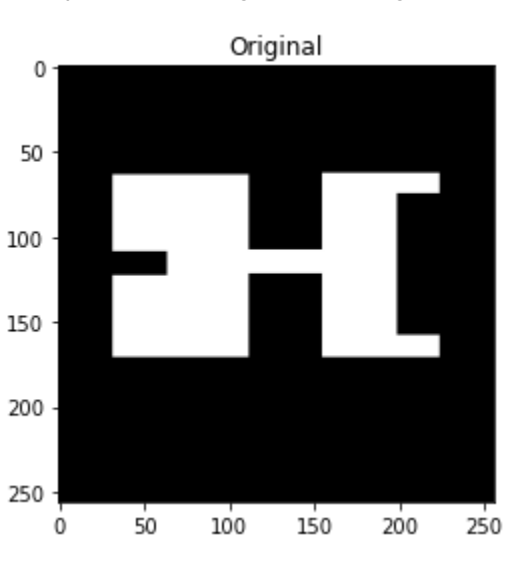
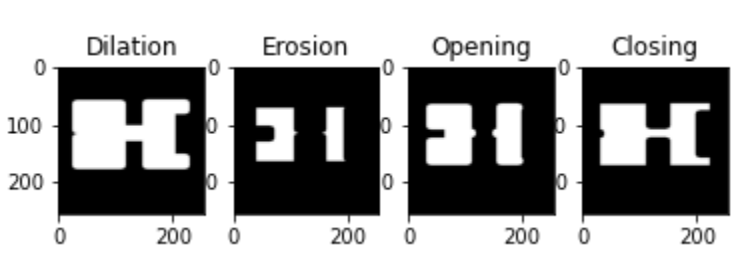
실습 1

In [1]:
```python
import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv

img0 = cv.imread('pattern.png')
kernel0 = cv.getStructuringElement(cv.MORPH_ELLIPSE, (15,15))
dil0 = cv.dilate(img0, kernel0)
ero0 = cv.erode(img0, kernel0)
ope0 = cv.morphologyEx(img0, cv.MORPH_OPEN, kernel0)
clo0 = cv.morphologyEx(img0, cv.MORPH_CLOSE, kernel0)

plt.title('Original')
plt.imshow(img0)
plt.figure()
plt.subplot(1,4,1)
plt.title('Dilation')
plt.imshow(dil0)
plt.subplot(1,4,2)
plt.title('Erosion')
plt.imshow(ero0)
plt.subplot(1,4,3)
plt.title('Opening')
plt.imshow(ope0)
plt.subplot(1,4,4)
plt.title('Closing')
plt.imshow(clo0)
```

Out[1]: <matplotlib.image.AxesImage at 0x18b5221dee0>



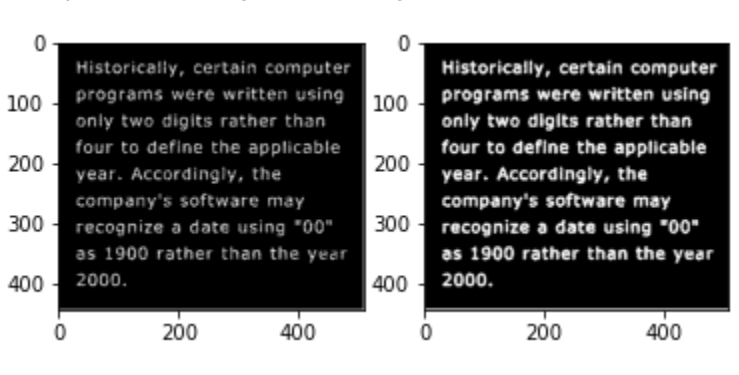


In [2]:
```python
#침식 Erosion
#팽창 Dilation
#열기 Opening -> 침식 후 팽창
#닫기 Closing -> 팽창 후 침식

#Structuring element(구조 요소) 생성 함수
#kernel = cv.getStructuringElement(shape, ksize, anchor)
#shape = 형태 지정, 사각형(cv.MORPH_RECT), 십자형(cv.MORPH_CROSS), 타원형(cv.MORPTH_ELLIPSE)
#ksize = 구조요소의 크기 지정 (3,3), (5,5) emd
#anchor = 구조요소의 원점 위치

#침식 함수 = erosion = cv.erode(img, kernel, iterations = 1)
#팽창 함수 = dilation = cv.dilate(img, kernel, iterations = 1)
#열기 함수 = opening = cv.morphologyEx(img, cv.MORPH_OPEN, kernel)
#닫기 함수 = closing = cv.morphologyEx(img, cv.MORPH_CLOSE, kernel)
```

