# ASSIGNMENT_3

**Loops**

(Any ten assignments)

1. Write a C program which prints all integers divisible by n between 1 and 100 where value of n is provided by the user.

2. Write a C program to find out sum of digits of a given number.

3. Write a C program to find the reverse of a given number. **[OPTIONAL]**

4. Write a C program to find out sum of the following series.

   S=1+2+3+4+ ... +n

5. Write a C program to find out sum of the following series.

   S=1.2+2.3+3.4+4.5+ ... +n.(n+1)

6. Write a C program to find out factorial of a given number.

7. C program to find out sum of the following series.

   S=1!+2!+3!+4!+ ... +n!

8. Write a C program to find out sum of the following series.

   S = 1@ + 2@ + 3@ + 4@ + … + n@

   where, n@ is the sum of all factors of n. Example: 6@ = 1+2+3+6 = 12 **[OPTIONAL]**

9. Write a C program which prints all prime numbers between 1 and 100.

10. Write a program which accepts a number n and prints all prime factors of n. **[OPTIONAL]**

11. Write a program to generate all combinations of digit 1, 2 and 3 using for loop. **[OPTIONAL]**

12. Write a program to print the multiplication table of the number entered by the user. The table should get displayed in the following form.

    29 * 1 = 29

    29 * 2 = 58

    ... ... ...

13. Write a menu driven program to accept a number in any Number System [from Binary, Decimal, Octal, and Hex] and convert and display the same in any other amongst these.

14. Write a menu-driven program for displaying log-series results for an input number for different log bases. **[OPTIONAL]**

# ASSIGNMENT_3

15. Write a program named SINE to find the sine of an angle. The angle and its unit (degree, radian or grade) should be provided as command line arguments. For the units, short forms as d/D (for degree), r/R (for radian) or g/G (for grade) may be used. The program should use the series

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots,$$

for evaluation. Take care of negative angles and angles in all the quadrants. **[OPTIONAL]**

16. Write a C program which accepts 100 integers and displays the counts of positives, negatives and zeros entered.

17. Write a C program to print the first n numbers of the Fibonacci sequence. The Fibonacci sequence is constructed by adding the last two numbers of the sequence so far to get the next number in the sequence. The first and second numbers of the sequence are defined as 0 and 1. We get:

    0, 1, 1, 2, 3, 5, 8, 13, 21…

18. Write a program to print out all Armstrong numbers between 1 and 500. If the sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number. For example, 153 = ( $1^3$ ) + ( $5^3$) + ( $3^3$). **[OPTIONAL]**

19. Write a C program which prints the first 10 happy numbers. If you iterate the process (assume maximum 100 iterations) of summing the squares of the decimal digits of a number and if the process terminates in 1, then the original number is called a Happy number. For example 7 is a happy number as 7 → 49 → 97 → 130 → 10 → 1. **[OPTIONAL]**

20. An important property of square numbers: If a natural number is a square number, then it has to be the sum of Successive Odd Numbers starting from 1.

    For example:

| Perfect Square | Sum of Odd Numbers |
|---|---|
| 4 | 1 + 3 |
| 9 | 1 + 3 + 5 |
| 16 | 1 + 3 + 5 + 7 |
| 25 | 1 + 3 + 5 + 7 + 9 |
| 36 | 1 + 3 + 5 + 7 + 9 + 11 |
| 49 | 1 + 3 + 5 + 7 + 9 + 11 + 13 |
| 64 | 1 + 3 + 5 + 7 + 9 + 11 + 13 + 15 |

    Now using this property, find the square root of any perfect square. **[OPTIONAL]**

21. Write a program which reads a positive integer value, and compute the following sequence: if the number is even, halve it; if it's odd, multiply by 3 and add 1. Repeat this process until the value is

# ASSIGNMENT_3

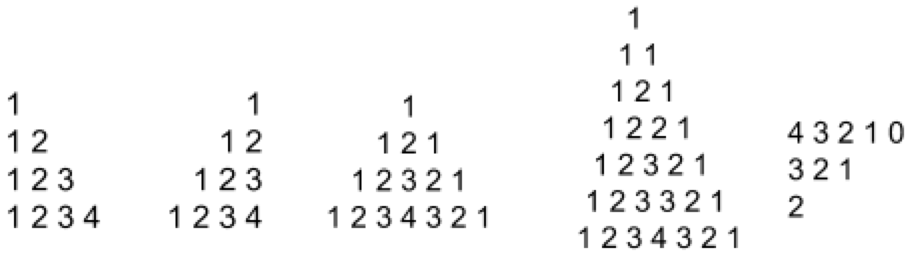
1, printing out each intermediate value. Finally the program should print how many of these operations were performed. Typical output might be:

```
Inital value is    3
Next value is     10
Next value is      5
Next value is     16
Next value is      8
Next value is      4
Next value is      2
Next value is      1
Number of operations is 7
```
**[OPTIONAL]**

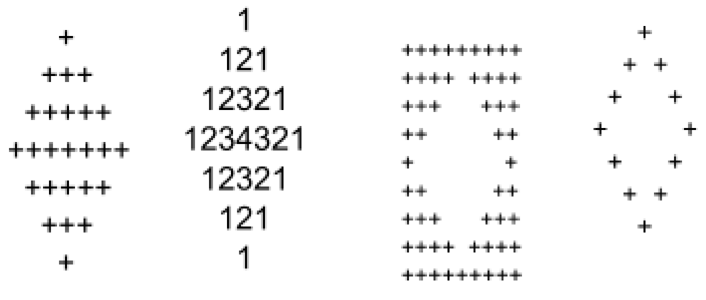
22. Write a program to print all the ASCII values and their equivalent characters using a while loop. The ASCII values vary from 0 to 255.

23. Write a program to find the octal equivalent of the entered integer.

24. Write a C program that prints the following patterns for the input of n=4. The value of n is input by the user.



```
1               1           1                 1
1 2           1 2         1 2 1              1 1
1 2 3       1 2 3       1 2 3 2 1           1 2 1
1 2 3 4   1 2 3 4     1 2 3 4 3 2 1        1 2 2 1        4 3 2 1 0
                                          1 2 3 2 1       3 2 1
                                         1 2 3 3 2 1      2
                                        1 2 3 4 3 2 1
```



```
   +           1        +++++++++       +
  +++         121       ++++ ++++      + +
 +++++       12321      +++   +++     +   +
+++++++     1234321     ++     ++    +     +
 +++++       12321      +       +     +   +
  +++         121       ++     ++      + +
   +           1        +++   +++       +
                        ++++ ++++
                        +++++++++
```

**[OPTIONAL]**

25. Write a program to add first seven terms of the following series using a for loop:

1/1!+2/2!+3/3!+··· **[OPTIONAL]**