In this homework, you will review logical indexing in MATLAB, a very important technique used to efficiently index a matrix or vector. You will also have a chance to work with functions in a separate file.

1. **Roll the Dice** In this part you will work with ***imshow*** and logical indexing.
   - Create 100x100 matrices A, B and C of all ones.
   - In matrix A, set the values of the entries a_{ij} equal to zero if $\sqrt{(i-25)^2+(j-75)^2} < 10$ *or* $\sqrt{(i-75)^2+(j-25)^2} < 10$. (**Hint:** Use ***meshgrid*** to create a matrix with values taking on the lefthand side of the inequalities, then logical indexing on A using |, & or ~).
   - In matrix B, set the values of the entries b_{ij} equal to zero if $\sqrt{(i-25)^2+(j-25)^2} < 10$ *or* $\sqrt{(i-75)^2+(j-75)^2} < 10$.
   - In matrix C, set the values of the entries c_{ij} equal to zero if $\sqrt{(i-50)^2+(j-50)^2} > 10$.
   - Now use ***figure*** and ***imshow*** to plot:
     - The complement of C
     - A
     - The next 3 faces of a die (so 3-5) on three separate figures. Use whatever logical operations (&, |, or ~) are necessary to accomplish this.

2. **Fun with find** Write a function to return the value and index of the first value in a vector with entries that are strictly decreasing that is below a certain threshold. The function should be called as $[val, ind] = findThreshold(x, threshold)$. This function can be accomplished in less than five lines. You may find ***min*** and/or ***find*** useful. **Hint**: Be sure to check out the Name-Value arguments of these functions and don't use a for loop! Show that it works by finding the first value below to 0.05 (and index of said value) in *normpdf(linspace(0,5,100),0,1)*.

3. **Sincing Ship** This problem will involve a function encountered often in signal processing: the *sinc* function.
   - Write a function, called *signSwitch*, in a separate file which inputs a vector v and outputs a vector with the indices i which represent a sign change in v; i.e. report 15 if the sign changed in v between index 14 and index 15. Do not consider going from positive or negative to zero. We could loop through and check this condition at every point - don't do that. Instead think of a way to use logical indexing: One suggestion is to write conditions on the vector and some kind of shifted version of itself. Beware however, when you do this you will have non-overlapping points. It is up to you to figure out what to with them. This will be a local function (see the documentation if you forgot what this means).

- Now we will write the main function, call it whatever you'd like, which will perform all the analysis. The function will input two vectors representing the x and y coordinates of the graph of the function and will return a vector with the approximate indices of the zero crossings and a vector with the approximate indices of the local extrema.
- First have the function use your local function to find the zero crossings.
- Next, have the function take the first derivative (approximately, see homework 2). Then use your local function to apply the first derivative test to see where the approximate local minima and maxima are.
- Finally, have the function plot all this information using ***figure*** then *plot(x,y,xzero,yzero,'r*',xminmax,yminmax,'ko')*. This will plot the function then plot red stars on the zero crossings and black circles on the local minima.
- Apply your function to $sinc(x)$ sampled at 9999 points on [-5.25,5.25]. Put a comment regarding what you see regarding the zero crossings.
- Now extract (you can hard-code the indices if you want) from your vector of sinc values the values between the middle extrema and the first zero crossing after that (this region should be strictly decreasing; if your graph doesn't show you that then you probably did something wrong) and apply your $findThreshold$ function to find the x value where the graph first goes below 0.2.