실습2

In [3]:
```python
img2 = cv.imread('0907.png')
kernel = cv.getStructuringElement(cv.MORPH_CROSS, (3,3))
#팽창
dil = cv.dilate(img2, kernel)
plt.subplot(1,2,1)
plt.imshow(img2)
plt.subplot(1,2,2)
plt.imshow(dil)
```

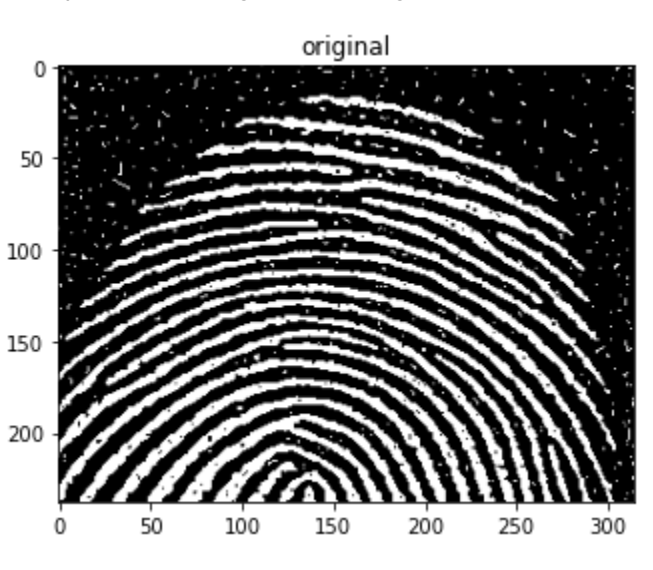
Out[3]: <matplotlib.image.AxesImage at 0x18b52369760>

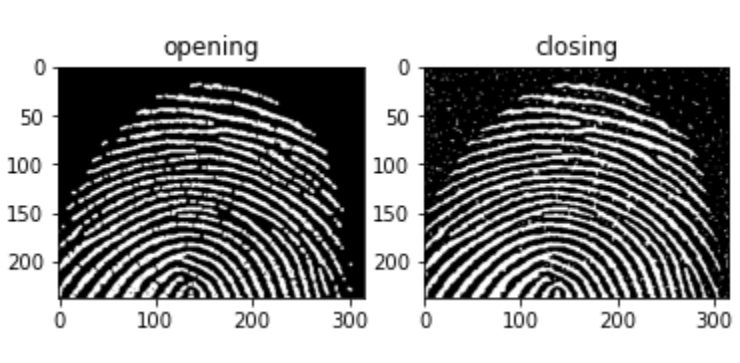


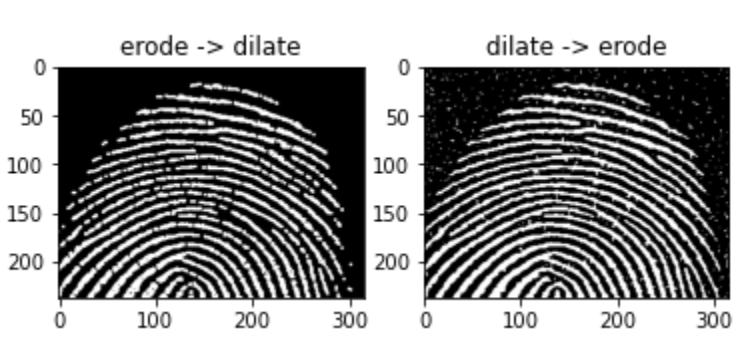
실습3

In [4]:
```python
img3 = cv.imread('0911.png')
kernel2 = cv.getStructuringElement(cv.MORPH_RECT, (3,3))
#팽창
dil2 = cv.dilate(img3, kernel2)
#침식
ero = cv.erode(img3, kernel2)
#열기
ope = cv.morphologyEx(img3, cv.MORPH_OPEN, kernel2)
#닫기
clo = cv.morphologyEx(img3, cv.MORPH_CLOSE, kernel2)

#침식후 팽창(열기)
ope2 = cv.dilate(ero, kernel2)
#팽창후 침식(닫기)
clo2 = cv.erode(dil2, kernel2)

plt.subplot(1,1,1)
plt.title('original')
plt.imshow(img3)
plt.figure()
plt.subplot(1,2,1)
plt.title('opening')
plt.imshow(ope)
plt.subplot(1,2,2)
plt.title('closing')
plt.imshow(clo)
plt.figure()
plt.subplot(1,2,1)
plt.title('erode -> dilate')
plt.imshow(ope2)
plt.subplot(1,2,2)
plt.title('dilate -> erode')
plt.imshow(clo2)
```

