

$$361. x = [4 \ 3 \ 7 \ -9 \ 1 \ 0 \ 0]$$

- 求 x 的快速傅立葉轉換。 (5 分)
 - 對 x 做 8 點的快速傅立葉轉換。 (5 分)
 - 對 x 做 6 點的快速傅立葉轉換。 (5 分)
 - 若 x 為多維度矩陣(請自行定義), 觀察 $\text{fft}(x)$ 變化並描述與向量之差異為何。 (5 分)

$$x = [4 \ 3 \ 7 \ -9 \ 1 \ 0 \ 0]$$

$$x = \begin{matrix} 1 \times 7 \\ 4 & 3 & 7 & -9 & 1 & 0 & 0 \end{matrix}$$

`x1=fft(x)`

```
x1 = 1x7 complex
 6.0000 + 0.0000i  11.5206 - 4.8312i  -7.9623 - 7.7059i   7.4417 +13.9204i ...
```

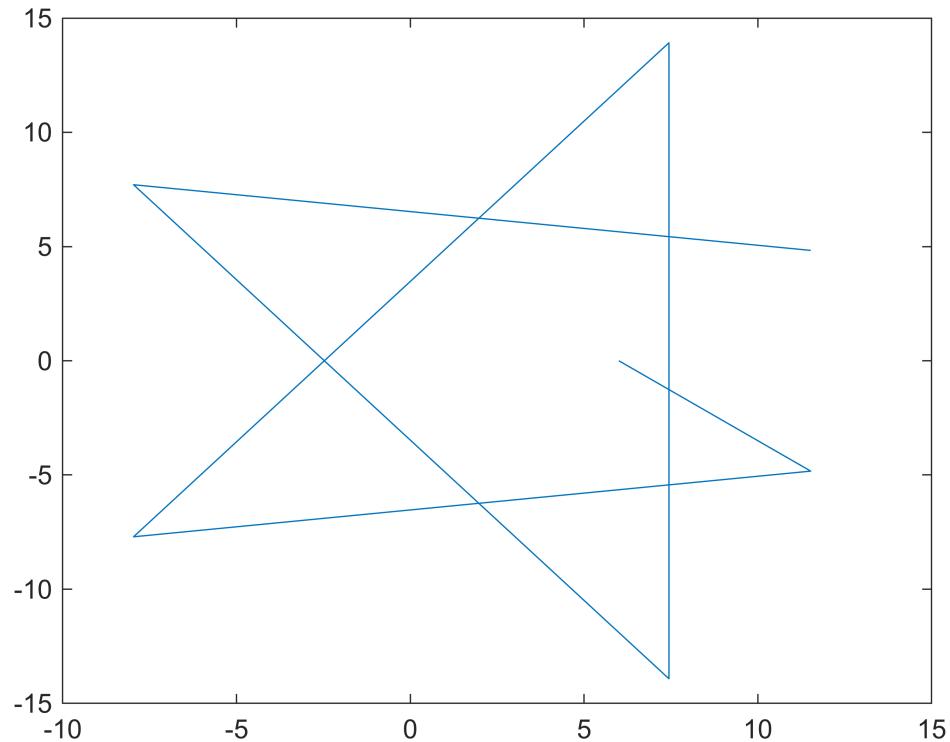
x2=fft(x,8)

```
x2 = 1x8 complex
 6.0000 + 0.0000i  11.4853 - 2.7574i  -2.0000 -12.0000i  -5.4853 +11.2426i ...
```

x3=fft(x,6)

```
x3 = 1x6 complex  
6.0000 + 0.0000i 10.5000 - 7.7942i -10.5000 + 2.5981i 18.0000 + 0.0000i ...
```

plot(x1)



```
y=[4 3 7 -9 ; 1 0 0 0]
```

```
y = 2x4
 4      3      7      -9
 1      0      0      0
```

```
x1=fft(y,8,2)
```

```
x1 = 2x8 complex
 5.0000 + 0.0000i 12.4853 - 2.7574i -3.0000 -12.0000i -4.4853 +11.2426i ...
 1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i 1.0000 + 0.0000i
```

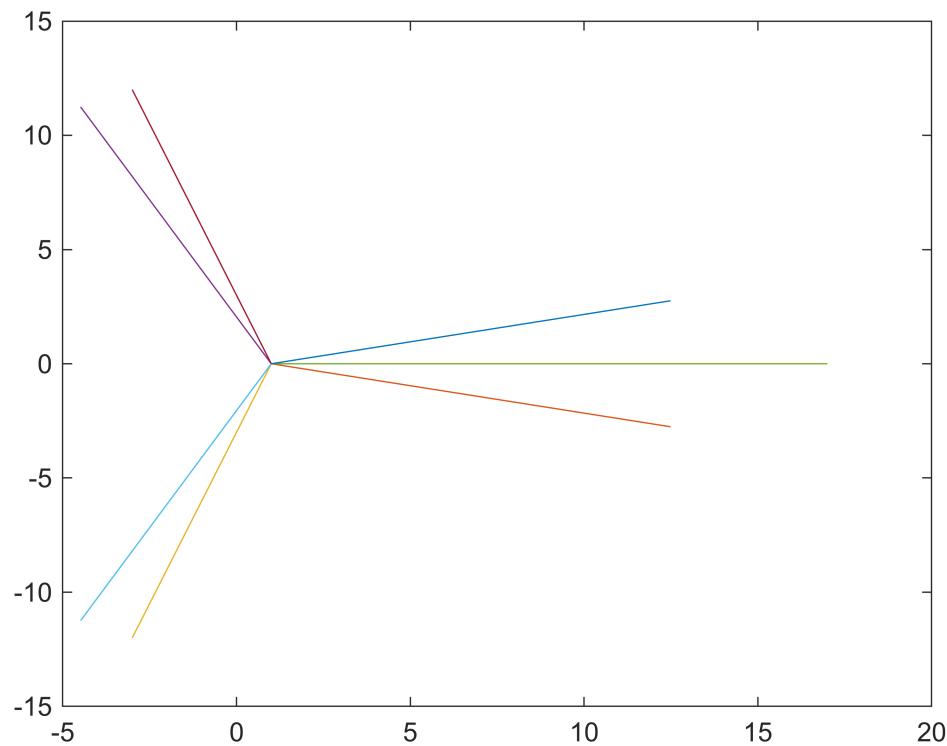
```
x2=fft(y,8)
```

```
x2 = 8x4 complex
 5.0000 + 0.0000i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 4.7071 - 0.7071i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 4.0000 - 1.0000i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 3.2929 - 0.7071i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 3.0000 + 0.0000i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 3.2929 + 0.7071i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 4.0000 + 1.0000i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 4.7071 + 0.7071i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
```

```
x3=fft(y,6)
```

```
x3 = 6x4 complex
 5.0000 + 0.0000i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 4.5000 - 0.8660i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 3.5000 - 0.8660i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 3.0000 + 0.0000i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 3.5000 + 0.8660i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
 4.5000 + 0.8660i 3.0000 + 0.0000i 7.0000 + 0.0000i -9.0000 + 0.0000i
```

```
plot(x1)
```



%%向量是每個連在一起，下面的那張圖變成每列一樣的連在一起

2.請參考講義 FFT 的範例，畫出訊號經過 FFT 處理前與處理後的圖

- 訊號源為 $0.5\sin(2\pi \cdot 50 \cdot \text{time}) + \sin(2\pi \cdot 120 \cdot \text{time})$ (10 分)
- 取樣頻率為 1000 (10 分)

%原本的

```
freq=(-N/2:N/2-1)
```

```
freq = 1x256
-128 -127 -126 -125 -124 -123 -122 -121 -120 -119 -118 -117 -116 ...
```

N=256

N = 256

fs=8000

fs = 8000

freqStep=fs/N

freqStep = 31.2500

```
time=(0:N-1)/fs
```

```
time = 1x256
      0    0.0001    0.0003    0.0004    0.0005    0.0006    0.0008    0.0009 ...
```

