**Exercise 1**

Write a console application that incorporates a recursive function called

**power (base, exponent)** that when invoked returns base exponent.

For example**, power(3,4) = 3 \* 3 \* 3 \* 3**. Assume that the **exponent** is an integer greater than or equal to 1.

### Hint: The recursion step would use the relationship

***base exponent = base \* base exponent-1***

*and the terminating condition occurs when the exponent is equal to 1 because*

***base1 = base***

Incorporate this method into the application that enables the user to enter the **base** and the **exponent** and calculates the answer.

Test your application with the following data:

* power(3,4)=81
* power(2,5)=32
* power(4,2)=16

**Exercise 2**

The greatest common divisor (GCD) (*also known as the highest common factor*) of two integers m and n is the largest integer that evenly divides into each of the two numbers e.g.

* Factors of a number divide into the original number evenly e.g. the factors of 16 are 1,2,4,6 and 16.
* If a factor of a number is a factor of another number, then it is said to be a common factor e.g. 2 is a factor of 4 and 8 so it is a common factor.
* The factors of 8 are 1,2,4 and 8.
* The gcd(8,16) = 8

The GCD of 15 and 10 is 5.

The GCD of 12 and 8 is 4.

Write a method gcd that returns the greatest common divisor of m and n. The gcd of m and n is defined recursively as follows:

m if m = n

gcd(m-n, n) if m > n

gcd(m,n-m) otherwise

**{**

Incorporate the method into an application that reads two values from the user. Display the result of the method in the text area.

Test your application with the following data.

* gcd(100,70)=10 gcd(8,16)=8 gcd(49,14)=7

**Exercise 3**

Using the following algorithm to write a program to reverse a number using recursion.

1. Create a function called **reverseNumbers** to accept an integer value as input.
2. If the number is less than 10 then print that number
3. Otherwise:
   * 1. Extract the last digit of the number using the **modulus** (**%**) operator and print that value.
     2. Call the **reverseNumbers** function recursively but change the parameter by number/10.