Out[4]: <matplotlib.image.AxesImage at 0x18b5349bcd0>



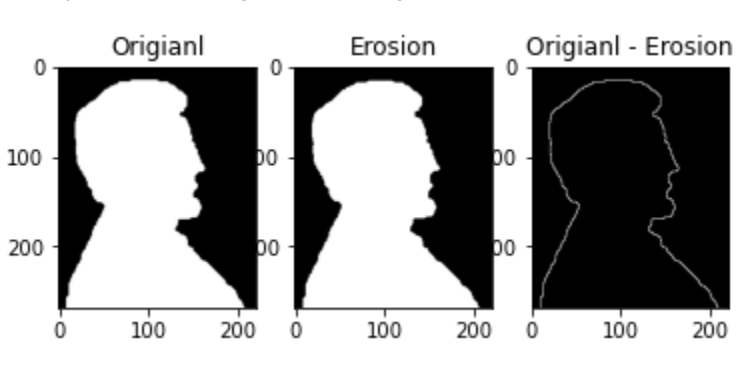




실습 4

In [5]:
```python
img4 = cv.imread('0914.png')
kernel3 = cv.getStructuringElement(cv.MORPH_RECT, (3,3))
ero2 = cv.erode(img4, kernel3)
ero3 = img4 - ero2

plt.subplot(1,3,1)
plt.title('Origianl')
plt.imshow(img4)
plt.subplot(1,3,2)
plt.title('Erosion')
plt.imshow(ero2)
plt.subplot(1,3,3)
plt.title('Origianl - Erosion')
plt.imshow(ero3)
```

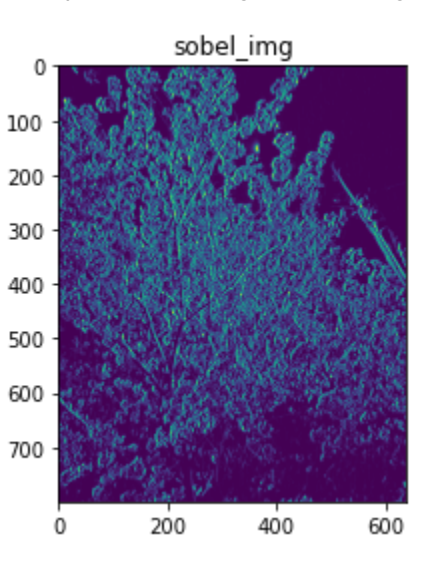
Out[5]: <matplotlib.image.AxesImage at 0x18b535bc760>

