CS30 Spring 2015
Lab 7

Use the command `diary` to record your answers and submit them. Submit code for the functions you write.

1. (30 points) Sum all of the digits of a nonnegative integer. Write a function `DigitSum` that takes as input a nonnega tive integer, and returns as output the sum of all of its digits. If the input is not a nonnegative integer, an error message should be printed and the function should return 0. (Note: call `return` to exit early.) Your function should produce the output shown on the following test cases:

```
>> DigitSum(-2)
Error: Input must be a nonnegative integer
ans =
     0
>> DigitSum(1.2123)
Error: Input must be a nonnegative integer
ans =
     0
>> DigitSum(123456789)
Sum of integer part is 45
ans =
    45
>> DigitSum(9999999990)
Sum of integer part is 81
ans =
    81
>> DigitSum(intmax)
Sum of integer part is 46
ans =
          46
```

2. (40 points) Gauss-Seidel Algorithm. The Gauss-Seidel algorithm is an iterative method for solving some systems of linear equations. In particular, it is known to work for diagonally dominant matrices (matrices where in each row, the absolute value of the diagonal is larger than the sum of the absolute values of the off-diagonals). In Gauss-Seidel, the matrix `A` is decomposed as

$$A = L + U$$

where L is the lower triangular part of A (all the elements at or below the diagonal of A), and U is the strictly upper triangular part of A (all the elements above the diagonal of A). E.g., for

$$A = \begin{pmatrix} 3 & 1 & 1 \\ 2 & 4 & -1 \\ 1 & -2 & 5 \end{pmatrix},$$

we have

$$L = \begin{pmatrix} 3 & 0 & 0 \\ 2 & 4 & 0 \\ 1 & -2 & 5 \end{pmatrix}, \qquad U = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix}.$$

The Gauss-Seidel iteration then proceeds by solving the lower triangular system as

while (not converged)
  solve $L\ x^{k+1} \ = \ -U\ x^k \ + \ b$

As a convergence criterion, we will check the norm of the residual vector, r, defined as

$$r = b - Ax.$$

Note that when the exact solution x is found, the residual is 0. In practice, we only require that the residual norm fall below a given threshold. Implement the Gauss-Seidel algorithm in a function called GaussSeidel, which takes as input the matrix, A, and the right-hand-side vector, b. Use a vector of all 0's as your initial guess. Iterate until the residual norm is less than 10^{-5}. You can use Matlab's \ operator to do the lower triangular solve. The following steps will help you get started:

(a) Construct `lowerA`, the lower triangular portion of `A`.
(b) Construct `upperA`, the strictly upper triangular portion of `A`.
(c) Initialized `x` to the initial guess of all zeros.
(d) Compute the initial residual vector.
(e) Write a loop to do the Gauss-Seidel iteration, checking the convergence criterion, terminating when it is met.
(f) Your function should produce the output shown on the following test cases:

```
>> x = GaussSeidel([ 5, 2, 1; 2, 10, 3; -1, -2, 7], ...
[ 1, 2, 3]' )
x =
   0.090908719759095
   0.045454483938442
   0.454545383947997
>> x = GaussSeidel([ 5, 2; 2, 10], [ 1, 2]' )
x =
   0.130435010560000
```

```
   0.173912997888000
>> rng(12345)  % set the seed for the random number generator
>> A = rand(10,10); % generate a random 10x10 matrix
>> A = A + 15*eye(10); % ensure that it is diagonally dominant
>> b = rand(10,1); % generate a random 10x1 vector
>> GaussSeidel(A,b) % solve
ans =
   0.002309370438524
   0.011793880505613
   0.034437814071005
  -0.002998180044434
   0.004243958275690
   0.026481507559552
   0.038035065571281
   0.047562791281957
   0.031682939367880
   0.030940798355999
```

3. (30 points) Write a function `Factorial` which takes as input a positive integer `n` and returns as ouput `n!`. Your function should check that the input is a positive integer and print an error message if it is not. Your function should produce the output shown on the following test cases:

```
>> Factorial(-1)
Error: input must be a nonnegative integer.
ans =
     1
>> Factorial(10.1231)
Error: input must be a nonnegative integer.
ans =
     1
>> Factorial(0)
ans =
     1
>> Factorial(1)
ans =
     1
>> Factorial(4)
ans =
    24
>> Factorial(6)
ans =
   720
>> Factorial(10)
ans =
     3628800
```