Exercises for Day 22

Exercise 1. Create a factorial function `fact.m`, as in the lecture slides. Create a main script that calls this function. The main script should print a table (to a text file) of the factorials of the odd integers from 1 to 19. The beginning of the table should look like the following:

```
i              i factorial
1                        1
3                        6
5                      120
```

Turn in your function file, the main script, and the text file.

Exercise 2. Start with your code from Day 6 Exercise 2. This is the code where we set some variable, say N, to be the number of rectangles to use to numerically integrate an acceleration function.

As we discussed in class, take the numerical integration portion out of the code and place it in a separate m-file called `IntegrateAcceleration.m`. The first line of this m-file should be in the following form (your variable names may be different):

```
function [Vf] = IntegrateAcceleration(Nrect,ts,te,Vi)
```

Modify your original program so the output is the same as before except that now it uses the `IntegrateAcceleration` function. Run it for 100 rectangles to be sure it still gives you a final velocity of 3280.1 m/s. Turn in your modified program, the `IntegrateAcceleration.m` function, and the text file output showing that it works the same as the original.

Exercise 3. Now make a new version of your program in Exercise 2 that uses `IntegrateAcceleration.m` to make a plot of the final velocity versus N (the number of rectangles). Make a plot of this for values of N large enough to show that the final velocity asymptotically approaches its exact value of 3458.5 m/s. Turn in this new version of your program and a printout of the plot.

> **Note:** You probably will want to make large steps in the number of rectangles because you're going up to such a high number. Consider something like `for N=100:100:100000`.
>
> **Also:** When you plot a very large range of values like this, it is often much more useful to use a log scale on your plot. Instead of `plot(x,y)`, use `semilogx(x,y)` for this problem — that will use a log scale for your x-axis.