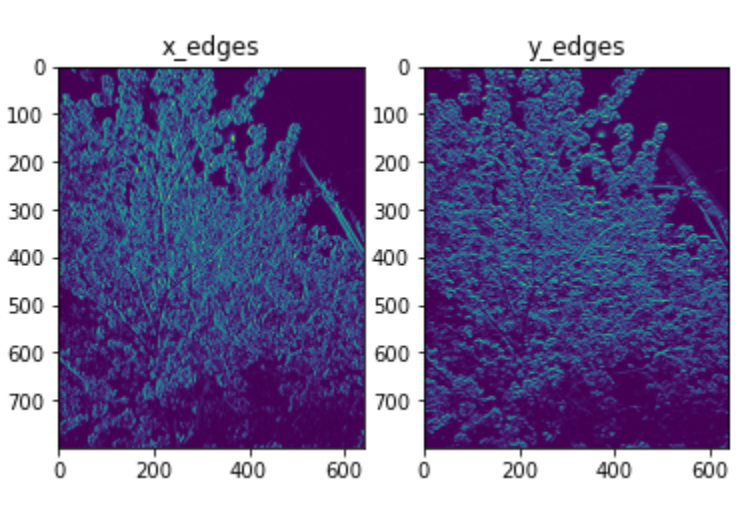


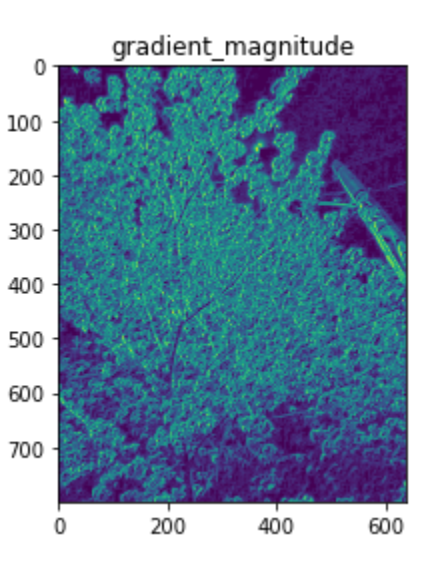
과제1

In [6]:
```python
IMG = cv.imread('g1.jpg')
IMG_gray = cv.cvtColor(IMG, cv.COLOR_BGR2GRAY)
sob_img = cv.Sobel(IMG_gray, cv.CV_8U,1,0)
x_edges = cv.Sobel(IMG_gray, -1, 1, 0, 5)
y_edges = cv.Sobel(IMG_gray, -1, 0, 1, 5)

scr = x_edges^2 + y_edges^2
gradient_magnitude = scr**(1/2)

plt.imshow(sob_img)
plt.title('sobel_img')
plt.figure()
plt.subplot(1,2,1)
plt.title('x_edges')
plt.imshow(x_edges)
plt.subplot(1,2,2)
plt.title('y_edges')
plt.imshow(y_edges)
plt.figure()
plt.title('gradient_magnitude')
plt.imshow(gradient_magnitude)
```

