List of Programmes which you need to solve using Pen and Paper only

1. **Given a parentheses string, return the minimum number of parentheses we must add to make the resulting string valid.**

   Ex 1**.** Input :- "()))(("

   Output :- 4

   Explanation : Here we need to add 2 parentheses at the beginning and 2 at the end so that our resultant string would be "**((**()))((**))**"

   Ex 2. Input :- "(())"

   Output :- 0

   - Here It is complete parentheses so that output will be 0.

2. **You are given a float variable in which a real number is stored. You need to find the numerator and denominator of That real number. The maximum number of digits after the decimal point in this real number will be 5. You can not use any type of inbuilt functions in this.**

   Example:

   Input - 0.4

   Output - Numerator: 2

   Denominator: 5

   NOTE: Our output should be in the most simplified form. E.g. For the above example we can not provide output 4 and 10. 2 and 5 will be the only unique answer for the above example.

3. **Finding First N prime numbers.**

4. **Find the first element of one array that is not in another Write a function that has two arguments: two integer arrays. The arrays may be of different sizes. The function returns the index of the first value in the first array that is not somewhere in the second array. If all values in the first array are also in the second array, return -1.**

   Input:

   Array A:

   2 2 3 3 7 1 29 19 2 19

   5

   Array B:

   29 19 5 3 7 2

   Value 1 is in A but not in B: returns 6

   Array A:

   2 2 3 3 7 5 29 19 2 19

   5

   Array B:

   29 19 5 3 7 2

   All elements in A are found in B: returns -1

5. **Check that the elements of an array are in ascending order write function that checks that the elements of an array are in ascending order. Return 0 if the array is in order; otherwise, return the number of elements that are out of order. Here are some sample runs:**

   Inputs:

   1 2 3 4 6 7 8 7 10 11 13 20 31 41 41 45 50 53 52 52

   3 elements out of order. Returns 3

   1 1 1 4 6 7 8 8 10 11 13 20 31 41 41 45 50 51 52 52

   All elements are in order. Returns 0