This exam consists of 5 questions (11 pages, including this page) worth a total of 100%. It is an open book exam. All answers are to be written in this booklet. Scrap work may be done on the back of each page; this will not be marked. The exam is 2 hours long and comprises 30% of your final mark. Assignments 1 and 2 are each worth 35% respectively, for a total of 70%.

Professor: John Barron
Answer the following questions briefly and concisely and show all relevant work. Where possible, use point-form. Generally, correct answers will be short.

(1) (20%) Consider the following MatLab morphological operators: \( A \oplus B \), the dilation of image \( A \) by structuring element \( B \) and \( (A \ominus B) \), the erosion of image \( A \) by structuring element \( B \). Figure 1 show the original grayvalue image \( A \). It is a \( 256 \times 256 \) image with a centered \( 64 \times 64 \) image of grayvalues of 1’s in a surrounding uniform grayvalue pattern of 0’s. The following MatLab code generates this images as well as a 4 other images:

```matlab
A=zeros(256,256,'int8');
C=zeros(256,256,'int8');
D=zeros(256,256,'int8');
E=zeros(256,256,'int8');
F=zeros(256,256,'int8');
A(97:160,97:160)=1;

imshow(A,[0 255]);
title('Original Image A');
print final09_image_A.eps -deps

% 11*11 structuring element
% centered at (6,6)
B=strel([1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;...
          1 1 1 1 1 1 1 1 1 1 1;]);
C=imdilate(A,B);
figure
imshow(C,[0 255]);
title('Dilated Image C');
print final09_q1a.eps -deps
```

Where is the surrounding uniform grayvalue pattern of 0’s?
D = imerode(A, B);
figure
imshow(D, []);
title('Eroded Image D');
print final09_q1b.eps -deps

E = C - D;
figure
imshow(E, []);
title('Difference C-D');
print final09_q1c.eps -deps

F = D - C;

% Put black border around the image - this
% is also a hint!!!
F(1:3,:) = -1;
F(254:256,:) = -1;
F(:,1:3) = -1;
F(:,254:256) = -1;
figure
imshow(F, []);
title('Difference D-C');
print final09_q1d.eps -deps

Figure 1: Original grayvalue image A.
(1a) (5%) Show the image resulting from computing \( C = A \oplus B \). Indicate black and white areas clearly.

![Dilated Image C](image1)

Figure 2: \( C = A \oplus B \).

(1b) (5%) Show the image resulting from computing \( D = A \ominus B \). Indicate black and white areas clearly.

![Eroded Image D](image2)

Figure 3: \( D = A \ominus B \).
(1c) (5%) Show the image resulting from computing $E = A \oplus B - A \ominus B = C - D$. Indicate black and white areas clearly.

![Difference C−D](image)

Figure 4: $E = A \oplus B - A \ominus B = C - D$.

(1d) (5%) Show the image resulting from computing $F = A \ominus B - A \oplus B = D - C$. Indicate black and white areas clearly.

![Difference D−C](image)

Figure 5: $F = A \ominus B - A \oplus B = D - C$. 
(2) (20%) Consider the following MatLab code:

```matlab
f = double(imread('gray_lena.ras'));
tau = 2.5;
black = 0;
white = 1;
g1 = ones(size(f,1),size(f,2),'double');
g2 = ones(size(f,1),size(f,2),'double');
f11 = zeros(size(f,1),size(f,2),'double');
f12 = zeros(size(f,1),size(f,2),'double');
h = [1 -8 0 8 -1]/12.0;
f1 = imfilter(f,h,'conv','symmetric','same');
f2 = imfilter(f,h,'conv','symmetric','same');
for i = 2:size(f,1)-1
    for j = 2:size(f,2)-1
        f11(i,j) = f1(i,j)*f1(i+1,j);
f12(i,j) = f2(i,j)*f2(i+1,j);
        if f11(i,j) > tau
            g1(i,j) = black;
        end
        if f12(i,j) > tau
            g2(i,j) = black;
        end
    end
end
imshow(g1,[]);
title('title 1 here');
print final09_title1.eps -deps
figure
imshow(g2,[]);
title('title 2 here');
print final09_title2.eps -deps
```
(2a) (5%) Figure 6 shows the output images generated by this program. What does the program do?

Suggest replacements for title 1 and title 2 in the title commands.

title 1:

title 2:
(2b) (15%) Vectorize this code. Hint: \texttt{circshift} will be useful. Note that 
\texttt{circshift(A,[1 0])} circularly shifts rows of \texttt{A} down by 1, 
\texttt{circshift(A,[0 1])} circularly shifts columns right of \texttt{A} by 1 while 
\texttt{circshift(A,[-1 -1])} circularly shifts rows up by 1 and columns left by 1 at the same time. [Note this 
means borders are handled by wraparound.]

\begin{verbatim}
f=double(imread('gray_lena.ras'));
tau=2.5;
black=0;
white=1;
g1=ones(size(f,1),size(f,2),'double');
g2=ones(size(f,1),size(f,2),'double');
f11=zeros(size(f,1),size(f,2),'double');
f12=zeros(size(f,1),size(f,2),'double');
h=[1 -8 0 8 -1]/12.0;
f1=imfilter(f,h,'conv','symmetric','same');
    \% Note the transpose operator ' on h
f2=imfilter(f,h','conv','symmetric','same');
q1=circshift(f1,[-1 0]);
q2=circshift(f2,[-1 0]);
f11=f1.*q1;
f12=f2.*q2;
g1(f11>tau)=black;
g2(f12>tau)=black;
imshow(g1,
            title('Vertical Edges');
print final09_vertical.eps -deps
figure
imshow(g2,
           title('Horizontal Edges');
print final09_horizontal.eps -deps
\end{verbatim}
(3) (20%) Consider $3 \times 3$ median filtering or $3 \times 3$ averaging. Which of these 2 methods is best for suppression of random Gaussian noise in the spatial domain and why?
(4) (20%) Consider representing an arbitrary shape, described initially as a 512 vertex polyline, using Fourier transform (FT) coefficients or using discrete cosine transform (DCT) coefficients. For a curve with an arbitrary shape which technique would be best and why for:

(10%) The best shape representation given a fixed number, \( n, n << 512 \), of coefficients:

(10%) The lowest computational cost for a fixed number, \( n, n << 512 \), of coefficients:

An answer without justification or the wrong justification is worth nothing.
(5) (20%) Consider a 1D signal (image), $f(x), x \in [1, 512]$. We can compute its Fourier transform as $F(u)$, with frequencies indexed by $u \in [1, 512]$. What can you say about the frequency content at pixel 56? State your answer concisely.