



<b>Problem Set:</b>	Assignment: AG05	<b>Semester:</b>	Fall 2018
<b>Points:</b>	See autograder		
<b>Date Set:</b>	See autograder	<b>Due Date:</b>	See autograder
<b>Course:</b>	CS118 Prog. Fundamentals	<b>Instructor:</b>	Dr. Nauman

## 1 Primes and Factors

Since you are reading this, you have already downloaded and extracted the zip file.

### 1.1 Tasks to do

1. Open the file `a05.py` and look between the markers. You may ignore the code outside the markers completely. You may run the code by typing the following from the shell: `python a05.py`  
This will not run the tests but the code itself.
2. Assumptions and requirements:
  - (a) You must not use the functions in the `math` or another library during this assignment. If you break this rule, you risk getting zero score on the whole assignment.
  - (b) For the whole assignment: A *Prime Number* can be divided completely (i.e. leaving no remainder) only by 1, or itself. And it must be a whole number greater than 1.<sup>1</sup>
  - (c) For this assignment, you will have to do some searching to solve the whole problem.
3. There are three main tasks to complete.
  - (a) Write a function with the name `is_prime` that takes in one number and decides if that number is prime. It should return a boolean value depending on the decision.  
You must use the definition of a prime number given above. Moreover, a real number – which has some value other than 0 in the mantissa (i.e to the right of the decimal point) – is not a prime number. However, if a real number with only 0s in the mantissa is passed, it should be considered as the integer counterpart. For instance, 6.0 should be considered as 6 and 5.0 should be considered as 5. However, 6.01 will not be prime and neither will 5.01.
  - (b) Write another function by the name `output_factors`. This function should output the factors of the number passed as input – one factor on each line. Factors of a number, say  $x$ , are those whole numbers which can be multiplied with other whole numbers to get  $x$ .  
For instance, if `output_factors` is given the number 10, it should output the following:  
1  
2  
5  
10
  - (c) The third function that you should define is `get_largest_prime`. This function will be passed a number and it should return the largest prime number that is *smaller than or equal to* than this input.  
For example, if we pass the number 10 to this function, it should return 7 since 7 is the largest prime that is smaller than 10.  
If no prime matching the rules is found, the function should return `None`.
4. You may change the values in function calls at the end of the file `a05.py` to check the functions.
5. Run local tests by issuing the command `py.test` in the folder where your files are located.

---

<sup>1</sup>Adapted from: <https://www.mathsisfun.com/definitions/prime-number.html>