Name: ____________________________
(Legibly print last name, first name, middle name)
NetID: ____________________________

Statement of integrity:
I did not, and will not, violate the rules of academic integrity on this exam.
(Signature)

Circle your lecture time:  9:05  or  11:15

Circle your section instructor’s name:

<table>
<thead>
<tr>
<th>Time</th>
<th>Tuesday</th>
<th>Wednesday</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:10</td>
<td></td>
<td>Amy Cochran / Wilson Zhou</td>
</tr>
<tr>
<td>11:15</td>
<td></td>
<td>Jeff Ames / Wilson Zhou</td>
</tr>
<tr>
<td>12:20</td>
<td>Ankit Arora</td>
<td>Amy Cochran / Wilson Zhou</td>
</tr>
<tr>
<td>1:25</td>
<td>Nipun Jasuja</td>
<td>Amy Cochran</td>
</tr>
<tr>
<td>2:30</td>
<td>Nipun Jasuja</td>
<td>Tim English</td>
</tr>
<tr>
<td>3:35</td>
<td>Jean Rouge</td>
<td>Nihar Naigaonkar</td>
</tr>
<tr>
<td>7:30</td>
<td></td>
<td>Ankit Arora</td>
</tr>
</tbody>
</table>

Instructions:
• This is a 90-minute, closed-book exam; no calculators are allowed.
• The exam is worth a total of 100 points, so it’s about one point per minute!
• Read each problem completely, including any provided code, before starting it.
• Raise your hand if you have any questions.
• Use the backs of pages or ask for additional sheets of paper as necessary.
• Clarity, conciseness, and good programming style count for credit.
• If you supply multiple answers, we will grade only one.
• Use only MATLAB code. No credit for code written in other programming languages.
• Assume there will be no input errors.
• Write user-defined functions only if asked to do so. Do not write subfunctions.
• Do not use switch, try, catch, or break statements.
• You may find the following MATLAB predefined functions useful:
  abs, sqrt, rem, floor, ceil, rand, zeros, ones, length, size, fprintf, disp, uint8, double,
  char, strcmp, cell, struct

Examples:
rem(5,2) → 1, the remainder of 5 divided by 2
floor(6.9), floor(6) → 6, rounds down to the nearest integer
ceil(8.1), ceil(9) → 9, rounds up to the nearest integer
zeros(2,4) → a 2-by-4 matrix of zeros, type double
strcmp(’cat’, ’Cat’) → 0, the two strings are not identical
cell(3,2) → a 3-by-2 cell array, each cell is the empty numeric vector []
struct(’a’,1,’b’,0) → a structure with 2 fields: a has value 1, b has value 0
Question 1: (15 points)

(a) Implement this function:

```matlab
function tot = sumDiag(M)
% tot is the sum of the elements on the main diagonal of numeric square matrix M.
% A matrix is square if its number of rows and number of columns are the same.
% Assume M is not empty. For example, if M is
% [ 10  1  2 ;
%   0  30 99 ;
%  -3  3  20 ]
% then tot is 60.
%
% THE ONLY BUILT-IN FUNCTION ALLOWED IS size.
```

(b) Complete the statement below to assign to variable `ch` a randomly generated capital (upper case) letter; each of the 26 letters in the alphabet should be equally likely to occur. *Only the built-in functions listed on the cover page of this exam are allowed.*

```
ch = ________________________________________________________________
```
Question 2: (30 points)

Implement this function:

```matlab
function newIm = enlargeImage(Im)
    % Perform 2-d interpolation on all three layers of image data Im.
    % Im is an nr-by-nc-by-3 array of type uint8 elements. The interpolated data is
    % added between existing data points so array newIm (type uint8) is
    % (2*nr-1)-by-(2*nc-1)-by-3.
    % Use the simple average as the interpolated value (see example below).
    % You may use built-in function zeros for initialization but otherwise
    % DO NOT USE VECTORIZED CODE.
```

*Hint:* In 2-d interpolation, work with one dimension at a time. For example, you can first add the interpolated columns and then add the interpolated rows. For example

<table>
<thead>
<tr>
<th>One layer of M</th>
<th>Interpolate columns</th>
<th>Interpolate rows</th>
</tr>
</thead>
</table>
| \[
\begin{bmatrix}
250 & 50 \\
20 & 100 \\
10 & 130 \\
\end{bmatrix}
\] | \[
\begin{bmatrix}
250 & 150 & 50 \\
20 & 60 & 100 \\
10 & 70 & 130 \\
\end{bmatrix}
\rightarrow
\begin{bmatrix}
250 & 150 & 50 \\
135 & 105 & 75 \\
20 & 60 & 100 \\
15 & 65 & 115 \\
10 & 70 & 130 \\
\end{bmatrix}
\] |
Question 3: (30 points)

(a) Implement this function:

```matlab
function z = overlap(diskA, diskB)
    z = 1; % diskA and diskB overlap; otherwise z is 0 (false).
end
```

(b) Implement the following function to return the indices of disk triplets that overlap. Three disks form a triplet if every disk overlaps with each of the other two. Make effective use of function overlap from part (a). Your code should be efficient—avoid unnecessary iterations.

```matlab
function idx = diskTriplets(D)
    idx = []; % diskA and diskB are each a disk structure with the following fields:
    % x: x-coordinate of center of disk
    % y: y-coordinate of center of disk
    % radius: radius of disk
end
```
Question 4: (25 points)

We will split a string into two parts at the first occurrence of a “marker.” For example, if the original string is ‘acagttaga’ and the marker is ‘ag’, then we split the original string into these two parts: ‘ac’ and ‘agttaga’. Note that the marker is included in the second part. Implement the following function and note the example at the bottom of the page.

```matlab
function CA = split(M, mar)
    % Split each row of matrix M into two parts at the first occurrence of the marker
    % (parameter mar); each part is stored in one cell in a row of 2-d cell array CA.
    % M is a matrix of characters; assume M is not empty.
    % mar is a vector of characters; assume mar is not empty.
    % CA is an nr-by-2 cell array of strings, where nr is the number of rows in M.
    % THE ONLY BUILT-IN FUNCTIONS ALLOWED ARE strcmp, size, length, cell.
    % HINT: For each row, first search for the position of the marker.

    % For example, if mar is the string 'ag' and M is
    ['aaggagtt' ; ...
     'atttcag ' ; ...
     'ag ' ; ...
     'aaaaaaaa' ]

    Then CA is a 4-by-2 cell array:
    Row 1: column 1 is the string 'a', column 2 is the string 'aggagtt'
    Row 2: column 1 is the string 'atttc', column 2 is the string 'ag'
    Row 3: column 1 is the empty string, column 2 is the string 'ag'
    Row 4: column 1 is the string 'aaaaaaaa', column 2 is the empty string
```