Course 12-746
Python Prototyping for Infrastructure Systems
Fall 2016
Assignment 1 (Total: 8 points)

Due: Tuesday, Sept. 13th, by Blackboard by noon. For each day the assignment is late, a penalty of 10% of your total grade will be deducted.

Task 1 (1 point) *Print the multiplication table*

a. Write **two** Python programs to print the following multiplication table, one program using the ***for*** and one using ***while*** loop:

```
1 * 1 = 1, 1 * 2 = 2, 1 * 3 = 3, 1 * 4 = 4, 1 * 5 = 5, 1 * 6 = 6, 1 * 7 = 7, 1 * 8 = 8, 1 * 9 = 9,
2 * 1 = 2, 2 * 2 = 4, 2 * 3 = 6, 2 * 4 = 8, 2 * 5 = 10, 2 * 6 = 12, 2 * 7 = 14, 2 * 8 = 16, 2 * 9 = 18,
3 * 1 = 3, 3 * 2 = 6, 3 * 3 = 9, 3 * 4 = 12, 3 * 5 = 15, 3 * 6 = 18, 3 * 7 = 21, 3 * 8 = 24, 3 * 9 = 27,
4 * 1 = 4, 4 * 2 = 8, 4 * 3 = 12, 4 * 4 = 16, 4 * 5 = 20, 4 * 6 = 24, 4 * 7 = 28, 4 * 8 = 32, 4 * 9 = 36,
5 * 1 = 5, 5 * 2 = 10, 5 * 3 = 15, 5 * 4 = 20, 5 * 5 = 25, 5 * 6 = 30, 5 * 7 = 35, 5 * 8 = 40, 5 * 9 = 45,
6 * 1 = 6, 6 * 2 = 12, 6 * 3 = 18, 6 * 4 = 24, 6 * 5 = 30, 6 * 6 = 36, 6 * 7 = 42, 6 * 8 = 48, 6 * 9 = 54,
7 * 1 = 7, 7 * 2 = 14, 7 * 3 = 21, 7 * 4 = 28, 7 * 5 = 35, 7 * 6 = 42, 7 * 7 = 49, 7 * 8 = 56, 7 * 9 = 63,
8 * 1 = 8, 8 * 2 = 16, 8 * 3 = 24, 8 * 4 = 32, 8 * 5 = 40, 8 * 6 = 48, 8 * 7 = 56, 8 * 8 = 64, 8 * 9 = 72,
9 * 1 = 9, 9 * 2 = 18, 9 * 3 = 27, 9 * 4 = 36, 9 * 5 = 45, 9 * 6 = 54, 9 * 7 = 63, 9 * 8 = 72, 9 * 9 = 81,
```

b. Write **two** Python programs to print the following multiplication table, one program using the ***for*** and one using ***while*** loop:

```
1 * 1 = 1, 1 * 2 = 2, 1 * 3 = 3, 1 * 4 = 4, 1 * 5 = 5, 1 * 6 = 6, 1 * 7 = 7, 1 * 8 = 8, 1 * 9 = 9,
2 * 2 = 4, 2 * 3 = 6, 2 * 4 = 8, 2 * 5 = 10, 2 * 6 = 12, 2 * 7 = 14, 2 * 8 = 16, 2 * 9 = 18,
3 * 3 = 9, 3 * 4 = 12, 3 * 5 = 15, 3 * 6 = 18, 3 * 7 = 21, 3 * 8 = 24, 3 * 9 = 27,
4 * 4 = 16, 4 * 5 = 20, 4 * 6 = 24, 4 * 7 = 28, 4 * 8 = 32, 4 * 9 = 36,
5 * 5 = 25, 5 * 6 = 30, 5 * 7 = 35, 5 * 8 = 40, 5 * 9 = 45,
6 * 6 = 36, 6 * 7 = 42, 6 * 8 = 48, 6 * 9 = 54,
7 * 7 = 49, 7 * 8 = 56, 7 * 9 = 63,
8 * 8 = 64, 8 * 9 = 72,
9 * 9 = 81,
```

Task 2 (1 point) *Calculation of the Char type variables*

In most programming language, letters/characters are represented using an integer number. Take a look at the ASCII table at http://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html. For example, character ‘a’ corresponds to the integer 97.

Hence, it is possible to “calculate” character using mathematic calculation. For example, ‘a’ + 2 = ‘c’

In Python, an integer can be transformed to a character using this command:

character = chr(97) # The value of variable character will be a string ‘a’

Please write **two** Python programs, one program using a *for* loop and one using a *while* loop to print out the alphabet:

ABCDEFG…XYZabcdefghijk….xyz

Task 3 (4 point) *Write methods*

Please write methods to:

a. Calculate the maximum value of two double type variables (1 point)
Hint: The method should have two parameters, and print a value.

Please use this method to calculate the maximum value of these pairs of numbers:
(1000.5, 1000); (500,5000);

Can this method be used to calculate two char type variables? Try it out:
('C', 'P')

b. Calculate the distance of two points in 2D (1 point)
Hint: The method should have four parameters, and print a value. The four parameters represent the x and y coordinates of two points.

Please use this method to calculate the distance of these pairs of points:
(Point 1: X = 100, Y = 200; Point 2: X = -100, Y = -400)
(Point 1: X = 0.5, Y = 0; Point 2: X = 101.5, Y = -300)

c. Fermat's Last Theorem (2 points)
Fermat's Last Theorem says that there are no positive integers a, b, and c such that
$a^n + b^n = c^n$
for any values of n greater than 2.

1. Write a function named `check_fermat` that takes four parameters—`a`, `b`, `c` and `n`—and that checks to see if Fermat's theorem holds. If n is greater than 2 and it turns out to be true that

   $a^n + b^n = c^n$
2. the program should print, "Holy smokes, Fermat was wrong!" Otherwise the program should print, "No, that doesn't work."
3. Write a function that prompts the user to input values for `a`, `b`, `c` and `n`, converts them to integers, and uses `check_fermat` to check whether they violate Fermat's theorem.

Task 4 (2 point) *Integer Factorization*

1. Please implement a function that takes an integer as input and judge whether this integer is a prime number. The function should return 1 if the input is prime, return 0 if otherwise. The function may be something similar as below:

```
def isPrime (n):
    ……..
    return  output   # either 1 or 0
```

   Hint: https://en.wikipedia.org/wiki/Prime_number

2. Use your first function, implement another function to find the largest prime factor of an integer. You may need to Google on how to judge if the residual of a division is zero. The function may be something similar as below:

```
def largestFactor (n):
    maxfactor = n
    while …… :
    return  maxfactor
```

3. Use your second function, try to implement a third function that prints a list of prime factors of an integer (Optional).