# Lab 4

1. Write a program to add two integers using a function. Take user input and print out result from main().
2. Write a program to calculate the factorial value of any integer, using a function. Take user input and print out result from main().
3. Write a program to calculate the power of any integer raised to another integer, using a function. Take user input and print out result from main(). Do not use math.h.
4. Check if an integer is prime or not using functions.
5. Check if an integer is an Armstrong number or not using functions.
6. Write a function that receives 5 integers and evaluates the sum, average and standard deviation of these numbers. Call this function from main( ) and print the results in main( ).
7. Write a function that receives the radius of a circle and evaluates the area and the circumference. Call this function from main( ) and print the results in main( ).
8. Write a program to find the greater of two input integers using a function.
9. Write a C program to demonstrate the usage of 4 types of functions depending on the presence and absence of return types and arguments.
    a. Function with void return type and no arguments.
    b. Function with void return type and arguments.
    c. Function with int/float return type and no arguments
    d. Function with int/float return type and arguments.
10. Write a program that continuously prompts a user for an integer value until the user enters 0. Print outputs from main(). The application passes the value in turn to the following functions:

a. A function that displays all whole numbers from 1 up to and including the entered number
b. A function that computes the sum of all the whole numbers from 1 up to and including the entered number
c. A function that computes the product of all the whole numbers from 1 up to and including the entered number – print results from main().

11. Write a C program which will take an integer as input and will contain the following functions:
    a. A function to determine the proper divisors of the input number. The proper divisors of the integer n are the positive divisors of n other than n itself. The proper divisors of 27 are 1, 3 and 9.
    b. Use the previous function to write a function which will determine if the original input number is a perfect number or not. A perfect number is a positive integer that is equal to the sum of its proper divisors. The smallest perfect number is 6, which is the sum of 1, 2, and 3. Other perfect numbers are 28, 496, and 8,128 etc. Print results from main().