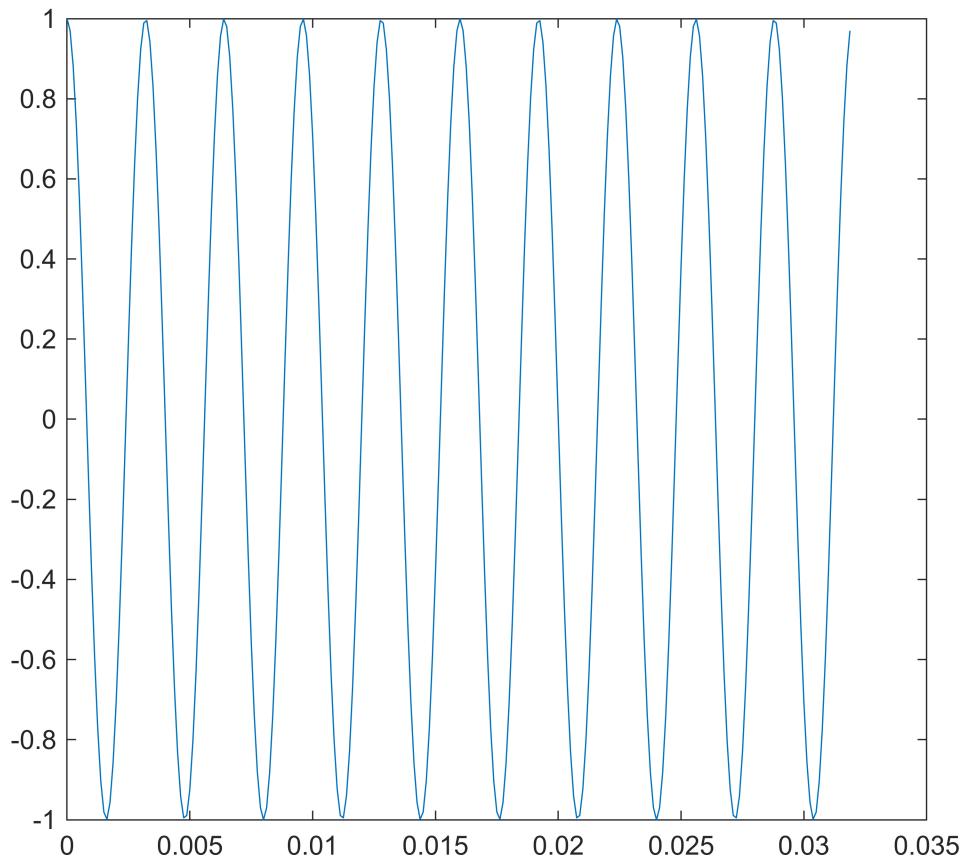
```
f=10*freqStep
```

```
f = 312.5000
```

```
y=cos(2*pi*f*time)
```

```
y = 1x256
  1.0000    0.9700    0.8819    0.7410    0.5556    0.3369    0.0980   -0.1467 ...
```

```
plot(time,y)
```



```
Y=fft(y)
```

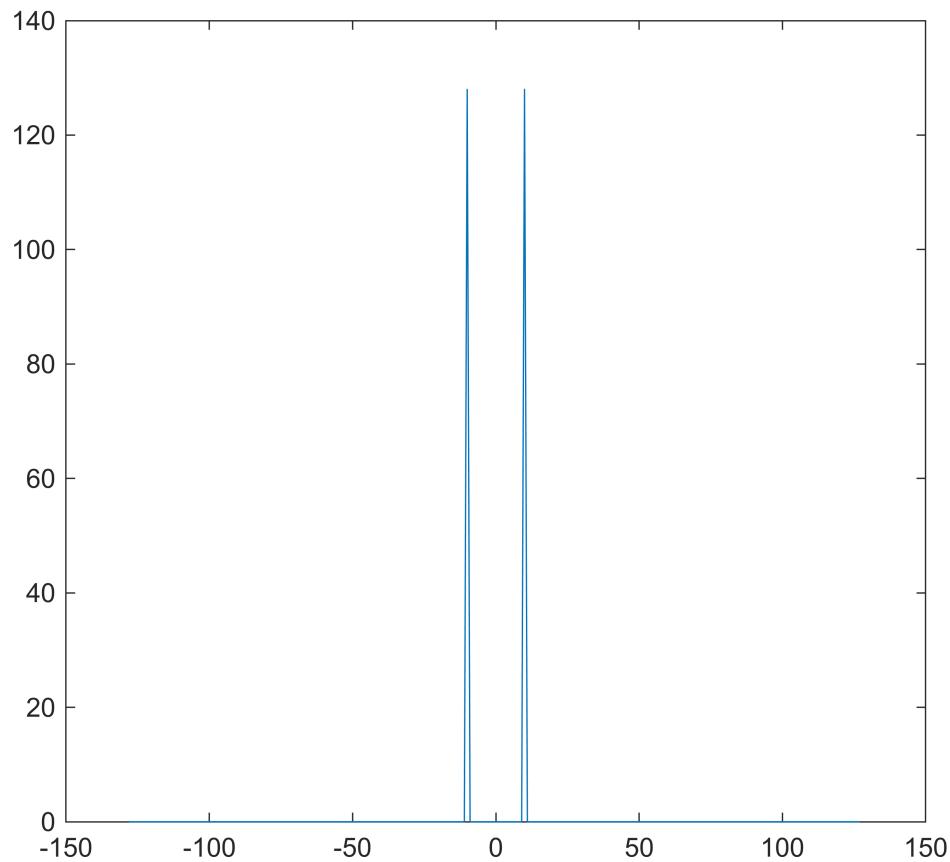
```
Y = 1x256 complex
10^2 ×
  0.0000 + 0.0000i  0.0000 + 0.0000i  -0.0000 + 0.0000i  -0.0000 - 0.0000i ...
```

```
Y=fftshift(Y)
```

```
Y = 1x256 complex
10^2 ×
```

```
-0.0000 + 0.0000i  0.0000 - 0.0000i  -0.0000 + 0.0000i  0.0000 - 0.0000i ...
```

```
plot(freq,abs(Y))
```



```
%改變訊號源
```

```
N=256
```

```
fs=8000
```

```
fs = 8000
```

```
freqStep=fs/N
```

```
freqStep = 31.2500
```

```
time=(0:N-1)/fs
```

```
time = 1x256
     0    0.0001    0.0003    0.0004    0.0005    0.0006    0.0008    0.0009 ...
```

```
y=0.5*sin(2*pi*50*time)+sin(2*pi*120*time)
```

```
y = 1x256
0    0.1137    0.2266    0.3378    0.4463    0.5515    0.6525    0.7486 ...
```

```
plot(time,y)
freq=(-N/2:N/2-1)
```

```
freq = 1x256
-128   -127   -126   -125   -124   -123   -122   -121   -120   -119   -118   -117   -116 ...
```

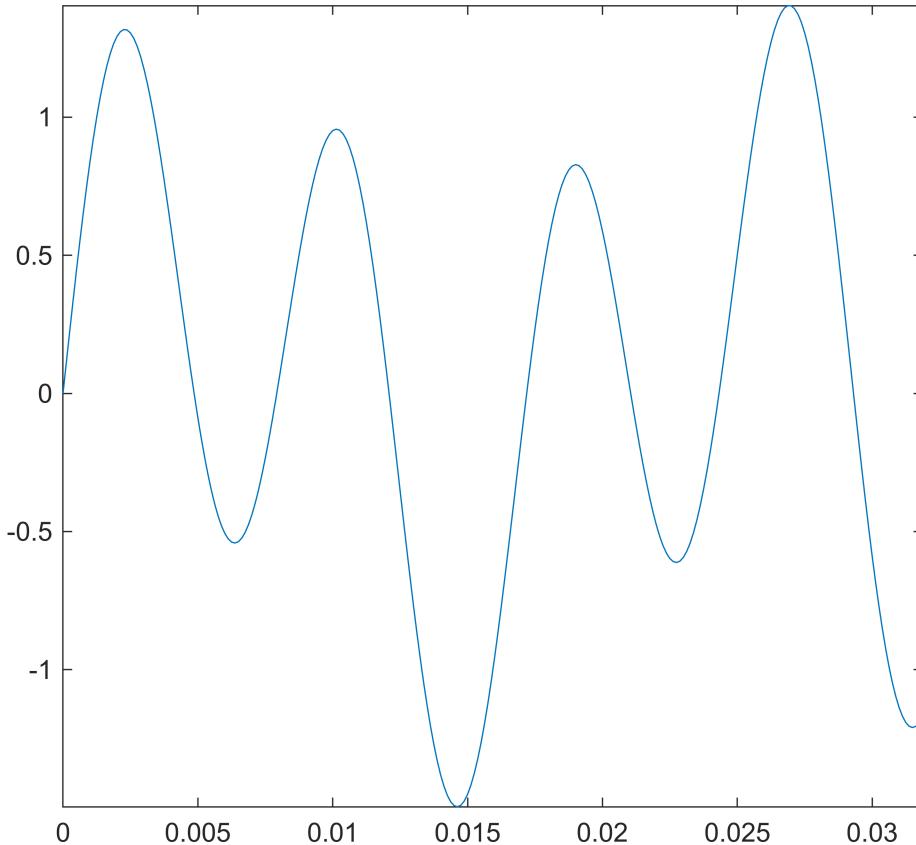
```
Y=fft(y)
```

```
Y = 1x256 complex
10^2 x
0.2852 + 0.0000i   0.4364 + 0.1018i   -0.3363 - 0.1022i   0.0405 + 0.1239i ...
```

