IT110 Lab Assignment 4

Objective: by the end of this lab session, you should have reasonable understanding of the following:

I. Loops
II. Functions

Note:

i. Every program that you write should have an accompanying comment specifying what the program does.
ii. Before implementing a program, create a flowchart for that program in your notebook.
iii. Indent your code for better readability.
iv. Give meaningful names to the variables and functions.

Sample program

```c
#include <stdio.h>
void main( )
{
    int a, b, c, sum ;
    int calcsum(int , int , int );

    printf ( "Enter any three numbers " ) ;
    scanf ( "%d %d %d", &a, &b, &c ) ;
    sum = calcsum ( a, b, c ) ;
    printf ( "Sum = %d", sum ) ;
}
int calcsum (int x, int y, int z )
{
    int d ;
    d = x + y + z ;
    return ( d ) ;
}
```
Assignment questions

1. Write a program to display the first n terms of Fibonacci series using a for loop. Fibonacci series is a series of numbers in which each number (Fibonacci number) is the sum of the two preceding numbers. (Fibonacci series → 0 1 1 2 3 5 8 13 21 34. . .).

2. Write a C program to display Pascal's triangle for n rows using a For loop.

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
```

3. Write a program to print the Diamond Star Pattern for any odd. Example for N =9 print the pattern as shown below.

```
* 
***
*****
*******
*********
*******
*****
***
* 
```

4. Write a C program using a function `product()` which receives a float and an int from main(), finds the product of these two and returns the product which is printed through main().
5. Write a function `power(int a, int b)`, to calculate the value of `a` raised to `b` using while loop.

6. Write a C program to take radius of a circle as input from the user and print its diameter, area and circumference using functions: `getDiameter()`, `getArea()` and `getCircum()`. These functions are called from the main() and the output is printed in the main().

7. Write a program to determine whether a given number is prime or not. A number is prime when it is divisible by itself and 1 only. (create a function `isPrime()` that returns 1 when the number is prime and 0 when n is not prime). Print whether the number is prime or not in main().

8. Write a program using function `isPerfect()` to check whether a given number is a perfect number or not. A perfect number is number such that it equals the sum of its factors. (Example: Factors of 6: 1, 2, 3 and 6 = 1 + 2+3.).

9. Write a C function `printArmstrong()` to print all Armstrong numbers between 1 and n. An Armstrong number is one such that it equals the sum of its digits raised to the power of N, where N is the number of digits in the number.
   Example: 153 = 1^3 + 5^3 + 3^3, 1 = 1^1, 371 = 3^3 + 7^3 + 1^3

10. A positive integer is entered through the keyboard. Write a function `getPrimeFact()` to obtain the prime factors of this number. You may define additional functions if required.
    For example, prime factors of 24 are 2, 2, 2 and 3, whereas prime factors of 35 are 5 and 7.

11. Write a C function to check whether a number is palindrome or not by defining a function `isPalindrome()`. Palindrome is a sequence that reads the same backwards as forwards. Example: 121, 345543.

12. Write a C function `getGCD()` to find the greatest common divisor. GCD is the highest factor of the two numbers. Example:
    Factors of 24: 1 2 3 4 6 8 12 24
    Factors of 18: 1 2 3 6 9 18
    GCD of 24 and 18 is 6.
13. Write a C function `getLCM()` to find the lowest common multiple (LCM) of two numbers and print it in `main()`. LCM of two numbers is the smallest number that they both divide. Example:
Multiples of 10: 10 20 30 40 50 60 70 80 90 100
Multiples of 12: 12 24 36 48 60 72 84 96 108 120
LCM of 10 and 12 is 60.

14. Write a general-purpose function `getRoman()` to convert any given year into its roman equivalent and print it. The following table shows the roman equivalents of decimal numbers:

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Roman equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>m</td>
</tr>
<tr>
<td>900</td>
<td>cm</td>
</tr>
<tr>
<td>500</td>
<td>d</td>
</tr>
<tr>
<td>400</td>
<td>cd</td>
</tr>
<tr>
<td>100</td>
<td>c</td>
</tr>
<tr>
<td>90</td>
<td>xc</td>
</tr>
<tr>
<td>50</td>
<td>l</td>
</tr>
<tr>
<td>40</td>
<td>xl</td>
</tr>
<tr>
<td>10</td>
<td>x</td>
</tr>
<tr>
<td>9</td>
<td>ix</td>
</tr>
<tr>
<td>5</td>
<td>v</td>
</tr>
<tr>
<td>4</td>
<td>iv</td>
</tr>
<tr>
<td>1</td>
<td>i</td>
</tr>
</tbody>
</table>

Example:
Roman equivalent of 1988 is mcmlxxxviii
Roman equivalent of 1525 is mdxxv
Roman equivalent of 2004 is mmiv