CS30 Spring 2014
Lab 2

General instructions on what to submit: Submit a separate sheet of paper with the answers to the questions given below. Submit copies of scripts and functions as indicated. Use the Matlab command `diary` to record your interactive Matlab session. Submit this diary to show the results of the commandline tasks given below.

1. (30 points) How many times do you need to fold a piece of paper to reach the moon? Assume that the distance to the moon is approximately 238,855 miles, and that the thickness of the piece of paper is approximately .1 mm.
   (a) Write a script called `NumberOfFolds.m` that computes the number of times you need to fold the piece of paper in half in order for its thickness to reach the moon. The number of folds should be an integer (use the Matlab function `ceil`). Your script should define the variables `distanceToMoonMiles`, `thicknessOfPaperMm`, and `numberOfFolds`. You can also define other variables useful for the calculation. Submit a copy of your script.
   (b) How many folds are needed for the piece of paper to reach the moon?
   (c) The closest distance of Earth to Mars is approximately 33.9 million miles. Modify your script to compute the number of folds needed to reach Mars. Use scientific notation to define the distance to Mars. Submit a copy of your script.
   (d) How many folds are needed for the piece of paper to reach Mars?
   (e) Is the variable `distanceToMoonMiles` visible from the command line? Use the Matlab function `exist` to check.

2. (35 points) Sum the six least significant digits of a positive integer.
   (a) On the command line, use the Matlab function `mod` to find the least significant digit of a positive integer.
   (b) Given a positive integer, `number`, write a function `DigitSum6(number)` that computes the sum of the six least significant digits of `number`. If number has less than six digits, it computes the sum of those digits. Your function should define variables `first`, `second`, `third`, `fourth`, `fifth` and `sixth`, holding the least significant digit, next least significant digit, etc. Running `help DigitSum6` should give the following output:

      ```
      Compute the sum of the six least significant digits of the input.
        Input is a positive integer.
      ```

      Submit a copy of your function.
   (c) Run your function on the following test cases:

```
>> DigitSum(123456)
>> DigitSum(123123123456)
>> DigitSum(456)
>> DigitSum(4)
>> DigitSum(DigitSum(999999))
```

(d) Is the variable `first` defined in your function visible from the command line?

3. (35 points) Floating point numbers.

(a) What are the smallest possible single and double precision floating pointer numbers?

(b) What are the largest possible single and double precision floating pointer numbers?

(c) Define a single precision floating point number as

```
>> myMaxNumber = realmax('single')
```

and

```
>> myEps = eps(myMaxNumber)
```

What happens if you add `1` to `myMaxNumber`? What happens if you add `myEps/3` to `myMaxNumber`? What happens if you add `myEps/2` to `myMaxNumber`? Explain this behavior.

(d) Define a single precision floating point number as

```
>> myNumber = single(12345678)
```

What larger single precision floating point number is closest to `myNumber`?

(e) What is the value of the following Matlab expression? What does it mean?

```
>> Inf - Inf
```

(f) Define a single precision floating point number as

```
>> myNumber = single(12345678)
```

What are the results of the following Matlab expression?

```
>> result1 = myNumber + .3 + .3
>> result2 = myNumber + (.3 + .3)
```

Why are `result1` and `result2` different?

(g) What is the absolute error in computing result1? What is the relative error?