Name : ______________________________

Homework Project #1

Math 365, Spring 2015

Due Wednesday, February 4th

Please following the guidelines below when asking for help from others, especially those in the class.

- If you are asking for help from a classmate, you may show them your code.
- If you are helping another classmate, you may *not* show them your solutions.
- Under no circumstances should you share a copy (electronic or any other easily accessible form) of your code with any other students in the class.
- Keep in mind the spirit of this class. While I encourage collaboration, everyone is responsible for learning the material.
- And most important, start early and have fun!

For this homework, you are asked to write several files to disk. To do this, you will use the function `write_file` that you downloaded from the course website for homework #0. Recall, that to save an array `v` to the file `mysoln.out`, use the command

```
write_file(v,'mysoln.out')
```

1. (**Generating random numbers**) Using the Matlab `rand` command, generate an array of $N = 10^5$ random numbers x_i, $i = 1, 2, \ldots, N$ in the interval $[0, 1]$. Then, use the cumulative sum function `cumsum` to construct an array containing the "partial averages" $s_1, s_2, \ldots, s_N$ defined as

$$s_n = \frac{1}{n}\sum_{i=1}^{n} x_i$$

   Plot s_n as a function of n and show that as n gets large, the average value of the first n random numbers in your array approaches $1/2$. To show this, add to your plot the line $y = 1/2$. Your plot should look like that shown in Figure 1.

   - Be sure to add a title and axis labels to your plot.

   For this problem, you will need the functions `rand`, `cumsum`, `plot` and the colon (:) operator.

2. (**Geometric series**) In Calculus II, you learned that the sum of the geometric series $1, a, a^2, a^3, a^4, \ldots$ converges as long as $|a| < 1$. The value of the infinite sum can be easily shown to be

$$S = \sum_{i=0}^{\infty} a^i = \frac{1}{1-a}, \qquad |a| < 1$$

   For large values of k, we expect that the partial sums

$$S_k = \sum_{i=0}^{k} a^i, \qquad k = 0, 1, 2, \ldots, N$$

   approach this exact value $1/(1-a)$ for large values of N. To test this, construct the series a_i and the series of partial sums S_k for $i = 0, 1, 2, \ldots, N$. (Note that Matlab does not allow you start with an index of 0, so you will have to shift your indexing slightly to get things to work.)

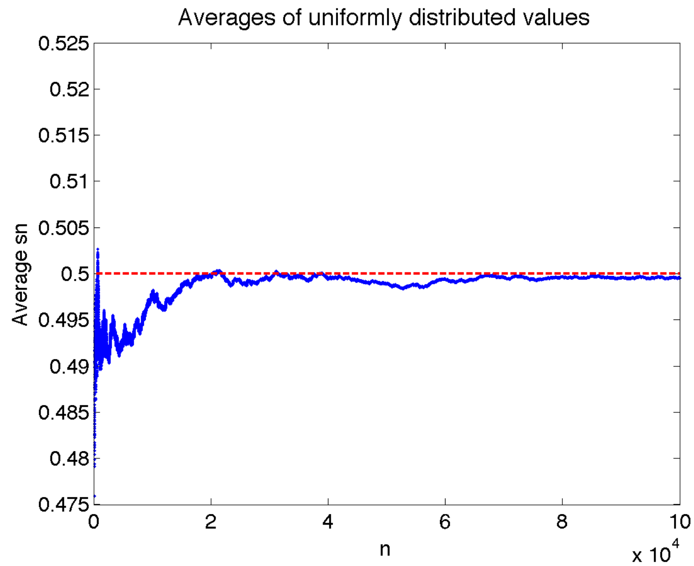

Figure 1: Problem 1 : Plot of cumulative averages of a randomly distributed variable.

- Using the value $a = 0.23$, convince yourself that for large values of k, that S_k approaches $1/(1-a)$. Plot the partial sums on a graph and show the convergence to the true value. Note that you do not need very large values of N to get almost exactly $1/(1-a)$. Store the value of S_N you get for your choice of N in a file `S.out`.
- **Grandi's Series** What happens if we let $a = -1$? Our series becomes $1, -1, 1, -1, 1, -1, \ldots$ and the partial sums S_k are $1, 0, 1, 0, 1, \ldots$. The partial sums clearly do not converge, yet at infinity, our formula for S suggests that the sum should be $1/2$. In fact, we can actually compute a "sum" of this divergent series by looking at the average value of the partial sums. To do this, construct the terms P_n in the series

$$P_n = \frac{1}{n}\sum_{k=0}^{n} s_k.$$

  Show that the P_n converge to $1/2$ when $a = -1$. To show this convergence, you can use a for-loop and the `fprintf` command to print the last five values of your series $P(n)$. For example,

```
for n = N-5:N,
    fprintf('P(%7d) = %24.16f\n',n,P(n));
end
```

  You will need fairly large values of N ($N \approx 1e6$) to see decent convergence.

