MATH 210 Assignment 2

Basic Python, List Comprehensions, Logic and Functions

INSTRUCTIONS

- Create a new Python 3 Jupyter notebook
- Answer each question in the Jupyter notebook and clearly label the solutions with headings
- There are 18 total points and each question is worth 3 points
- Submit the `.ipynb` file to Connect by **11pm Monday, January 23, 2017**
- You may work on these problems with others but you must write your solutions on your own
- Do **not** import any Python packages such as `math` or `numpy` to complete this assignment. These questions require only the standard Python library. Solutions will be given 0 if any Python package/module is used.

QUESTIONS

1. Write a function called `linear` which takes 3 input parameters `start`, `stop` and `by` (in that order) and returns the list of numbers (as floats) beginning with `start`, incremented by the value `by` and up to (but not exceeding) `stop`. For example:

   `linear(0,10,2)` returns the list `[0.0, 2.0, 4.0, 6.0, 8.0, 10.0]`

   `linear(3,4,0.25)` returns the list `[3.0, 3.25, 3.5, 3.75, 4.0]`

   `linear(-2,9,1.5)` returns the list `[-2.0, -0.5, 1.0, 2.5, 4.0, 5.5, 7.0, 8.5]`

   The input `start` must be less than `stop` therefore the function `linear` should test if `start < stop` and:

   If `start < stop`, then compute and return the list as described above

   If `start >= stop`, then display the message `'Error: First argument must be less than the second argument'` and return the value `None`

2. Write a function called `squares_between` which takes 2 input parameters a and b (in that order) and returns the list of squares in the closed interval $[a, b]$. (Recall, an integer n is a square if there is an integer d such that $n = d^2$.) For example:

   `squares_between(0,10)` returns the list `[0,1,4,9]`

   `squares_between(50,100)` returns the list `[64, 81, 100]`

   `squares_between(26.9,160.2)` returns the list `[36, 49, 64, 81, 100, 121, 144]`

If there are no squares in $[a, b]$ then the function should return the empty list `[]`.

3. Write a function called `phi` which takes 2 input parameters a and N (in that order) and returns the sum

$$\phi(a, N) = 1 + 2\sum_{k=1}^{N} \frac{1}{(ak)^3 - ak}$$

The number a must be greater than 1 and N must be greater than or equal to 1 therefore the function `phi` should test these conditions and:

If $a > 1$ and $N \geq 1$, then compute and return the sum as described above

If $a \leq 1$ or $N < 1$, then display the message `'First argument must be greater than 1 and second argument must be greater than or equal to 1'` and return the value `None`

4. Write a function called `eig` which takes 4 input parameters `a`, `b`, `c` and `d` (in that order) representing the matrix

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

and returns the eigenvalues of the matrix as a list of length 2 (in any order).

5. Write a function called `max_eig` which takes 4 input parameters `a`, `b`, `c` and `d` (in that order) representing the matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

and returns the maximum eigenvalue of the symmetric matrix AA^T.

6. Write a function called `longest_string` which takes a list of strings and returns the length of the longest string. For example:

`longest_string(['Hello','Bonjour','Hola','Ciao'])` returns 7

`longest_string(['one','two','three','four','five'])` returns 5