1. **(a)** Describe in English what the following script displays. Assume that `ClockTower.jpg` represents a color image and is in the current working directory.

```matlab
A = imread('ClockTower.jpg');
B = rgb2gray(A);
[m,n] = size(B);
C = zeros(n,m,'uint8');
for i=1:m
    C(:,i) = B(m+1-i,:);
end
imshow(C)
```

**Solution**
The original color image is displayed in black-and-white (1 point) and is rotated 90 degrees (3 points) clockwise (1 point).

"The image is flipped and black and white" = 3 points.

1. **(b)** What is the output if the following script is run?

```matlab
x = uint8(200);
y = uint8(300);
a = (x+y)/2
b = double((x+y)/2)
```

**Solution**
The value of \( y \) is 255 since a uint8 variable houses integers between 0 and 255.
The value of \( x+y \) is 255
The value of \( a \) is 128
The value of \( a \) and \( b \) is the same.

**Full credit:**

```
128 128
or 127 127
```

**Sample -1 deductions:**
Output with fractions
Different \( a \) and \( b \):
Think uint8 values between something different than 1-to-255

2. Recall that the built-in function `strcmp` has the property that `strcmp(s1,s2)` is 1 if \( s1 \) and \( s2 \) are identical strings and 0 otherwise. Assume that \( C \) is an initialized cell array of strings and that \( s \) is an initialized string. Complete the while-loop condition so that the following fragment is correct:

**Solution**
k = 1;
while k <= length(C) && strcmp(s,C{k})==0
    k = k+1;
end
if k > length(C)
    disp('The string in s does not occur in C')
end

3 points: \( k \leq \text{length}(C) \)

2 points: && not ||

4 points: strcmp(s,C{k})==0

1 point for correct order, i.e., \( k \leq \text{length}(C) \) && strcmp(s,C{k})==0
instead of

(Otherwise you will get a subscript out of bounds.)
3. Consider the following definitions:

**Definition 1.** For a given black-and-white image, we say that pixel \((i, j)\) is an *interior pixel* if it is not on the edge of the image.

**Definition 2.** For a given black-and-white image, we say that pixel \((i_1, j_1)\) is a *neighbor* of pixel \((i_2, j_2)\) if \(|i_1 - i_2| + |j_1 - j_2| \leq 1.\)

**Definition 3.** For a given black-and-white image, we say that a pixel is *very bright* if it is an interior pixel and each of its neighbors has a lesser intensity.

Complete the following function so that it performs as specified:

```matlab
function C = VeryBrightPixels(X)
% X names a black-and-white jpg file in the current directory.
% C is a cell array of length-2 vectors that collectively identify
% all the bright pixels in X. Thus, if the function outputs the cell
% array \{[40,300],[200,30],[100,150]\}, then pixels (40,300), (200,30),
% and (100,50) are the very bright pixels.

A = rgb2gray(imread('X.jpg'));

[m,n] = size(A);
k=0
for i=2:m-1
    for j = 2:n-1
        if A(i,j)>A(i-1,j) && A(i,j)>A(i+1,j) &&
            A(i,j)>A(i,j-1) && A(i,j)>A(i,j+1)
            k = k+1;
            C{k} = [i j];
        end
    end
end

Note that the loops visit only interior pixels
Note that an interior pixel \((i,j)\) has four neighbors:
\((i-1,j), (i+1,j), (i,j-1), (i,j+1)\)

Loop ranges 2 points
if condition 4 points
The k counter 2 points
Cell array assignment 2 points