  To learn more about this series, search for the term "Grandi's Series" in Wikipedia.

  If the Grandi's Series does not seem weird enough for you, search for the expression "1+2+3+4+5", also in Wikipedia. You will see that using some very sophisticated notions of "summation", the series $1 + 2 + 3 + 4 + \ldots = -1/12$.

For this problem, you will need the functions `cumsum`, `fprintf`, `plot` and the colon (:) operator. You should only need a `for` loop for printing out values of Grandi's Series.

3. (**Plotting tangent lines**) Given the two functions $f(x)$ and $g(x)$ below, you are going to construct a composite function $h(x)$ and its derivative.

$$f(x) = \sin(4x) + 2, \qquad g(x) = \cos(e^x), \qquad h(x) = f(x)^{g(x)} - 1.25$$

You will be evaluating this function over the domain $[-1, 2.35]$. Create Matlab *anonymous functions* $f(x)$, $g(x)$ and $f'(x)$ and $g'(x)$. Use these to construct anonymous functions for $h(x)$ and $h'(x)$. **Note :** Use the *logarithmic* derivative to compute the derivative $h'(x)$ in terms of $f(x)$, $g(x)$, $f'(x)$ and $g'(x)$.

(a) Evaluate the function $h(x)$ and its derivative $h'(x)$ at the points $x = -0.98$, $x = 0.45$ and $x = 1.74$ and write the values you obtain to a file `h.out`. Your file should contain two columns of three rows each. The first column is for the value $h(x)$ and the second is for $h'(x)$. You can do this with the following code :

```
x = [-0.98 0.45 1.74];
% ..... define your anonymous functions h(x) and hp(x)
write_file([h(x(:)) hp(x(:))],'h.out');
```

(b) Using the functions you created above, construct a function $T(x; a)$ for the *tangent* line to the curve $h(x)$ at the point $a = 1.2$. Evaluate your tangent line at the three points given in part 3a and write these values to a file `htangent.out`.

(c) Create a plot of the function $h(t)$ along with a plot of the tangent line at $x = a = 1.2$. Add a symbol to your plot clearly showing the point of tangency, a title and axis labels.

For this problem, you will need the Matlab functions `linspace` and plot, as well as anonymous function handles.

4. (**Area of a polygon**) We can describe a polygon by its vertices

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\},$$

where we assume that the vertices are listed in counter-clockwise order as we traverse the polygon. From this list, we can exactly compute the area of the polygon using the formula

$$A = \left| \sum_{i=1}^{N} \frac{(y_{i+1} - y_i)(x_{i+1} + x_i)}{2} \right| \tag{1}$$

We assume that the vertices "wrap" so that $x_{N+1} = x_1$ and $y_{N+1} = y_1$. In each of the following problems, you are asked to use the above area formula to compute the area of the requested shape. The area you get will always be the exact area of the polygon, but may only be an approximate area to the requested shape.

In the following problems, you should

- Plot the requested shape by plotting the vertices and lines connecting the vertices. See Figure 2 for a sample plot.
- Compute the area using the area formula (1).
- Store the value of the area you compute in the indicated file.

Since you will be using area formula (1) several times, you should create an *anonymous* function to implement the formula. Also, be aware that you will need to wrap the endpoints of your vertices to get the correct area value.

(a) To check that you have implemented the area formula (1) correctly, use it to compute the area of the rectangle with vertices $(0, 0)$, $(5, 0)$, $(5, 3)$ and $(0, 3)$. You should get exactly 15. Store the value you get in the file `rect.out`.

(b) Compute the exact area of a parallelogram created by the intersection of the four lines

$$y = 2x + 5 \qquad y = -x + 1$$
$$y = 2x - 2 \qquad y = -x - 6$$

See Figure 2. You can use Matlab to compute the vertices using the following hints.

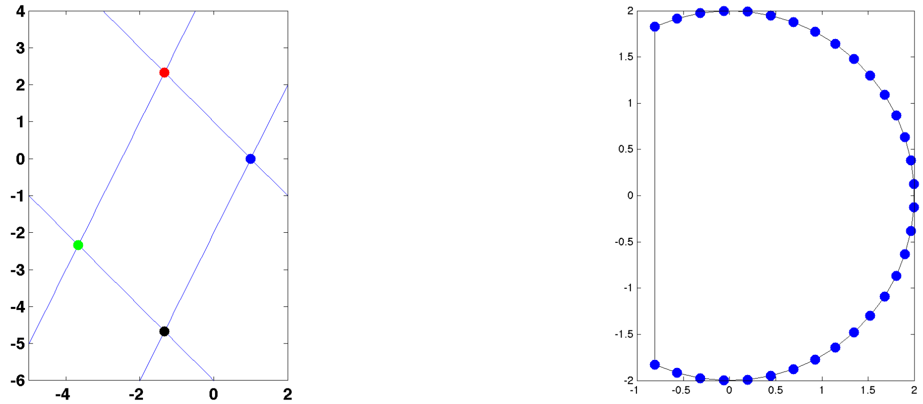

Figure 2: Plot of parallelogram in Problem 4b and sliced circle in Problem 4d.

