**Higher powers of two**

It is a straightforward task to write a function that will create powers of two, *without using Math library functions* (otherwise you could just use Math.Pow !). You could use a loop e.g.:

public static int TwoToThePowerOf(int power)

{

int number = 1;

for (int i = 0; i < power; i++)

{

number = number \* 2;

}

return number;

}

However, this will work only up to 230 - because a standard int in C# is 32 bits (and that has to represent both positive and negative numbers). You could handle higher powers by switching it to work with long - which has 64 bits, but this will still work only up to 262. How could we calculate much higher powers of 2 e.g. 2100 or 21000 ?

One way would be to handle the numbers as strings – which are not limited in length as follows:

public static string TwoToThePowerOf(int power)

{

string number = "1";

for (int i = 0; i < power; i++)

{

number = MultiplyByTwo(number);

}

return number;

}

public static string MultiplyByTwo(string digits)

{

}

Your task is to implement the MultiplyByTwo function, to work on a positive decimal number *of any length*.

*Start* by copying the code above and then writing unit tests both for your MultipluByTwo and for the final TwoToThePowerOf functions:

[TestMethod]

public void TestMultiplyByTwo()

{

Assert.AreEqual("0", Program.MultiplyByTwo("0"));

Assert.AreEqual("2", Program.MultiplyByTwo("1"));

Assert.AreEqual("12", Program.MultiplyByTwo("6"));

Assert.AreEqual("198", Program.MultiplyByTwo("99"));

Assert.AreEqual("202", Program.MultiplyByTwo("101"));

}

[TestMethod]

public void TestTwoToThePowerOf()

{

Assert.AreEqual("1", Program.TwoToThePowerOf(0));

Assert.AreEqual("2", Program.TwoToThePowerOf(1));

Assert.AreEqual("4", Program.TwoToThePowerOf(2));

Assert.AreEqual("1024", Program.TwoToThePowerOf(10));

Assert.AreEqual("65536", Program.TwoToThePowerOf(16));

Assert.AreEqual("1267650600228229401496703205376", Program.TwoToThePowerOfMk(100));

Assert.AreEqual("10715086071862673209484250490600018105614048117055336074437503883703510511249361224931983788156958581275946729175531468251871452856923140435984577574698574803934567774824230985421074605062371141877954182153046474983581941267398767559165543946077062914571196477686542167660429831652624386837205668069376", Program.TwoToThePowerOfMk(1000));

}

Some hints:

1. You will need to process the string, one digit at a time, starting from the least significant digit
2. You can convert the char digit into an int for the purposes of doubling it
3. You will need to handle the carry correctly between digits. *Do a multiplication neatly on paper, step-by-step, to ensure you understand how the carry works.*
4. Build up your answer into a new string using StringBuilder. Using .Append will add to the end of the string, but there is also a mechanism for adding to the front of a string in StringBuilder - look it up.