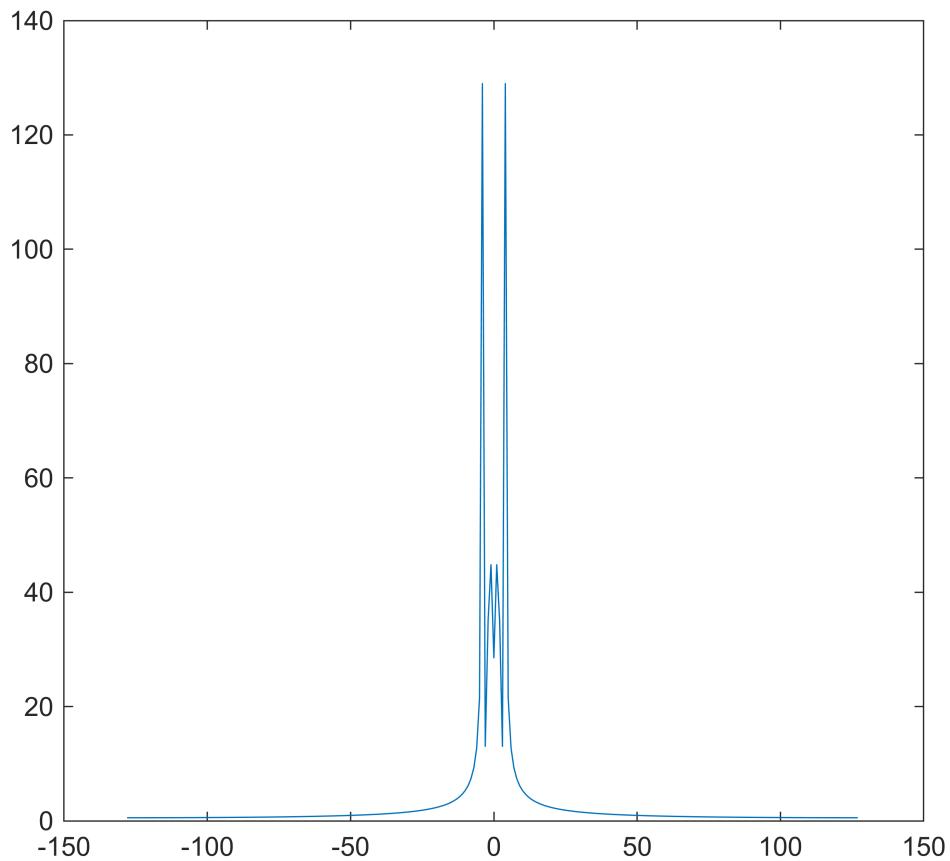
```
Y=fftshift(Y)
```

```
Y = 1x256 complex
10^2 x
0.0055 + 0.0000i   0.0055 + 0.0001i   0.0055 + 0.0001i   0.0055 + 0.0002i ...
```

```
axis("tight")
```



```
plot(freq,abs(Y))
```



%改變取樣頻率

N=256

N = 256

fs=1000

fs = 1000

freqStep=fs/N

freqStep = 3.9063

time=(0:N-1)/fs

time = 1x256

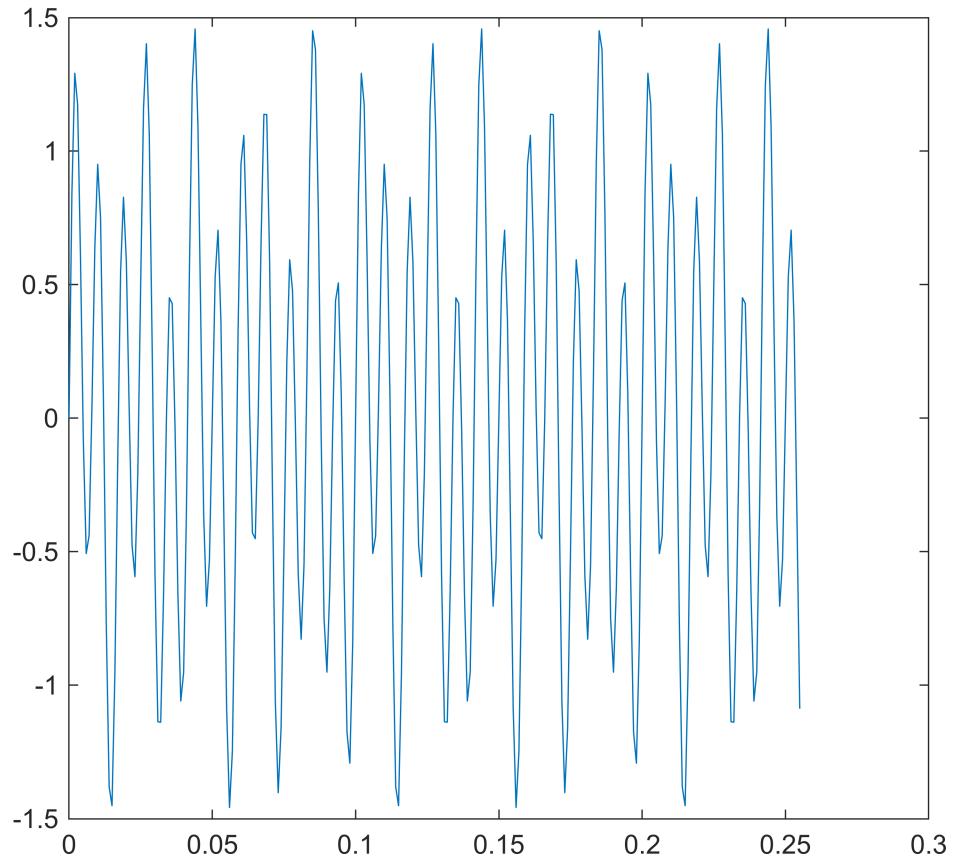
0 0.0010 0.0020 0.0030 0.0040 0.0050 0.0060 0.0070 ...

y=0.5*sin(2*pi*50*time)+sin(2*pi*120*time)

y = 1x256

0 0.8391 1.2919 1.1750 0.6009 -0.0878 -0.5068 -0.4398 ...

```
plot(time,y)
```



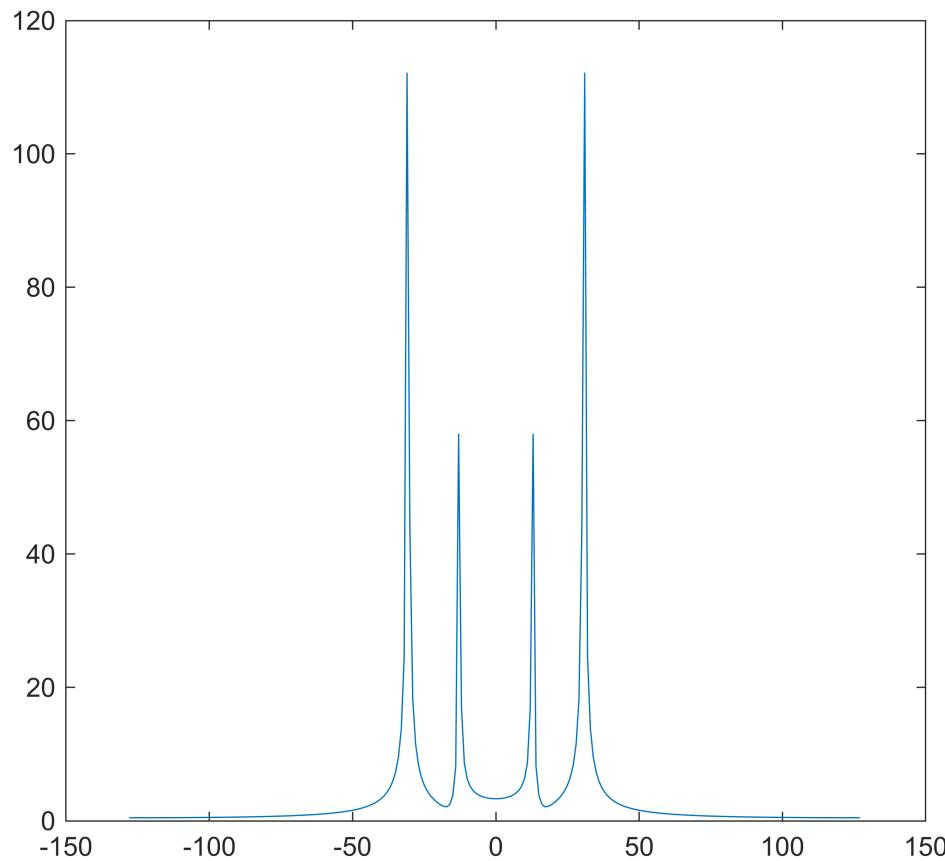
```
Y=fft(y)
```

```
Y = 1x256 complex
 3.3191 + 0.0000i  3.3275 + 0.1645i  3.3533 + 0.3337i  3.3981 + 0.5130i ...
```

```
Y=fftshift(Y)
```

```
Y = 1x256 complex
 0.4665 + 0.0000i  0.4664 + 0.0100i  0.4663 + 0.0199i  0.4661 + 0.0299i ...
```

```
plot(freq,abs(Y))
```



3. 讀取影像檔 L5_1.bmp

- 此圖形經過直方圖等化處理並比較差異。 (10分)
 - 將處理前後的灰階直方圖畫出並比較差異。 (10分)

```
image 1=imread("15_1.bmp")
```

```
image_1 = 511x512 uint8 matrix
 94 91 94 96 99 96 96 96 94 94 94 94 94 94 ...
 95 96 96 95 99 99 99 96 96 94 94 94 94 94
 93 94 93 93 94 96 96 96 95 96 94 94 94 94
 90 93 94 93 90 94 93 96 95 96 96 96 96 95
 93 93 94 94 94 94 94 94 95 96 94 94 94 94
 96 93 94 96 96 96 96 96 96 96 96 96 94 95
 99 96 94 93 93 95 93 94 94 94 93 93 93 94
 96 96 93 94 93 91 90 91 93 95 95 95 95 95
 94 91 94 96 96 96 96 95 96 96 99 99 99 103
 96 96 96 96 96 99 99 99 99 99 98 102 104
 .
 .
```

```
image_2=histeq(image_1)%%值方圖等化處理
```

```
image_2 = 511x512 uint8 matrix
231 223 231 235 239 235 235 235 235 231 231 231 231 231 ...
235 235 235 235 239 239 239 235 235 235 231 231 231 231
227 231 227 227 231 235 235 235 235 235 231 231 231 231
219 227 231 227 219 231 227 235 235 235 235 235 235 235
227 227 231 231 231 231 231 231 235 235 231 231 231 231
235 227 231 235 235 235 235 235 235 235 235 231 235 235
239 235 231 227 227 235 227 231 231 227 227 227 227 231
235 235 227 231 227 223 219 223 227 235 235 235 235 235
231 223 231 235 235 235 235 235 235 235 239 239 239 243
235 235 235 235 235 239 239 239 239 239 239 243 243
:
:
```