- Create an anonymous function that evaluates a line in $y = mx + b$ form. Plot the four lines above to get an idea as to where they intersect. To do this, you might find it useful to create an array of slopes $[m_1, m_2, m_3, m_4]$ and an array of y-intercept values $[b_1, b_2, b_3, b_4]$. Then use a for-loop to plot each line.
- Use Matlab to compute the four corners of the parallelogram. For example, you might create an anonymous function that finds the x value of the intersection of two lines in $y = mx + b$ form, given (m_1, m_2) and (b_1, b_2). Then use this function to compute the four x-intersections that you need.
- Compute the y values associated with each x intersection value you found above.

Compute the area of the resulting parallelogram, and store the value in the file `pgram.out`.

(c) Approximate the area of a circle of radius 2 centered at the origin. Construct the vertices using the parameterization $x(\theta) = R\cos(\theta)$ and $y(\theta) = R\sin(\theta)$, where R is the radius of the circle. Use enough points so that you get four digits of accuracy in your area approximation (i. e. four digits after the decimal place are correct). Store the value you compute in the file `circle.out`.

(d) Compute the area described by the intersection of a circle of radius 2, centered at the origin and the half plane $x \geq -0.81$. Compare your solution to the exact solution (which you can easily compute by hand), and use enough points to get four digits of accuracy. Produce a plot of your sliced circle, and write your results to the file `slicedcircle.out`.

(e) (**Extra Credit - 10 pts**) Compute the area of the red figure in Figure 3 using the area formula. Compare your calculated value using the area formula to an exact value (which you can easily compute). Store the 12 vertices you found in a 12×2 array containing an (x, y) pair for each point. Write this array to the file `vertices.out`. In a second file named `cross.out` write out both your computed value and the true area (that you compute analytically).

For this problem, you will need the Matlab functions `linspace`, `sum` and `plot`, as well as anonymous function handles. To compute vertices, you will also need a few trigonometric functions.

5. (**Area of the Rim Fire**) The California Rim Fire near Yosemite Valley, burned for several weeks in late summer, 2013, and was one of the largest fires in California history. Using data from the web, we can estimate how large an area was damaged by this fire. From the course website, download a file called `rimfire.dat`. This contains latitude-longitude coordinates for the outline of the fire on September 2, 2013. Your goal for this problem is to obtain an approximation for the area of the fire, in square miles, square kilometers, and acres.

   To load the file `rimfire.dat`, use the Matlab `load` command :

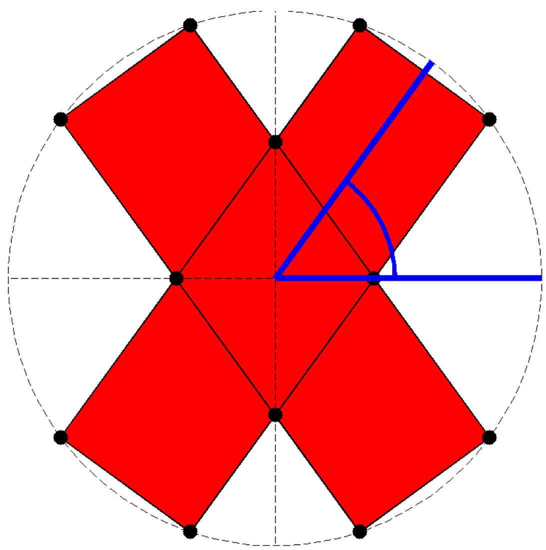

Figure 3: Extra Credit : Compute the area of the red cross. The angle indicated in blue is 54°, the radius of the circumscribing circle is 2, and the width of each of the two arms forming the cross is 1.2.

```
d = load('rimfire.dat');
x = d(:,1);     % First column of d
y = d(:,2);     % Second column of d
```

For this homework, you will do the following :

(a) Plot the outline of Rim Fire and add a title and axes labels.

(b) Estimate the total area burned by the fire in (1) acres, (2) square miles and (3) square kilometers. You may assume that over the region covered by the fire that the earth is flat (so you do not need to compute the area of a "spherical polygon"). Explain how you convert from "square degrees" to the area units requested above. Write the three area approximations you get to a file `firearea.out`.

(c) Compare your answer to what you can find from official reports on the web. In your final document, include links to any websites you use.