Exercise E

Section E Examples that require loop construct with counters
– use *for* loop construct for the following:

1. Given a number find out its factorial.

   Write two different C# program variations for the problem:
   a. Using increment counter
   b. Using a decrement counter.

   Carefully study the similarities and differences between the two approaches.

2. Write a program to print all numbers between 1 and 10 with the values of its inverse, square root and square as below:

NO	INVERSE	SQUARE ROOT	SQUARE
1.0	1.0	1.0	1.0
2.0	0.5	1.414	4.0
3.0	0.333	1.732	9.0
4.0	0.25	2.0	16.0
5.0	0.2	2.236	25.0
6.0	0.167	2.449	36.0
7.0	0.143	2.646	49.0
8.0	0.125	2.828	64.0
9.0	0.111	3.0	81.0
10.0	0.1	3.162	100.0

3. Given an integer as input determine whether the number is a prime number or not. Your program should output “Prime” or “Not Prime” as the case may be.

   A Prime Number is one which is only divisible by one and itself.

   Consider how the efficiency of the program can be improved. Normally the order of complexity is proportional to the number of times a loop is executed.

4. Given an integer as input write a C# program to determine whether the number is a Perfect Number or not.

   A perfect number is one for which the sum of its factors (including number one) add up to the number itself. For example number *six* is a perfect number because,
   6 = 1 + 2 + 3.

5. Modify the Prime Number C# program to print out all the prime numbers from 5 to 10000.

6. Modify the Perfect Number C# program to print out all the perfect numbers from 1 to 1000.