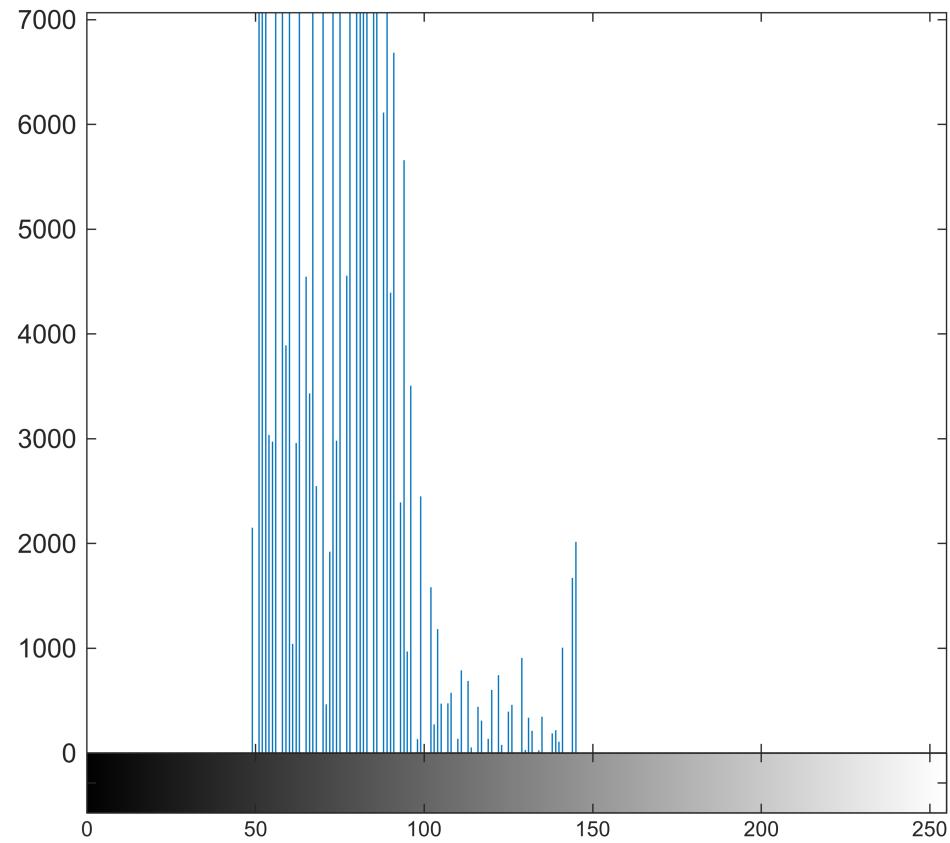
```
%%show 出圖
imshow(image_1)
```



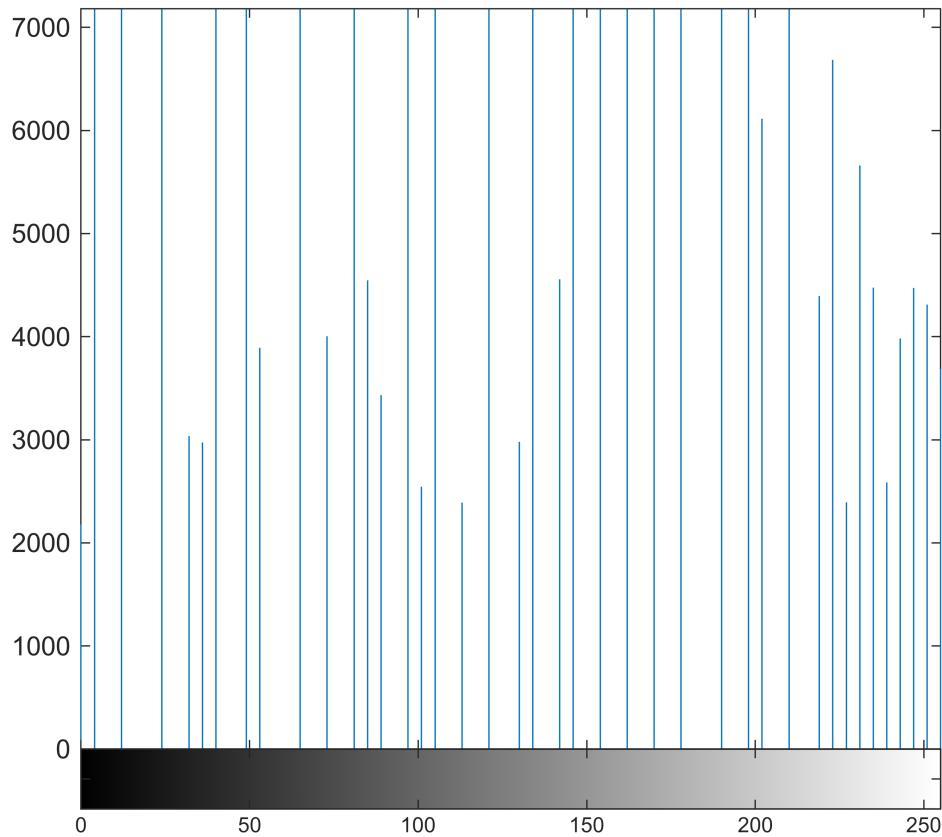
```
imshow(image_2)
```



```
%%秀出直方圖  
imhist(image_1)
```



```
imhist(image_2)
```



4. 讀取影像檔 L5_2.bmp

- 參考講義，將胡椒鹽式的雜訊加入原影像中並顯示出來。 (6 分)
- 以低通濾波器處理，顯示結果。 (6 分)
- 以中值濾波器處理，顯示結果並與低通進行比較。 (8 分)

```
image_3 = imread("L5_2.bmp")
```

```
image_3 = 554x500 uint8 matrix
202 201 201 201 202 202 204 204 204 205 203 202 202 ...
199 198 199 201 202 202 202 202 204 204 204 204 202 200
198 198 198 200 202 202 202 202 204 204 203 202 200 199
200 201 200 200 201 202 202 203 204 204 204 203 201 200
201 201 201 200 201 201 204 204 204 202 202 202 202 201
201 201 201 202 202 200 200 199 201 201 201 202 202 201
202 202 202 202 202 200 200 202 203 202 202 202 202 200
198 200 200 201 202 202 202 202 202 202 202 201 201 202
196 196 202 201 202 202 202 202 198 201 200 199 199 199
200 196 197 199 201 202 202 200 201 199 199 199 200
:
```

```
image_noisy=imnoise(image_3,"salt & pepper",0.06)
```

```
image_noisy = 554x500 uint8 matrix
 202 201 201 201 202 202 204 204 204 205 203 202 202 ...
 199 0 199 201 202 202 202 202 204 204 204 204 202 0
 198 198 198 200 202 202 202 202 204 204 203 202 200 199
 200 201 200 200 201 202 202 203 204 204 204 203 201 200
 201 201 201 200 201 201 204 204 204 202 202 202 201
 201 201 201 202 202 200 200 199 201 201 201 202 201
 202 202 202 202 200 200 202 203 0 202 202 202 200
 198 200 200 201 202 202 202 202 202 202 202 0 201 202
 196 196 202 201 202 255 202 202 198 201 201 200 199 199
 200 196 197 199 201 202 202 200 201 199 199 199 200
 .
.
```

```
image_2_noisy=double(image_noisy)/255
```

```
image_2_noisy = 554x500
 0.7922 0.7882 0.7882 0.7882 0.7922 0.7922 0.8000 0.8000 ...
 0.7804 0 0.7804 0.7882 0.7922 0.7922 0.7922 0.7922
 0.7765 0.7765 0.7765 0.7843 0.7922 0.7922 0.7922 0.7922
 0.7843 0.7882 0.7843 0.7843 0.7882 0.7922 0.7922 0.7961
 0.7882 0.7882 0.7882 0.7843 0.7882 0.7882 0.8000 0.8000
 0.7882 0.7882 0.7882 0.7922 0.7922 0.7843 0.7843 0.7804
 0.7922 0.7922 0.7922 0.7922 0.7922 0.7843 0.7843 0.7922
 0.7765 0.7843 0.7843 0.7882 0.7922 0.7922 0.7922 0.7922
 0.7686 0.7686 0.7922 0.7882 0.7922 1.0000 0.7922 0.7922
 0.7843 0.7686 0.7725 0.7804 0.7882 0.7922 0.7922 0.7843
 .
.
```

```
h=[1/9 1/9 1/9; 1/9 1/9 1/9 ;1/9 1/9 1/9]
```

```
h = 3x3
 0.1111 0.1111 0.1111
 0.1111 0.1111 0.1111
 0.1111 0.1111 0.1111
```

```
image_low=filter2(h,image_2_noisy)
```

