Lab09: String Conversions
CSE1010 Fall 2012
Jeffrey A. Meunier
University of Connecticut

Part 1

This is one way to convert a string into a vector of digits.

```
>> n = 123;
>> s = num2str(n) % convert n to a string
s =
123
>> V = double(s) % convert chars to character codes
V =
  49  50  51
>> V = V - '0' % convert character codes to digits 0-9
ans =
  1  2  3
```

Type each of these statements to see how they work. Try them with different numbers. Can you combine these statements into one statement that does the same thing?

Part 2

Turn the conversion statement (or statements) from part 1 into a function called **num2vec**.

Examples of using the function:

```
>> num2vec(123)
ans =
  1  2  3
>> num2vec(98765)
ans =
  9  8  7  6  5
```

```
>> length(ans)
ans =
  5
```

Part 2a

Write a second **num2vec** function (call it **num2vec2**) that does the same thing that **num2vec** does, but it uses a loop to create the vector. In this function you are not allowed to use strings, nor are you allowed to use functions that use strings (**num2str** or **str2num**).

Hint: The **rem** function would be really useful here.

Part 3a

Write a **vec2num** function that does the opposite of the **num2vec** function: it takes a vector of digits and converts it into a number. *Hint*: This function must do the opposite operations as the **num2vec** function, and it must do them in the opposite order.

```
>> n = vec2num([9 8 7 6 5])
n =
  98765
>> class(n)
ans =
double
>> size(n)
ans =
  1  1
```

Part 3b

Write a second **vec2num** function (call it **vec2num2**) that does the same thing that **vec2num** does, but it uses a loop to create the number. In this function you are not allowed to use strings, nor are you allowed to use functions that use strings (**num2str** or **str2num**).