Tutorial MATH 1MP3 – January 19, 2017

The following questions are to be done in groups of two or three.

- Open PyCharm and create a new project (File -> New Project, make sure the drop down box in “interpreter” points to your Anaconda installation!)
- In this project create a new python file (File -> New, then click “Python File” in the menu that appears)
- Write the code for the following questions in that python file. Just have each question answer follow the previous one in the code. (I recommend doing each question one at a time, though)

For the following questions, write a python program (in PyCharm or otherwise) that prints the answer. The first question is answered for you to get you started.

1) In many software-related job interviews employers pick a language and have applicants write what is called a “fizz-buzz” program in it. They use this to weed out applicants with computer-related degrees who are not strong in programming. The program is simple:
   It should print all the numbers from 1 to 100 inclusive. If a number is divisible by 3 it should instead print “fizz”, if a number is divisible by 5 it should instead print “buzz”, finally if a number is divisible by both 3 and 5, it should instead print “fizzbuzz”.

   Using Python write a “fizz-buzz” program.
2) Write python code to produce the following pattern: You should include some variable `n` which represents the biggest number to print in the “pyramid” (in this case, it should be 5), so you could change `n` on the fly and get a bigger (or smaller) pyramid.
   ```
   1
   22
   333
   4444
   55555
   ```
3)
   a) In python there is a function called `index` that works on lists. Using the “`help`” command in the python console (see Tutorial 2: Question 3 if you’ve forgotten this command) to see what index does.
   b) Write your own python code to operate on the list
      `lista=[‘a’,’b’,’c’,’d’,’e’,’f’,’g’,’h’,’g’,’f’,’g’]`
      Make a variable called `letter` and have your code find the index of the first appearance of `letter` in `lista`.
   c) Modify your code so that it prints out all the indices of any appearance of `letter` in `lista`.
4)
   a) Take a vector called `vector1 = [10,20,15,12,18]` and write code to find the MEDIAN of the vector.
   b) Update your code using an `if` statement so that it can find the median of `vector1` even if you remove the element 18 from the list.
5) `While` loops are like `for` loops, but the nice thing about `while` loops is you don’t need to know how many times they are to execute before they finish (which you do for `for` loops).
   There is a particular population of bacteria whose year-to-year population changes according to this function
   ```
   New_Population = 6 * (Old_Population) / (2 + (Old_Population))
   ```

I.E. If the `Old_Population` was 18, then the N`ew_Population` would be `6*18/(2+18)=5.4`

a) Write a program using a `for` loop that starts with a population of 120. In each iteration of the loop calculate the new population based on the population from the last time through the loop. Have this `for` loop run 100 times. What is the result?

b) If you did a) correctly, your result should be very close to 4. Now say the scientists modelling this bacteria cared about figuring out after how many years it took to get the population *close enough* to 4. Using a `while` loop we want to figure out how many years it takes until the population starts with 4.00 (i.e. it's 4.001 or 4.009423, as long as those first two numbers after the decimal are 0). (Hint, you want the (population – 4) to be less than what? Maybe use that condition as the `while` loop condition?)