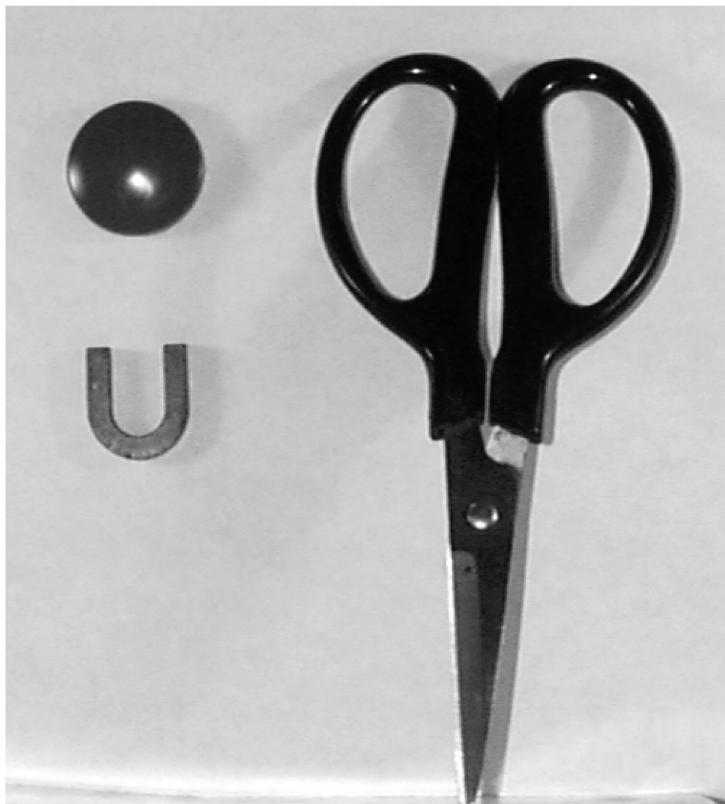
```
image_low = 554x500
 0.2623 0.4366 0.4370 0.5255 0.5272 0.5290 0.5298 0.5316 ...
 0.4349 0.6954 0.6967 0.7869 0.7904 0.7930 0.7939 0.7965
 0.4340 0.6941 0.6959 0.7856 0.7895 0.7917 0.7926 0.7952
 0.5224 0.7834 0.7839 0.7856 0.7882 0.7917 0.7939 0.7969
 0.5251 0.7874 0.7874 0.7878 0.7882 0.7900 0.7908 0.7935
 0.5264 0.7895 0.7895 0.7900 0.7887 0.7887 0.7887 0.7917
 0.5246 0.7874 0.7891 0.7904 0.7900 0.7887 0.7874 0.7891
 0.5203 0.7834 0.7869 0.7904 0.8135 0.8135 0.8135 0.7900
 0.5168 0.7778 0.7808 0.7865 0.8126 0.8148 0.8144 0.7891
 0.5181 0.7786 0.7795 0.7852 0.8118 0.8148 0.8144 0.7887
 .
.
```

```
image_med=medfilt2(image_noisy,[3 3])
```

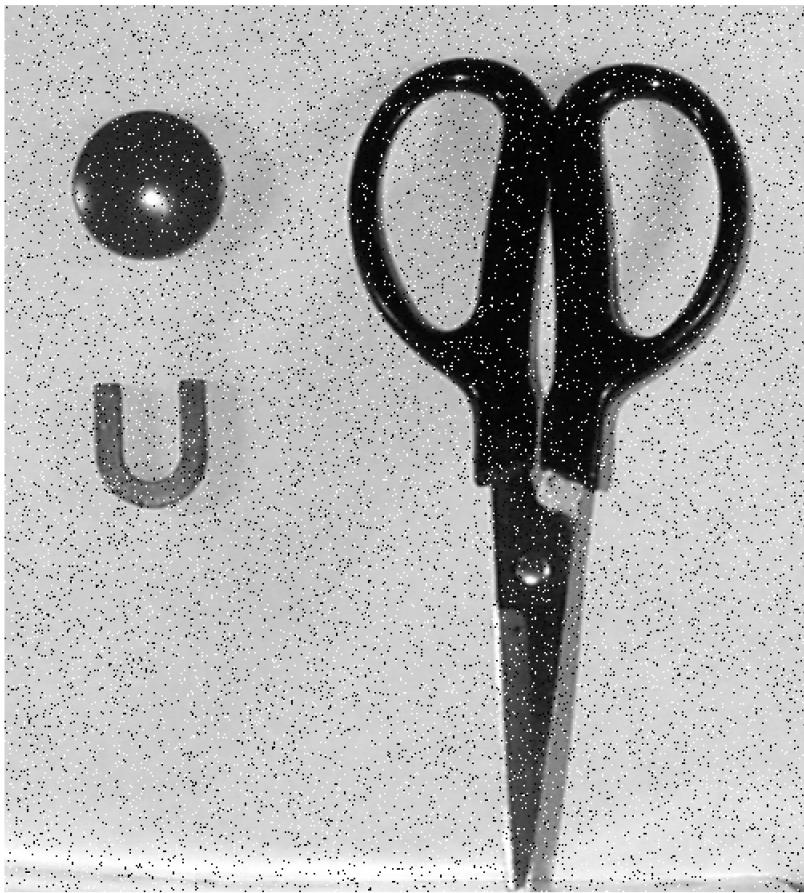
```
image_med = 554x500 uint8 matrix
 0 199 199 201 201 202 202 202 204 204 204 202 202 201 ...
 198 199 200 201 202 202 202 204 204 204 204 203 202 201
 198 199 200 200 202 202 202 202 204 204 204 203 201 200
 198 200 200 200 201 202 202 204 204 203 204 202 201 200
```

```
201 201 201 201 201 201 202 203 203 202 202 202 202 201  
201 201 201 202 201 200 202 202 202 202 202 202 202 201  
200 201 201 202 202 202 200 202 202 202 202 202 202 201  
196 200 201 202 202 202 202 202 202 201 201 201 200 199  
196 198 200 201 202 202 202 202 201 200 199 199 199 199  
196 199 199 201 202 202 202 201 201 201 200 199 199 199  
.
```

