1. A fine arts major is enrolled for a class that meets in the Engineering building. Immediately after the first class, the student leaves the classroom and panics. At the first intersection of hallways (which are all aligned in the N, S, E or W directions) the student turns in a random direction and runs to the next intersection. Again they randomly select a direction and run to the next intersection. At some point the student emerges from the building. If the student emerges on the W side of the Engineering building, it is a success because they can clearly see their way back to the more familiar W side of campus. If they emerge on the N, S, or E side of the building, it is a failure.

   Tradition dictates that the floor plan of any Engineering building be approximated as a square, containing, for example, three interior N-S hallways and three interior E-W hallways, for a total of nine interior hallway intersections. If the student starts their escape from the Engineering building at each of the nine interior intersections, determine the probability of emerging on the W side of the building. (This should be a number between 0 and 1.)

   Run your simulation about 10,000 times for each interior intersection. Report your results on a (hand drawn) 5-by-5 grid where the probability is 0 for all the N, S, and E exterior exits, and the probability is 1 for all the W exits. (A "5-by-5 grid" means "five fence posts" and "four fence rails".)

   When you write your program, refer to the intersections using a cartesian coordinate scheme (i, j) where i and j refer to the i-th N-S hallway, and the j-th E-W hallway, respectively. For example, the southwest corner of the building is (0,0) and the northeast corner is (5,5). Also, write your program so that it can *easily* be extended to a building with more hallways in either the E-W or N-S directions.

   Your program should ask the user to input the coordinates of the intersection and print out the probability of success for that particular intersection.

2. Assume a guy enters a bar, walks straight in a distance d, and encounters a woman. After his best attempt at conversation, he is rebuffed and moves off in a random direction a distance d where he encounters another woman. Again he is rebuffed, moves another distance d in a random direction, and so on. If he is rebuffed 10 times, he will be completely demoralized and remove himself from the scene. Also, if he "leaves" by passing back through the front wall, he is out of the game.

   You are sitting opposite the entrance, at the far end of the bar. What fraction of guys actually make it to your end of the bar if the bar is $3d$ deep? How about $5d$ deep? How about $10d$ deep? Continue this investigation until you can determine the nature of the relationship between the depth of the bar and the fraction of guys that can make it all the way across the bar to the far side. Make a hand drawn, or MatLab, plot of your results (fraction versus bar depth in multiples of d).

   Your program should ask the user how deep the bar is in multiples of d. For example, if the bar is 3d deep, the user should input 3. The program should print out the percentage of successes (number of guys that make to the back of the bar). In addition, your program should only allow the guy entering the bar to move in directions corresponding to degree increments. In other words, $0°$, $1°$, $2°$, and so on.

3. Recently in class we considered code to calculate the area inside a region in the first quadrant of the xy-plane defined by the level curve $f(x, y) = 1$. To do this, we realized that if we take a total of N_t randomly distributed points inside a rectangle with sides of length a in the x-direction and b in the y-direction, and count how many of the points, N_i, are inside the

curve, then $N_i/N_t = A_i/A_t$, where A_i is the area inside the curve, and $A_t = ab$ is the area of the bounding rectangle. Hence, if we solve for the area inside the curve $f(x,y) = 1$, we arrive at $A_i = \iint_R dA = (ab)\, N_i/N_t$. Specifically in lab, we used the curve $f(x,y) = x^2 + y^2 = 1$ and arrived at the value $A_i = \iint_R dA = \pi/4$.

We also considered code to estimate the area inside the curve $f(x,y) = \dfrac{x^2}{9} + y^2 = 1$, in the first quadrant. Note that the smallest rectangle that encloses the curve in the first quadrant is three units long in the x-direction and one unit long in the y-direction.

Finally, we considered the problem of evaluating the mean value of a function $g(x,y)$ over the region R is defined as $\bar{g}_R = \iint_R g(x,y)\, dA \Big/ \iint_R dA$. However, the mean value can also be estimated by taking the sum of a large sample of function values over the region and calculating the average, $\bar{g}_R \approx \dfrac{1}{N} \sum_{i=1}^{N} g(x_i, y_i)$. Equating these two expressions for $\bar{g}_R$, we see that

$$\iint_R g(x,y)\, dA \approx \left(\frac{A}{N}\right) \sum_{i=1}^{N} g(x_i, y_i)\,, \tag{1}$$

where $A = \iint_R dA$ is the area of the region R, as calculated in the previous problem. (Also note that N in this problem is N_i from the previous problem.)

Write (modify) the code demonstrated in class to perform the calculations necessary to estimate the value of equation 1. Your program should meet the following requirements:

- Your main program file should be called `TheIntegral.m`. It should have no input values, but should print out the value of equation 1.
- Use the file named `TheBoundary.m` to evaluate $f(x,y) = \dfrac{x^2}{9} + y^2$. It should reside in the same folder as the main file `TheIntegral.m`.
- Although the actual function $g(x,y)$ may vary, when we test your program we will use a function file named `TheFunc.m` containing something similar to the following code:

  ```
  function g = TheFunc(x,y)
  g = 2*x + 3*y;
  ```

  The file `TheFunc.m` should also reside in the same folder containing the file `TheIntegral.m`.

As you write up this problem, be sure to comment on how many samples must be taken to correctly estimate (within reason) the integral value to, say 2, significant figures. Also, pick a few different $g(x,y)$ functions with known integral values to test your program.

Please observe the following guidelines for homework submission.

Hard-copy work should:

- be turned in at the beginning of class on the due date.
- have all pages stapled together.
- contain a clear statement of the problem, in your own words.
- as appropriate, contain a clear discussion of the results of the problem/program. (Do the results make sense? Are they reasonable? Why, or why not? Do you *trust* the results of your program? Does one method seem to work better than another?)
- contain a copy of the code and a meaningful sample of the output.

Electronic work should:

- contain the following comment lines at the beginning of each file, (with the appropriate information filled in):

```
% a one-line description of the program
% your name
% today's date
% APPM 3050, Homework XX
```

- use lots of white-space, indentation, and comments to make the code readable and understandable, and provide enough information to clearly explain what is being done at various points in the program.
- have all code and any output inside a directory (folder) named `YourLastName`. This directory should be "zipped" in an archive file named `YourLastName.zip` and submitted via email to **adam@colorado.edu** with a subject line **YourLastName HWxx submission**.