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.
• Write user-defined functions only if asked to do so.
• Do not write subfunctions.
• Do not use switch, try, catch, break, or continue statements.
• You may find the following MATLAB predefined functions useful:
  abs, sqrt, rem, floor, ceil, rand, zeros, ones, sum, mean, length, size, fprintf, disp, bar, uint8, double, char, strcmp, isempty, str2double, fopen, fclose, fgetl, feof, cell

Examples:
rem(5,2) → 1, the reminder of 5 divided by 2
rand, rand(1) → a random real value in the interval (0,1)
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
linspace(2,9,5) → a vector with 5 values uniformly distributed in the interval [2,9]
cell(3,2) → a 3-by-2 cell array, each cell is the empty numeric vector []
strcmp(’cat’,’Cat’) → 0, the two strings are not identical
str2double(’-2.6’) → -2.6, a type double scalar
Question 1: (15 points)

(a) What is the output from executing the following script? If the program doesn’t terminate or if there will be an error during execution, write the word “error” instead of the output.

```matlab
M = [ 12 16 9 5 7 1; ... 13 17 80 1 5 7; ... 68 66 13 7 1 5];
for r = 1:3
    c = 1;
    d = 6;
    while c<3
        M(r,c) = M(r,d);
        c = c + 1;
        d = d - 1;
    end
end
disp(M)
```

Output:

(b) Implement the following function as specified:

```matlab
function s = sumMatrix(M)
    % s is the sum of all the values in columns 2, 4, 6, ... of matrix M.
    % M is of type double and has more than three columns.
    % DO NOT USE any built-in functions other than size.
```

Exam score:

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>(15)</td>
</tr>
<tr>
<td>Q2</td>
<td>(10)</td>
</tr>
<tr>
<td>Q3</td>
<td>(15)</td>
</tr>
<tr>
<td>Q4</td>
<td>(25)</td>
</tr>
<tr>
<td>Q5a</td>
<td>(10)</td>
</tr>
<tr>
<td>Q5b</td>
<td>(25)</td>
</tr>
<tr>
<td>Total</td>
<td>(100)</td>
</tr>
</tbody>
</table>
Question 2: (10 points)

$W$ is a 1-d cell array of strings. Each cell in $W$ stores one word and all the words in $W$ are different. Write a code fragment below to randomly select and print two different words from cell array $W$; each word in $W$ should be equally likely to be selected.

% Assume $W$, 1-d cell array of strings as described above, is given.
% $W$ has a length greater than 3.
% ONLY these built-in functions are allowed: length, rand, ceil, floor, strcmp, disp
Question 3: (15 points)

Implement the following function as specified:

```matlab
function d = darkestInLowerTriangle(A)
    % A is a 3-d uint8 array of image data. The number of rows in A is the
    % same as the number of columns.
    % The gray value of a pixel is the arithmetic mean of the red, green, and
    % blue values of a pixel.
    % d is the darkest gray value of all the pixels in the lower left triangular
    % part of the image, including the main diagonal. d is a uint8 scalar.
    % DO NOT USE any built-in functions other than size and uint8.
```
Question 4: (25 points)

Complete the following function as specified. For full credit, *do not use* vectorized code. Note the example at the bottom of the page.

```matlab
function B = rearrange(A,n)

% A is a 3-d uint8 array of image data and n is a positive integer.
% A is to be divided into n blocks of equal size horizontally. Assume that the number
% of columns in A is divisible by n.
% In every other block beginning with the first block, mirror the pixels row-wise, i.e.,
% flip that block upside down. Do not change the order of the columns and layers.
% B is A rearranged as specified above. B has the same size and type as A.

B = A;

[nr, nc, np] = size(A);

colsInBlk = nc/n;  % number of columns of pixels in a block

% Write your code below. For full credit DO NOT USE vectorized code.
```

Example: Suppose array A is $5 \times 6 \times 3$ and is to be divided into 3 blocks. Below left is one layer of the original array values in A; on the right is that layer with blocks 1 and 3 flipped row-wise in B.

*Decompose the problem!* Can you flip just one block? How do you flip multiple blocks?

```
<table>
<thead>
<tr>
<th>Blk 1</th>
<th>Blk 2</th>
<th>Blk 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 11 10 13 17 19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 31 30 33 27 29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 41 40 43 47 49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 61 60 63 67 69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 71 70 73 77 79</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>Blk 1</th>
<th>Blk 2</th>
<th>Blk 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 71 10 13 77 79</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 61 30 33 67 69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 41 40 43 47 49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 31 60 63 27 29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 11 70 73 17 19</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Question 5a: (10 points)

Implement the following function as specified:

```matlab
function v = getIndices(str, sep)
    % str and sep are each a string and str is longer than sep. sep is the
    % separator string, i.e., the delimiting string.
    % v is the vector of the indices where the separator begins in str.
    % Therefore the length of v is the number of times that sep occurs in str.
    % Examples: If str is 'Hi!?Ann!?Bob' and sep is '!?' then v is [3 8].
    %         If str is 'Hi!Ann!Bob' and sep is '!' then v is [3 7].
    %         If str is 'Hi!Ann!Bob' and sep is '?' then v is [].
    % Assume that the characters in sep are used only as the delimiter and not
    % in the separated substrings. Assume that separators are always correctly
    % placed--never incomplete and never side-by-side not separating anything.
    % DO NOT USE any built-in functions other than length and strcmp.
```
Question 5b: (25 points)

Assume that function `getIndices` of Question 5a has been correctly implemented; make effective use of it in implementing function `aveScores` below. Note the example at the bottom of the page.

```plaintext
function CA = aveScores(M)
% M is a 2-d array of characters. Each row of M stores the scores of one student:
% a netID followed by one or more scores and these data items are separated by
% commas. There may be trailing spaces in a row of M.
% CA is an n-by-2 cell array where n is the number of students whose record includes
% at least two scores. In each row of CA, the first cell stores the netID of a
% student who has at least two scores and the second cell stores the average score
% of that student. If no student has at least two scores then CA is an empty cell
% array.
% ONLY these built-in functions are allowed: length, size, str2double, sum, mean
% Recall that str2double can handle leading and trailing spaces, e.g.,
% str2double('87   ') returns the type double scalar 87.
```

Example: Suppose `M` is

```plaintext
['vaf34,80,100,90';
'aaj91,100    ';
'rt2253,75,95 ']
```

Then `aveScores(M)` should return a 2 × 2 cell array `CA`:

- In row 1 column 1 is 'vaf34' and in row 1 column 2 is the type double scalar 90.
- In row 2 column 1 is 'rt2253' and in row 2 column 2 is the type double scalar 85.

**Hint:** Decompose the problem! For each row of `M` you need to solve several subproblems: ● find the locations of the commas, ● convert each score substring to a number, ● calculate the average. Organize the subtasks first and then work on them one at a time.