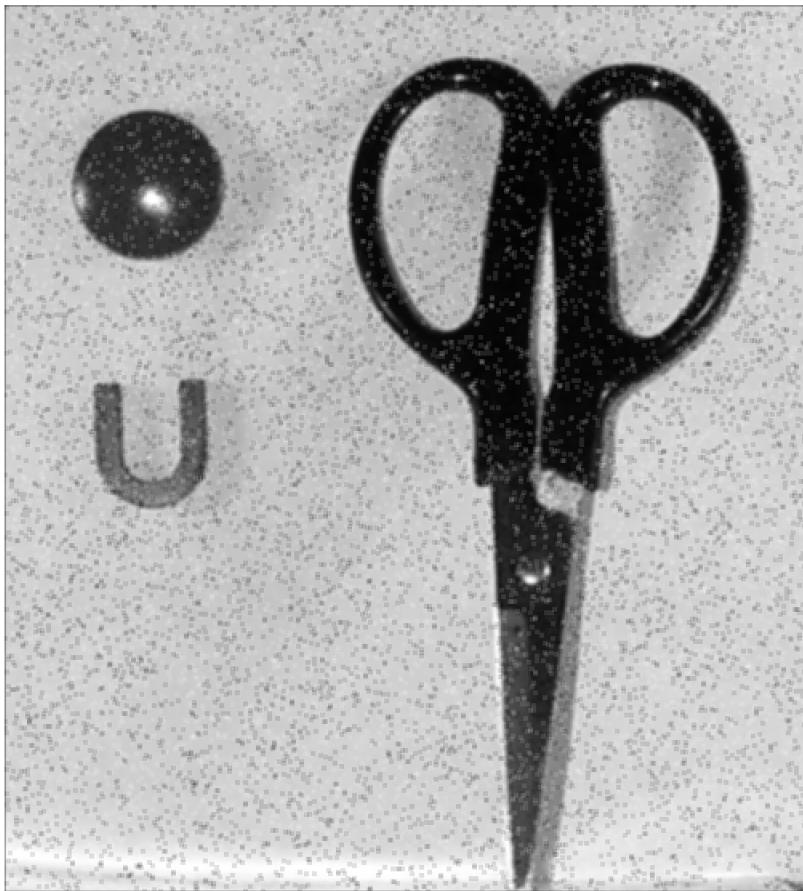
```
imshow(image_3)
```



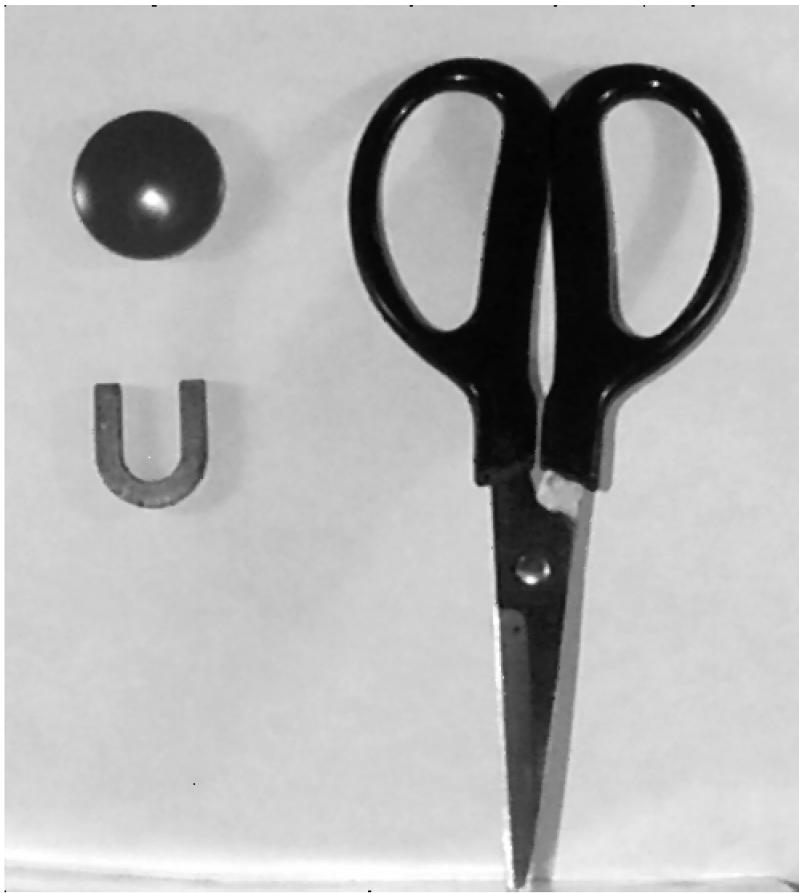
```
figure,imshow(image_noisy)
```



```
figure,imshow(image_low)
```



```
figure,imshow(image_med)
```



%%中濾通波比較乾淨

5.重做 Q4 低通濾波器處理

- 將遮罩大小改為 5*5，並與遮罩大小為 3*3 的結果做比較。 (10 分)
- 將所加雜訊改為高斯雜訊並與胡椒鹽式雜訊的結果做比較。 (10 分)

```
image_3 = imread("15_2.bmp")
```

```
image_3 = 554x500 uint8 matrix
 202  201  201  201  202  202  204  204  204  205  203  202  202 ...
 199  198  199  201  202  202  202  202  204  204  204  204  202  200
 198  198  198  200  202  202  202  202  204  203  202  202  200  199
 200  201  200  200  201  202  202  203  204  204  203  201  201  200
 201  201  201  200  201  201  204  204  204  204  203  202  202  201
 201  201  201  202  202  200  200  199  201  201  201  202  202  201
 202  202  202  202  202  200  200  202  203  202  202  202  202  200
 198  200  200  201  202  202  202  202  202  202  202  201  201  202
 196  196  202  201  202  202  202  202  198  201  201  200  199  199
 200  196  197  199  201  202  202  200  201  199  199  199  200
```

```
image_noisy=imnoise(image_3,"salt & pepper",0.06)
```

```
image_noisy = 554x500 uint8 matrix
 202 201 201 201 202 202 204 204 204 205 203 202 202 ...
 199 198 199 201 202 202 202 202 204 204 204 202 202 200
 198 198 198 200 202 202 202 202 204 203 203 202 200 0
 200 201 200 200 201 202 202 203 204 204 204 203 201 255
 201 201 201 200 255 201 204 204 204 202 202 202 201
 201 201 201 202 202 200 199 201 201 201 202 202 201
 202 202 202 255 202 200 200 202 203 202 202 202 200
 198 200 200 201 202 202 202 202 202 202 201 201 201 202
 196 196 202 201 202 202 202 202 198 201 201 200 255 199
 200 196 197 199 201 202 202 200 201 199 199 199 200
  ...
  :
```

```
image_2_noisy=double(image_noisy)/255
```

```
image_2_noisy = 554x500
 0.7922 0.7882 0.7882 0.7882 0.7922 0.7922 0.8000 0.8000 ...
 0.7804 0.7765 0.7804 0.7882 0.7922 0.7922 0.7922 0.7922
 0.7765 0.7765 0.7765 0.7843 0.7922 0.7922 0.7922 0.7922
 0.7843 0.7882 0.7843 0.7843 0.7882 0.7922 0.7922 0.7961
 0.7882 0.7882 0.7882 0.7843 1.0000 0.7882 0.8000 0.8000
 0.7882 0.7882 0.7882 0.7922 0.7922 0.7843 0.7843 0.7804
 0.7922 0.7922 0.7922 1.0000 0.7922 0.7843 0.7843 0.7922
 0.7765 0.7843 0.7843 0.7882 0.7922 0.7922 0.7922 0.7922
 0.7686 0.7686 0.7922 0.7882 0.7922 0.7922 0.7922 0.7922
 0.7843 0.7686 0.7725 0.7804 0.7882 0.7922 0.7922 0.7843
  ...
  :
```

```
h=[1/9 1/9 1/9; 1/9 1/9 1/9 ;1/9 1/9 1/9]
```

```
h = 3x3
 0.1111 0.1111 0.1111
 0.1111 0.1111 0.1111
 0.1111 0.1111 0.1111
```

```
h1=[1/25 1/25 1/25 1/25 1/25;1/25 1/25 1/25 1/25 1/25; 1/25 1/25 1/25 1/25 1/25; 1/25 1/25 1/25 1/25 1/25]
```

```
h1 = 5x5
 0.0400 0.0400 0.0400 0.0400 0.0400
 0.0400 0.0400 0.0400 0.0400 0.0400
 0.0400 0.0400 0.0400 0.0400 0.0400
 0.0400 0.0400 0.0400 0.0400 0.0400
 0.0400 0.0400 0.0400 0.0400 0.0400
```

```
image_low=filter2(h,image_2_noisy)
```

```
image_low = 554x500
 0.3486 0.5229 0.5233 0.5255 0.5272 0.5290 0.5298 0.5316 ...
 0.5211 0.7817 0.7830 0.7869 0.7904 0.7930 0.7939 0.7965
 0.5203 0.7804 0.7821 0.7856 0.7895 0.7917 0.7926 0.7952
 0.5224 0.7834 0.7839 0.8092 0.8118 0.8153 0.7939 0.7969
 0.5251 0.7874 0.7874 0.8113 0.8118 0.8135 0.7908 0.7935
 0.5264 0.7895 0.8126 0.8366 0.8353 0.8122 0.7887 0.7917
 0.5246 0.7874 0.8122 0.8135 0.8131 0.7887 0.7874 0.7891
 0.5203 0.7834 0.8100 0.8135 0.8135 0.7904 0.7904 0.7900
```

```
0.5168    0.7778    0.7808    0.7865    0.7895    0.7917    0.7913    0.7891
```

```
0.5181    0.7786    0.7795    0.7852    0.7887    0.7917    0.7913    0.7887
```

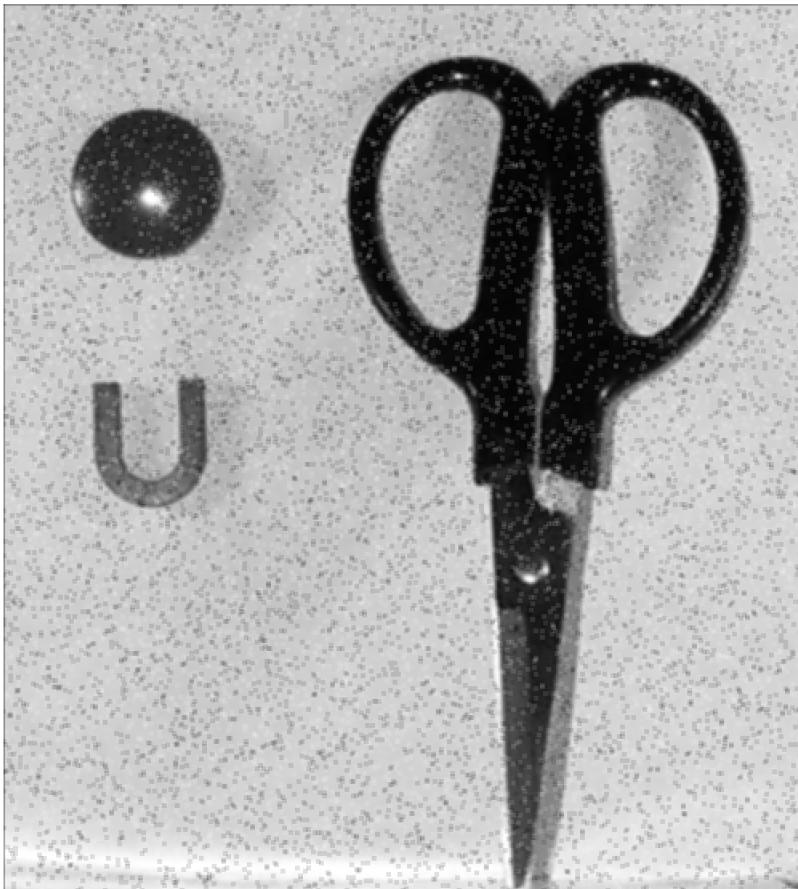
```
.
```

```
image_low2=filter2(h1,image_2_noisy)
```

