Do the following problems.

(1) Write a function that takes two arguments, numeric values, and returns the larger of the two (make an arbitrary decision if the values are the same - return either one, it doesn't matter).

(2) Write functions that achieve the following: a function called calc_average, that should accept 5 test scores (as five numeric values) as arguments, and return the average of the scores. A function called determine_grade should accept a single score as an argument, and return a letter grade (a string) based on that score (90-100 is A, 80 - 89 is B, 70 - 79 is C, 60 - 69 is D, below 60 is F). Finally, the function question2_driver should ask the user for 5 test scores, then USE the previous two functions to calculate the overall letter grade for the student based on those scores. That means that question 2 driver should CALL both the previous functions, and print the results, nicely.

(3) A prime number is a number that is only evenly divisible by itself and by 1. For example, 5 is prime, because you cannot evenly divide it by anything else. 6 is NOT prime because it's evenly divisible by 2, and 3 as well as 1 and 6.

(a) Write a boolean function, isPrime, that takes an integer as a number, and returns True if that number is a prime number, and False otherwise.

(b) Write a driver function that asks the user to enter a number, and uses isPrime from (a) to report on if that number is a prime number or not. Print the answer nicely (i.e. it IS a prime, or it ISN'T a prime).

(4) Write a function that has no parameters, that goes through every number from 1 to 100, and displays ONLY the prime numbers. It must use isPrime, from 3. Use the right loop.

(5) Write a function that accepts two arguments, a list and a number, n. Assume the list contains numbers (integers and floats). The function should return a new list that contains all the of numbers from the original list that are larger than the number n.

(6) Imagine there is a multiple choice exam. The correct answers are given in the text file correct.txt, attached to this problem set. Create your own text file called answers.txt. Use the same format as correct.txt - one answer per line.

Create code that: Read in the correct answers from correct.txt, and return the correct answers in a single list. Read in the student answers from answers.txt, and return the answers in a single list. Compares the two lists, and returns the number of correct answers. Determines if a student has passed the exam, based on a number of correct answers, and a passing grade (for example, 75% required to pass).

How you divide this up into functions is up to you. Think about what makes sense, based on the idea of each function doing only one thing.

Type up your answers, and submit them through Nexus. For python code, I expect to see:

1. Code listing (with comments)

Please make sure to add your name, and [CSC103] Problem Set 6 at the top of your answer file.