Out[6]: <matplotlib.image.AxesImage at 0x18b536c5880>

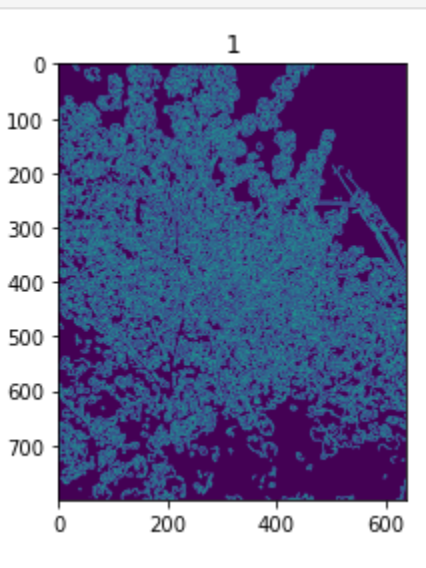


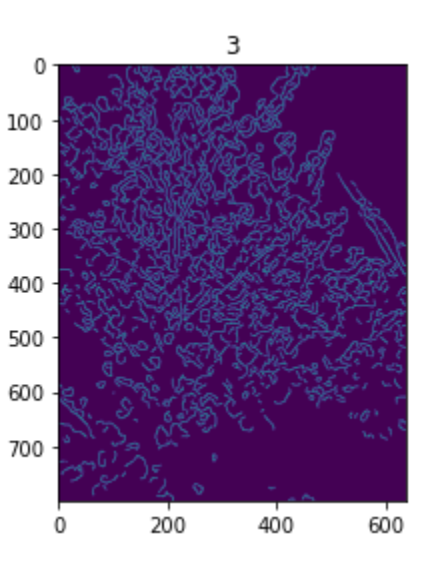


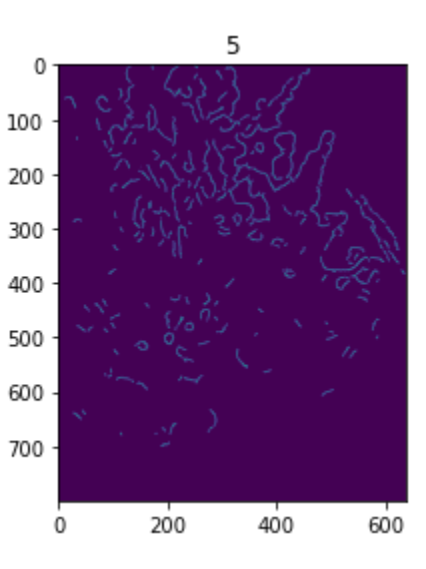


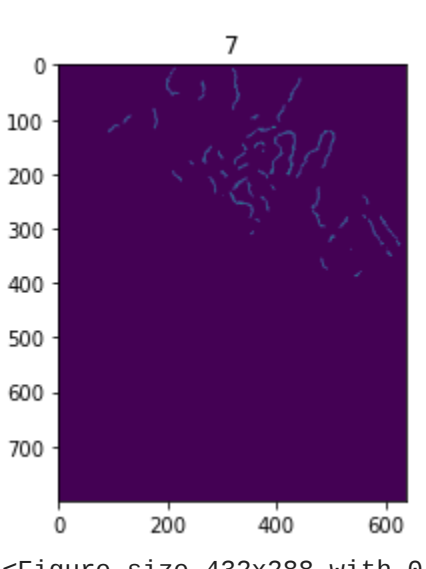
과제2

In [7]:
```python
sigma = [1,3,5,7]
for i in range(len(sigma)):
    w = 6 * sigma[i] + 1
    blur_img = cv.GaussianBlur(IMG_gray, (w,w), sigma[i])
    canny_edges = cv.Canny(blur_img, 50, 70)
    plt.imshow(canny_edges)
    plt.title(sigma[i])
    plt.figure()
```









<Figure size 432x288 with 0 Axes>

과제3

30~60 사이는 원을 기준으로 오른쪽이 0도부터 시작하는지 왼쪽에서부터 시작하는지에 따라 30도와 60도 사이의 위치가 다릅니다.

In [8]:
```python
import math

IMG2 = cv.imread('g2.jpg')

cv.circle(IMG2, (128,128), 80, (200,200,200), -1)

sigma = 1
w = 6 * sigma + 1
blur_IMG2 = cv.GaussianBlur(IMG2, (w,w), sigma)
plt.imshow(blur_IMG2)
plt.title('Gaussian')
plt.figure()

canny_IMG2 = cv.Canny(blur_IMG2, 10, 200)
plt.imshow(canny_IMG2)
plt.title('Canny Edge')
plt.figure()

x_edges = cv.Sobel(canny_IMG2, -1, 1, 0, 5)
y_edges = cv.Sobel(canny_IMG2, -1, 0, 1, 5)
#plt.subplot(1,2,1)
#plt.imshow(x_edges)
#plt.subplot(1,2,2)
#plt.imshow(y_edges)
#plt.figure()

#direction = 90 + np.degrees(np.arctan2(y_edges, x_edges))
#print(direction)


ynanugix = y_edges / x_edges
np.nan_to_num(ynanugix, copy=False)

#ynanugix_re2 = np.degrees(np.arctan(ynanugix))
#ynanugix_re1 = 90 + np.degrees(np.arctan(ynanugix))
ynanugix_re1 = 90 - np.degrees(np.arctan(ynanugix))
#print(ynanugix_re1)

#plt.imshow(ynanugix_re1)
#plt.figure()


#z = np.zeros((256,256,3), np.uint8)
#e = np.zeros((256,256,3), np.uint8)

z1 = np.zeros((256,256), np.uint8)
e1 = np.zeros((256,256), np.uint8)

#for i in range(255):
    #for j in range(255):
        #for k in range(3):
            #if(60<ynanugix_re1[i][j][k]<30):
                #z[i][j][k] = ynanugix_re1[i][j][k]

            #if(30.0<=ynanugix_re1[i][j][k]<=60.0):
                #e[i][j][k] = ynanugix_re1[i][j][k]
            #else:
                #z[i][j][k] = ynanugix_re1[i][j][k]

for i in range(255):
    for j in range(255):
        if(30.0<=ynanugix_re1[i][j]<=60.0):
            e1[i][j] = ynanugix_re1[i][j]
        else:
            z1[i][j] = ynanugix_re1[i][j]

#plt.imshow(z)
#plt.figure()
#print(z[200][23])

plt.imshow(e1)
plt.title('30~60 Display')
#plt.figure()
#plt.imshow(z1)
#plt.figure()
```

```
<ipython-input-8-47d889969c7b>:31: RuntimeWarning: divide by zero encountered in true_divide
  ynanugix = y_edges / x_edges
<ipython-input-8-47d889969c7b>:31: RuntimeWarning: invalid value encountered in true_divide
  ynanugix = y_edges / x_edges
```

Out[8]: Text(0.5, 1.0, '30~60 Display')