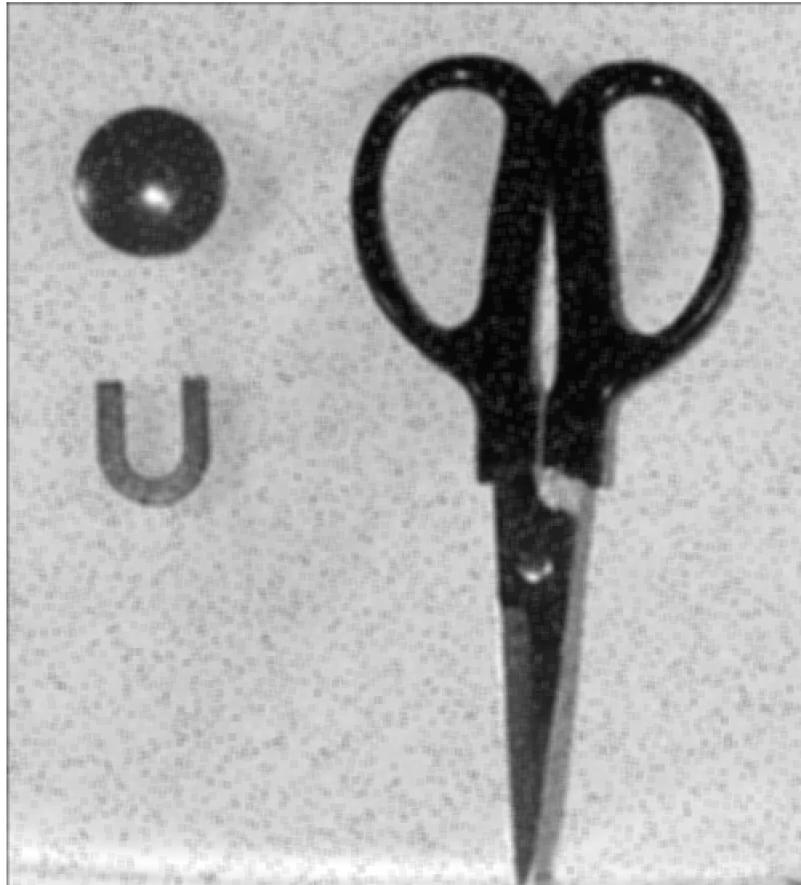
```
image_low2 = 554x500
0.2814    0.3758    0.4709    0.4720    0.4737    0.4753    0.4769    0.4778    ...
0.3757    0.5015    0.6281    0.6295    0.6314    0.6334    0.6356    0.6370
0.4703    0.6275    0.7940    0.7955    0.7978    0.8003    0.8031    0.7962
0.4701    0.6275    0.7940    0.7953    0.7970    0.7987    0.8009    0.7934
0.4717    0.6375    0.8041    0.8045    0.8053    0.8066    0.8002    0.7923
0.4723    0.6383    0.8049    0.8053    0.8058    0.8067    0.7998    0.7918
0.4712    0.6373    0.8041    0.8052    0.8064    0.8069    0.7989    0.7903
0.4696    0.6356    0.7939    0.7953    0.7970    0.7975    0.7892    0.7885
0.4692    0.6347    0.7929    0.7947    0.7969    0.7981    0.7903    0.7898
0.4690    0.6257    0.7838    0.7944    0.8050    0.8067    0.8075    0.8072
.
```

```
.
```

```
figure,imshow(image_low)
```



```
figure,imshow(image_low2)
```



%%遮罩越大雜訊變少 也相對有可能損失清晰度

```
image_3=imread("15_2.bmp")
```

```
image_3 = 554x500 uint8 matrix
 202 201 201 201 202 202 204 204 204 205 203 202 202 ...
 199 198 199 201 202 202 202 202 204 204 204 202 200
 198 198 198 200 202 202 202 202 204 203 202 200 199
 200 201 200 200 201 202 202 203 204 204 203 201 200
 201 201 201 200 201 201 204 204 204 202 202 202 201
 201 201 201 202 202 200 200 199 201 201 202 202 201
 202 202 202 202 202 200 200 202 203 202 202 202 200
 198 200 200 201 202 202 202 202 202 202 201 201 202
 196 196 202 201 202 202 202 202 198 201 200 199 199
 200 196 197 199 201 202 202 200 201 199 199 199 200
 :
```

```
image_noisy=imnoise(image_3,"gaussian",0.06)
```

```

image_noisy = 554x500 uint8 matrix
  208   211   243   191   241   180   204   204   169   199   200   255   169 ...
  215   207   224   175   199   255   225   237   194   214   226   186   237
  206   213   187   220   216   206   192   200   204   213   174   187   211
  205   254   186   162   253   214   211   199   147   205   246   226   189
  194   255   232   211   179   207   223   221   215   233   255   199   238
  227   206   171   255   215   255   183   245   255   241   233   169   205
  255   208   209   216   196   225   255   255   192   255   188   208   229
  216   146   247   239   201   219   241   208   248   229   199   236   210
  204   199   174   195   212   209   208   178   231   213   225   187   225
  240   230   203   180   231   229   219   210   209   230   211   207   206
  :

```

```
image_2_noisy=double(image_noisy)/255
```

```

image_2_noisy = 554x500
  0.8157   0.8275   0.9529   0.7490   0.9451   0.7059   0.8000   0.8000 ...
  0.8431   0.8118   0.8784   0.6863   0.7804   1.0000   0.8824   0.9294
  0.8078   0.8353   0.7333   0.8627   0.8471   0.8078   0.7529   0.7843
  0.8039   0.9961   0.7294   0.6353   0.9922   0.8392   0.8275   0.7804
  0.7608   1.0000   0.9098   0.8275   0.7020   0.8118   0.8745   0.8667
  0.8902   0.8078   0.6706   1.0000   0.8431   1.0000   0.7176   0.9608
  1.0000   0.8157   0.8196   0.8471   0.7686   0.8824   1.0000   1.0000
  0.8471   0.5725   0.9686   0.9373   0.7882   0.8588   0.9451   0.8157
  0.8000   0.7804   0.6824   0.7647   0.8314   0.8196   0.8157   0.6980
  0.9412   0.9020   0.7961   0.7059   0.9059   0.8980   0.8588   0.8235
  :

```

```
image_low=filter2(h,image_2_noisy)
```

```

image_low = 554x500
  0.3664   0.5699   0.5451   0.5547   0.5407   0.5682   0.5686   0.5373 ...
  0.5490   0.8340   0.8153   0.8261   0.8205   0.8357   0.8292   0.7969
  0.5664   0.8266   0.7965   0.7939   0.8279   0.8588   0.8449   0.7882
  0.5782   0.8418   0.8366   0.8044   0.8139   0.8283   0.8161   0.7895
  0.5843   0.8410   0.8418   0.8122   0.8501   0.8453   0.8532   0.8275
  0.5861   0.8527   0.8553   0.8209   0.8536   0.8444   0.9015   0.8906
  0.5481   0.8214   0.8266   0.8492   0.8806   0.8671   0.9089   0.9072
  0.5351   0.8096   0.7987   0.8231   0.8331   0.8566   0.8706   0.8784
  0.5381   0.8100   0.7900   0.8200   0.8344   0.8580   0.8370   0.8505
  0.5743   0.8275   0.7987   0.8083   0.8479   0.8719   0.8484   0.8357
  :

```

```
image_med=medfilt2(image_noisy,[3 3])
```

```

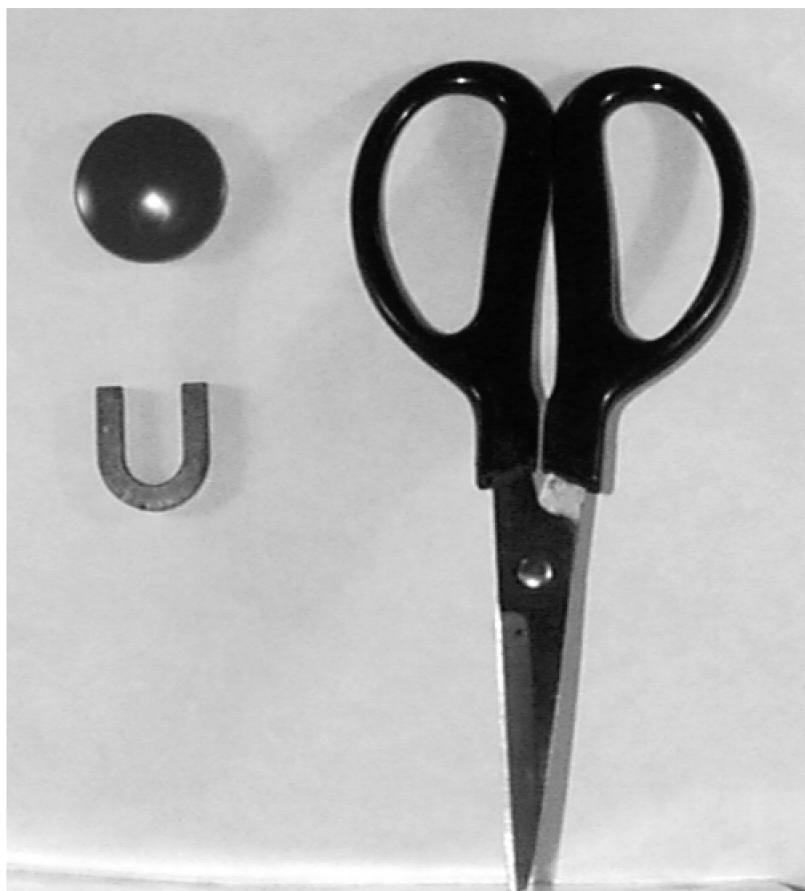
image_med = 554x500 uint8 matrix
  0   208   191   191   180   199   204   194   194   194   199   186   183 ...
  207   211   211   216   206   206   204   204   204   200   200   200   200
  206   207   207   199   214   214   211   200   204   205   213   211   200
  205   206   213   211   211   211   207   204   205   213   213   211   200
  205   206   211   211   214   214   214   215   221   233   233   226   199
  206   209   211   211   215   215   225   223   241   233   233   208   199
  206   209   209   215   219   219   241   245   245   233   229   208   205
  199   208   208   209   212   212   219   231   229   225   213   210   210
  199   204   199   203   212   219   210   210   213   225   213   210   210
  204   204   203   204   212   219   210   210   213   211   211   207
  :

