CS1112 Exercise 7 Name: __________________ NetID: ________

You have until *Thursday, 10/13, at 9pm* to get Problem 1 this exercise checked off (during this discussion section or during *consulting hours* or *TAs' office hours*). Problems 2 and 3 are to be submitted using MATLAB *Grader* by the same deadline.

1 Linear interpolation (GET CHECKED OFF BY YOUR TA)

For this problem, you must discuss your solution with a TA or Consultant during discussion or consulting/office hour to get it checked off. No online submission or emailed code will be accepted.

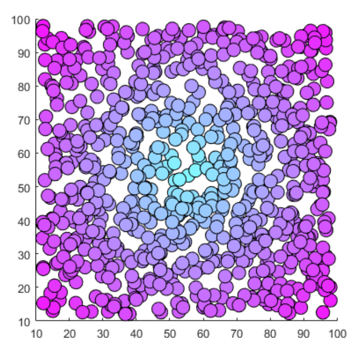
Implement the following function as specified. Do not use any built-in functions other than rand, sqrt, and sum.

```
function randSpheres_interp(N, r, axisLims, colr1, colr2)
% Draws random spheres in a box with colors interpolated between colr1 and colr2
% The x values of the box range from axisLims(1) to axisLims(2)
% The y values of the box range from axisLims(1) to axisLims(2)
% inputs:
%   N: number of spheres to draw
%   r: radius of every sphere (scalar value)
%   axisLims: length 2 vector that defines the axis limits of the box
%   colr1: color of sphere generated in the center of the box
%   colr2: color of sphere generated furthest from the center of the box
```

Some notes:

1. The entirety of each sphere should lie inside the box and the center of each sphere should be randomly generated.
2. The distance between points $(x1, y1)$ and $(x2, y2)$ is $d = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$.
3. A disk generated at the exact center of the box should have color colr1 and the disk that could be potentially generated at the furthest possible point in the box should have color colr2. Disks at intermediate distances between these two extremes should be interpolated between colr1 and colr2.
4. All disks should show up on the same plot (use `hold on`).
5. Make use of DrawDisk.
6. **First deal with the problem without the color interpolation, then deal with the color interpolation.**

If you run `randSpheres_interp(800, 2, [10, 100], [0.5, 1, 1], [1, 0, 1]);` in the command window, MATLAB should create a figure like the following:



2 Average neighbor vector (Answer in MATLAB Grader https://grader.mathworks.com)

The neighborhood of an element in a vector includes the element itself and the elements to it's left and right. Write a function aveEvenNeighbors that takes a vector v as input and outputs a vector storing the average value of the neighborhood of each even indexed element in v. For example, for a vector $v = [2, 3, 4, 10, 1]$, the output of aveEvenNeighbors would be $[3, 5]$ because we only store the the average value of the neighborhood of the 2nd element (avg([2, 3, 4]) = 3) and the 4th element (avg([4, 10, 1]) = 5). You can assume that the length of the input to the function will always be odd and of length greater than or equal to 3.

Be efficient: do not loop through every index.

```
function medVec = aveEvenNeighbors(v)
% Computes the average even neighborhood vector of v
% v: vector with odd length and length(v) >= 3
% medVec: vector storing the average neighborhood values for each even index
```

3 Find a value in a matrix (Answer in MATLAB Grader https://grader.mathworks.com)

Implement the following function as specified. Use loops in this problem; *do not* use any built-in functions other than `size`.

```
function [rvec, cvec] = findInMatrix(n,M)
% Find all occurrences of the number n in matrix M.
% rvec and cvec are column vectors of row and column numbers such that
%   M(rvec(k),cvec(k)) is equal to n.
% The length of rvec and cvec is the number of times n appears in M.
% If n is not found in M, rvec and cvec are empty vectors.
% Do not use any built-in functions other than size().
```