Morphological operations on binary images.

In this exercise you practice to use the basic morphological operations erosion and dilation on binary images.
Erosion is written as b=(b1 θ se), where b1 is the original binary image, se is the *structure element*, and b is the eroded image. The symbol θ is used for the erosion operation. Erosion is implemented as a *Min-filter* (i.e. the result is the minimum of the image pixels pointed out by the structure element).
Dilation is written as b=(b1 ⊕ se), where b1 is the original binary image, se is the structure element, and b is the dilated image. The symbol ⊕ is used for the dilation operation. Dilation is implemented as a *Max-filter* (i.e. the result is the maximum of the image pixels pointed out by the structure element).

1. Erosion
Create a binary image b1, and two structure elements se1 and se2.
Erode the binary image b1, using the two structuring elements se1 and se2.

```
% create the binary image b1
b1=zeros(16,16); %white object on a black background
b1(7:9,4:12)=ones(3,9);
subplot(2,2,1); imshow(b1); axis on %display
% create the structuring elements
se1=ones(1,3); % 1 x 3
se2=ones(3,1); % 3 x 1
se1  %print them on screen
se2  %
% erode the image b1
e1b1=imerode(b1,se1); %by se1
e2b1=imerode(b1,se2); %by se2
subplot(2,2,2); imshow(e1b1); axis on;
subplot(2,2,4); imshow(e2b1); axis on;
```

Erode b1 by using the 2D structure element se=ones(3,3). Compare this result to the previous results using se1 and se2 (se=se1 ⊕ se2).
What is the general result of erosion?
Do one of the above erosions yourself on a paper. Check against MATLAB result!
You want the object to disappear when doing the erosion. How should the structuring element look like for doing this? Test your idea in MATLAB.

2. Dilation
Create a binary image b2.
Dilate the binary image b2, using the two structuring elements se1 and se2.

```
% create the binary image b2
b2=zeros(16,16);  %black background
b2(4:12, 6:10)=ones(9,5); %object
b2(6:10, 7:9)=zeros(5,3); %hole
subplot(2,2,1); imshow(b2); axis on;
% dilate the image b2
d1b2=imdilate(b2,se1); %by se1
d2b2=imdilate(b2,se2); %by se2
subplot(2,2,2); imshow(d1b2); axis on;
subplot(2,2,4); imshow(d2b2); axis on;
```

Dilate b2 by using the 2D structure element se=ones(3,3). Compare this result to the previous results using se1 and se2 (se=se1 ⊕ se2).

What is the general result of a dilation?
Do one of the above dilations yourself on a paper. Check against MATLAB result!
You want to "fill" the hole in the object when doing the dilation. How should the structuring element look like to do this? Test your idea in MATLAB.

3. Boundary extraction using erosion and dilation
Create a binary image b3. Extract the "inner boundary ib" of the object by using ib=b3/(b3 θ se).
Where the symbol / means set difference, that is "everything in b3 but not in (b3 θ se)".
In the code below the symbols & and ~ means logical AND respectively logical complement (NOT).

b3=zeros(32,32);
b3(6:26,12:20)=ones(21,9);
subplot(2,2,1); imshow(b3); axis on;

se=ones(3,3); % 3 x 3 structuring element
se %print it on screen

b3e=imerode(b3,se);
subplot(2,2,3); imshow(b3e); axis on
ib=b3 & ~b3e; % extract the "inner boundary", can be interpreted as the difference b3-b3e
subplot(2,2,2); imshow(ib); axis on;

Now, extract the "outer boundary ob" of the object, by using ob=(b3 ⊕ se)/b3 (first dilate "then take the difference").
Display the original, the dilated and the outer boundary image.

4. Closing and Opening operations for noise reduction
Reduce "noise" in the binary image b4, i.e. take away small objects in the background and "fill" small holes in the object. The object size should not be altered!

%Create the image b4
b4=zeros(32,32);
b4(6:26,12:20)=ones(21,9); %object
b4(18,15:16)=zeros(1,2); %small holes (noise)
b4(7:9,18)=zeros(3,1);
b4(27,27)=ones(1,1); %small objects (noise)
b4(10:12,3)=ones(3,1);
subplot(2,2,1); imshow(b4); axis on; %display image

Use the Opening operation: (b θ se) ⊕ se and the Closing operation: (b ⊕ se) θ se to apply "noise reduction to the binary image b4. Display the original image and the result images from the opening respectively the closing operation.
Motivate the size of the structuring element se.
Why are you doing imerode and imdilate on the dilated respectively the eroded image for the closing and the opening operation?

5. Extract objects with a special property (here: big objects)
Extract the big object from the binary image b5.
Display the original image and your result image.
Hint:
First, use erosion to "take away" only the small objects (eb5 is the result image of this operation).
Second, recover the big object by using iteratively $k_{i+1}=(k_i \oplus \text{ones}(3,3))$ AND b5 until no changes, wher k_0=eb5.

```
%Create the image b5.
b5=zeros(32,32);

b5(10:24,18:24)=ones(15,7); %the big object
b5(8:9,18)=ones(2,1);  %that should
b5(24,16:17)=ones(1,2); be extracted

b5(4:5,5)=ones(2,1); %small object
b5(28,16:17)=ones(1,2); % small object

subplot(3,3,1); imshow(b5); axis on;  %display image

%delete small objects < se
se=ones(3,3);
eb5=imerode(b5,se);
subplot(3,3,2); imshow(eb5); axis on;
%recover the big object iteratively
k1=imdilate(eb5,se) & b5;
subplot(3,3,4); imshow(k1); axis on;
k2=imdilate(k1,se) & b5;
subplot(3,3,5); imshow(k2); axis on;
```

Complete the above code until the big object is fully recovered.
Explain in words the effect of using the operation imdilate and the logical operation AND in the iteratively recovering of the big object from the eroded original image.

If you have time:
Load the character E and extract its boundary using morphological operations.