```

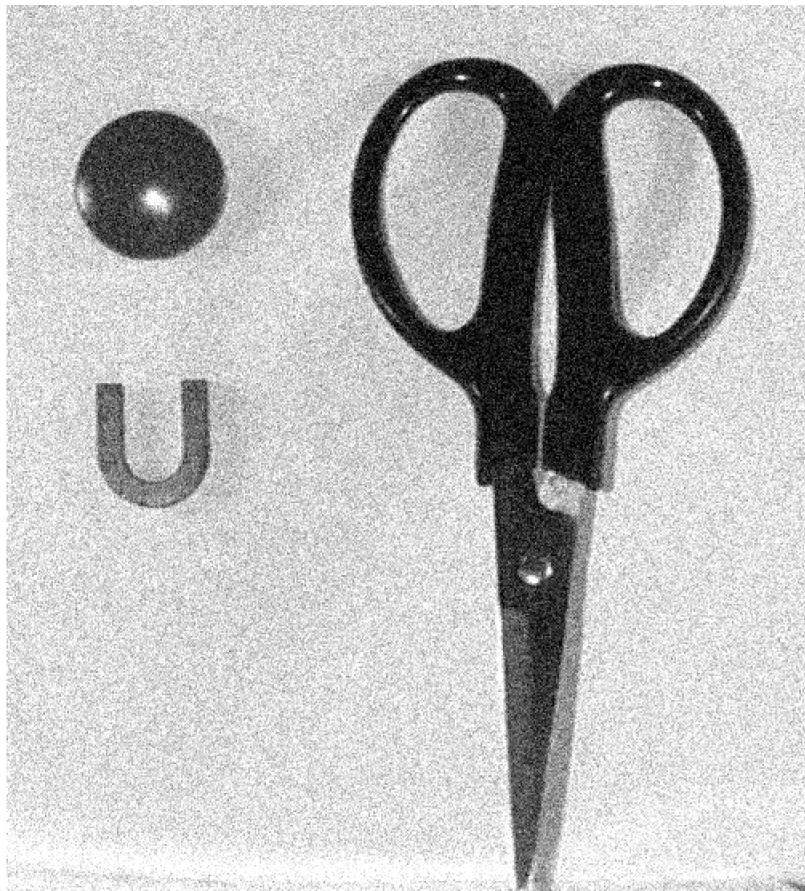
```
image_med2=medfilt2(image_noisy,[5 5])
```

```
image_med2 = 554x500 uint8 matrix
 0   0   191   187   187   191   192   192   192   186   174   183   183 ...
  0   187   206   206   199   200   200   200   200   199   194   200   200
194   206   211   211   211   207   206   206   205   204   205   209   208
187   206   211   213   211   214   214   214   215   214   213   209   205
187   206   211   213   211   215   214   214   215   213   211   208   205
186   208   211   214   215   216   215   223   229   226   226   209   208
174   206   209   209   212   215   215   225   229   225   225   213   208
174   204   209   209   215   216   219   229   225   213   213   211   207
199   204   208   209   212   216   219   219   213   210   211   211   210
177   201   204   212   210   212   212   213   211   211   213   223   211
  :
:
```

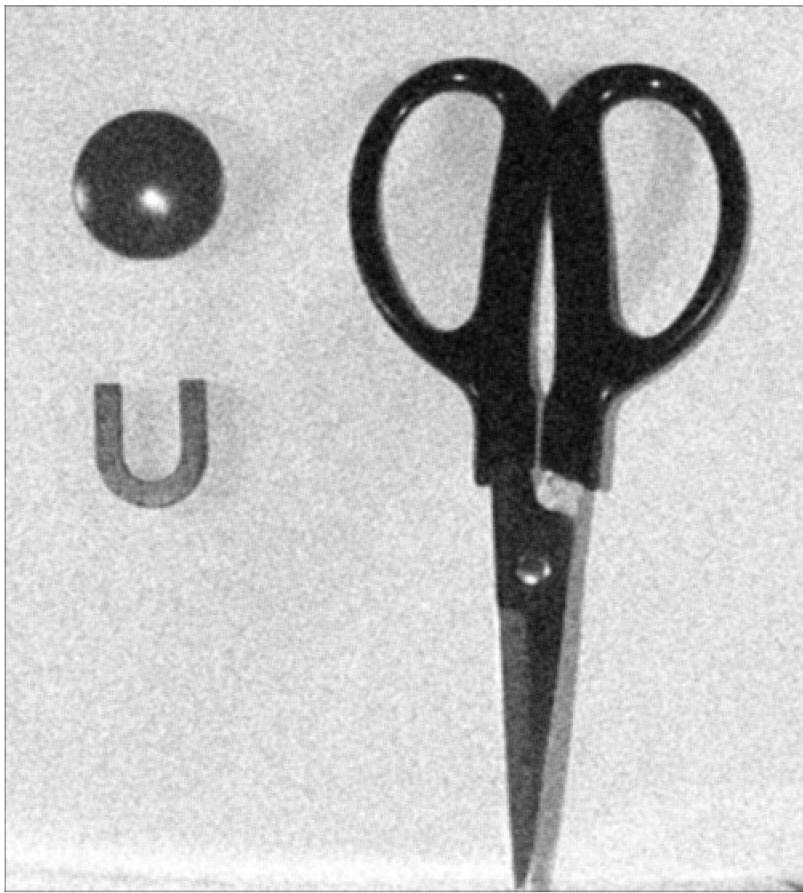
```
imshow(image_3)
```



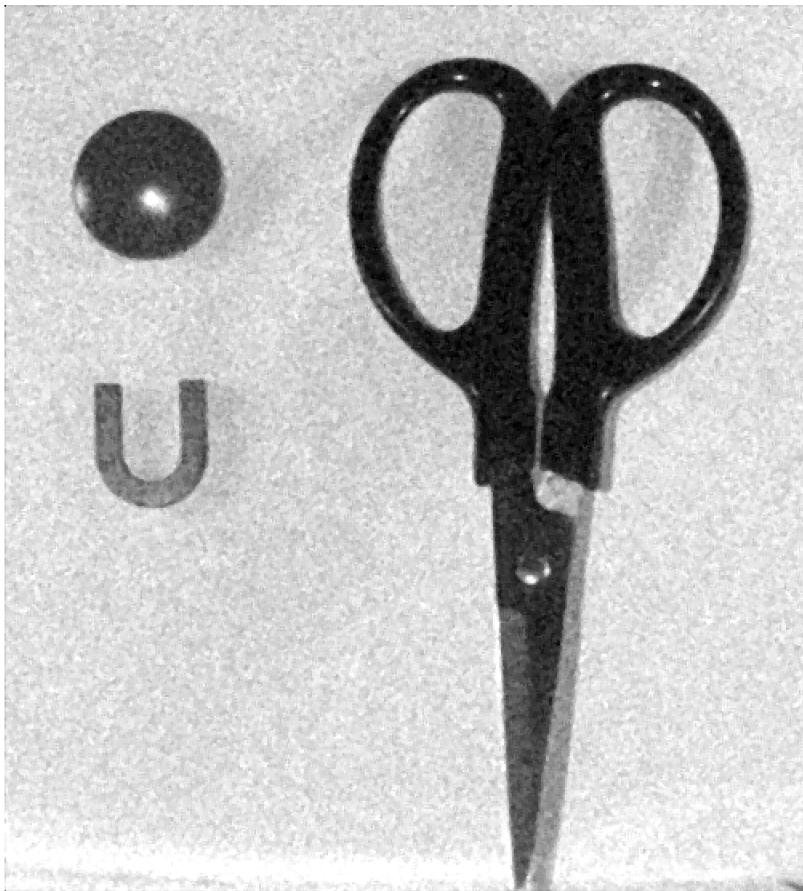
```
figure,imshow(image_noisy)
```



```
figure,imshow(image_low)
```



```
figure,imshow(image_med)
```



%%高斯雜訊顆粒感更重