Assignment 3

Functions

Make sure you are using functions in this assignment, and in all your coding going forward. Once you create a function, try to make use of it (by "calling it") wherever possible.

1. Write a function that takes as input a Fahrenheit value and returns the corresponding Celsius value.
HINT: Use the formula C = (5/9)(F – 32). Make sure you are not doing "integer division".

2. Take as input a character ch (eg: 'g', 'h', 'A', '5', '+'). Write a function that returns true if the character is uppercase and false otherwise. Print the value returned.

3. Take as input two numbers x and n. Write a function which calculates and returns the x^n. Print the value returned.

4. Write a function that returns true if a given number is prime, and false otherwise.

5. Take as input two numbers x and n. Write a function which calculates and returns the $Log_n(x)$ of two numbers. Print the value returned. Assume that values of x and n are such that the result is going to be a whole number.

6. Take as input a number N and P. Write a function which returns the square root of N within +/- 0.0001 error range. Print the value returned.
HINT: Keep making better guesses (Newton's method).

7. Take as input a number N. Write a function which returns the integral part of square root of the number. Print the value returned.

8. Take as input the following
    a. A number (eg: 31416)
    b. A digit (eg: 1)

Write a function that returns the number of times digit is found in the number. Eg: 1 is found **2 times** in 31416 Print the value returned.
HINT: To get the last digit of 1234, you can get its remainder when divided by 10... (1234 % 10 = 4). Now divide 1234 by 10 to get 123 (integer division).

9. Take as input a number. Write a functions which returns true if the number is Armstrong number and false otherwise. Print the value returned. An example of **Armstrong number** is 371 because $371 = 3^3 + 7^3 + 1^3$

10. Take as input two numbers N1 and N2. Write a function to print all Armstrong numbers between N1 and N2. **HINT**: “Call” the function from previous question inside this function.

11. Take as input two numbers N1 and N2. Write a function which calculates and returns the **GCD (= HCF)** of two numbers. Print the value returned.
HINT: You can use Euclidean algorithm. GCD(a, b) = GCD(a, b % a) where b > a.

12. Take as input two numbers N1 and N2. Write a function which calculates and returns the **LCM** of two numbers. Print the value returned. **HINT**: N1 x N2 = GCD x LCM

13. Take as input two numbers N1 and N2. Write a function which prints first N1 terms of the series 3n + 2 which are not multiples of N2.

14. Take as input a number N. Following this, take N more inputs from the user to form a sequence S = s1, s2, .., sN. Compute if it is possible to split sequence into two sequences s1 to si and si+1 to sN such that first sequence is strictly decreasing and second is strictly increasing. Print true/false as output.

15. Write a program that works as a simple calculator. It reads two integers and a character.
    a. If the character is +, the sum is printed.
    b. If it is -, the difference is printed.
    c. If it is *, the product is printed.
    d. If it is /, the quotient is printed.
    e. If it is %, the remainder is printed.
    f. If the user enters 'X' or 'x', the program exits, otherwise again asks for two numbers and a new operation.

16. Write another function that takes as input 3 values:
    a. **start** - This represents the starting Fahrenheit value
    b. **end** - This represents the ending Fahrenheit value
    c. **step** - This represents the increment you need to make after each value

Now run a loop from **start** to **end**, increasing the value by **step** each time. Each value in the loop represents a Fahrenheit value that you have to convert into Celsius. “**Call” the function you had created in previous question to do the conversion from F to C.**
Print as output the following table, with Fahrenheit values in left column and Celsius values in the right column. e.g. for an input of 0 (start), 100 (end) and 20 (step) the output is:

0 : -17
20 : -6
40 : 4
60 : 15

80 : 26
100 : 37