**Lab Report1**

Problem 1

1. Write a recursive implementation of Factorial.
2. Write an iterative (i.e. use any of the looping constructs (for, do-while, while) to accomplish the task) implementation of Factorial.

Problem 2

1. Write a recursive implementation of Fibonacci.
2. Write an iterative implementation of Fibonacci.

Problem 3

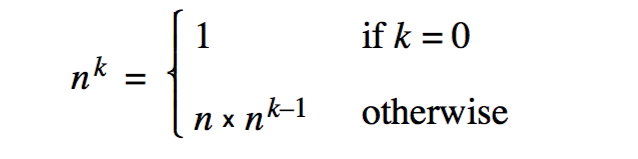
The **greatest common divisor** (g.c.d.) of two nonnegative integers is the largest integer that divides evenly into both. In the third century B.C., the Greek mathematician Euclid discovered that the greatest common divisor of *x* and *y* can always be computed as follows:

* If *x* is evenly divisible by *y,* then *y* is the greatest common divisor.
* Otherwise, the greatest common divisor of *x* and *y* is always equal to the greatest  common divisor of *y* and the remainder of *x* divided by *y*. Use Euclid’s insight to write a recursive function **GCD(x,y)** that computes the  greatest common divisor of *x* and *y*.

Problem 4

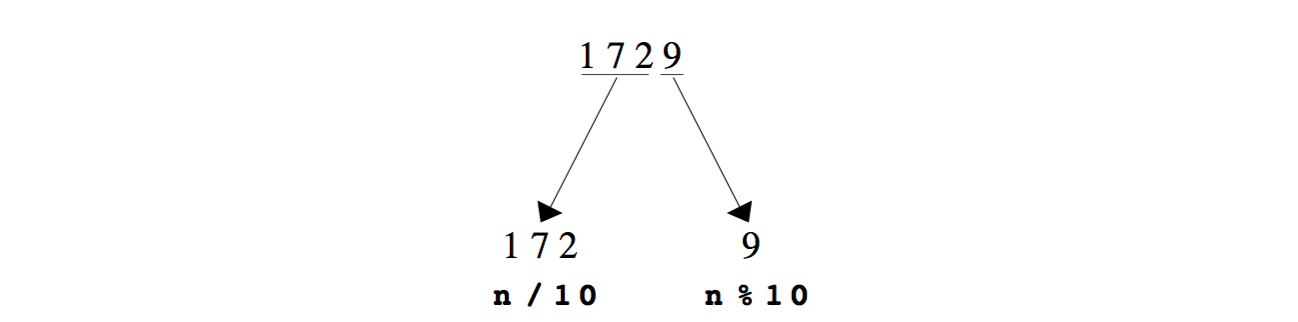
Unlike many programming languages, C++ does not include a predefined operator that raises a number to a power. As a partial remedy for this deficiency, write a recursive implementation of a function   **int RaiseIntToPower(int n, int k)**

that calculates *nk*. The recursive insight that you need to solve this problem is the mathematical property that



**Problem 5**

Write a recursive function **DigitSum(n)** that takes a nonnegative integer and returns the sum of its digits. For example, calling **DigitSum(1729)** should return 1 + 7 + 2 + 9, which is 19.  The recursive implementation of **DigitSum** depends on the fact that it is very easy to break an integer down into two components using division by 10. For example, given the integer 1729, you can divide it into two pieces as follows:



Each of the resulting integers is strictly smaller than the original and thus represents a simpler case.

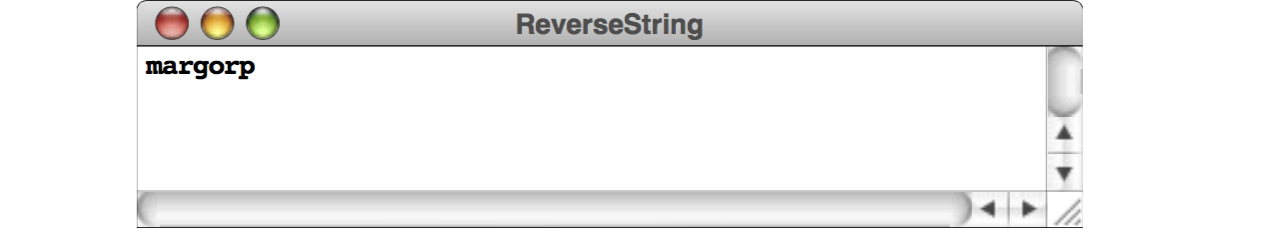
**Problem 6**

Write a recursive function that takes a string as argument and returns the reverse of that string. The prototype for this function should be  

**string Reverse(string str);**

and the statement  

**cout << Reverse("program") << endl;**

should display 

Your solution should be entirely recursive and should not use any iterative constructs such as **while** or **for**.