategy help, read this: http://en.wikipedia.org/wiki/Binary_search_algorithm.

- **Optional:** make your modified `guess2.py` program print out an angry message if the user is found to be cheating.

I will provide instructions on how to submit your programs online.