In [1]:
```
import cv2
import numpy as np
```

In [2]:
```
img_j = cv2.imread("images/j.png",0)
width = int (img_j.shape[1] *2)
height = int(img_j.shape[0] * 2)
dim = (width, height)

# resize image
re_img = cv2.resize(img_j, dim, interpolation = cv2.INTER_AREA)
```

In [18]:
```
kernel = np.ones ((3,3) , np.uint8)
```

In [26]:
```
erosion = cv2.erode(re_img,kernel,iterations=5)
dilation = cv2.dilate(re_img,kernel,iterations=5)
opening = cv2.morphologyEx(re_img, cv2.MORPH_OPEN, kernel)
closing = cv2. morphologyEx(re_img,cv2.MORPH_CLOSE,kernel)
gradient = cv2.morphologyEx(re_img, cv2.MORPH_GRADIENT, kernel)
```

In [27]:
```
cv2.imshow('original', re_img)
cv2.imshow('erode', erosion)
cv2.imshow('dilated',dilation)
cv2.imshow('OPEN' ,opening)
cv2.imshow('CLOSE',closing)
cv2. imshow('Gradient',gradient)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [3]:
```
def getSkeleton(image):

    img_cv = cv2.imread(image, 0)
    size = np.size(img_cv)
    skel = np.zeros(img_cv.shape, np.uint8)
    cv2.imshow('original', img_cv)

    ret, img_cv = cv2.threshold(img_cv, 127, 255, 0)
    element = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))

    done = False

    while(not done):
        eroded = cv2.erode(img_cv, element)
        temp = cv2.dilate(eroded, element)
        temp = cv2.subtract(img_cv, temp)
        skel = cv2.bitwise_or(skel, temp)
        img_cv = eroded.copy()

        zeros = size - cv2.countNonZero(img_cv)
        if zeros == size:
            done = True

    cv2.imshow('skel',skel)

    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

In [4]:
```
getSkeleton('images/opencv.png')
getSkeleton('images/j.png')
```

In [2]:
```
# hit or miss
import cv2 as cv

input_image = np.array((
    [0, 0, 0, 0, 0, 0, 0, 0],
    [0, 255, 255, 255, 0, 0, 0, 255],
    [0, 255, 255, 255, 0, 0, 0, 0],
    [0, 255, 255, 255, 0, 255, 0, 0],
    [0, 0, 255, 0, 0, 0, 0, 0],
    [0, 0, 255, 0, 0, 255, 255, 0],
    [0,255, 0, 255, 0, 0, 255, 0],
    [0, 255, 255, 255, 0, 0, 0, 0]), dtype="uint8")

kernel = np.array((
    [0, 1, 0],
    [1, -1, 1],
    [0, 1, 0]), dtype="int")

output_image = cv.morphologyEx(input_image, cv.MORPH_HITMISS, kernel)
rate = 50
kernel = (kernel + 1) * 127
kernel = np.uint8(kernel)
kernel = cv.resize(kernel, None, fx = rate, fy = rate, interpolation = cv.INTER_NEAREST)
cv.imshow("kernel", kernel)
cv.moveWindow("kernel", 0, 0)

input_image = cv.resize(input_image, None, fx = rate, fy = rate, interpolation = cv.INTER_NEAREST)
cv.imshow("Original", input_image)
cv.moveWindow("Original", 0, 200)

output_image = cv.resize(output_image, None , fx = rate, fy = rate, interpolation = cv.INTER_NEAREST)
cv.imshow("Hit or Miss", output_image)
cv.moveWindow("Hit or Miss", 500, 200)

cv.waitKey(0)
cv.destroyAllWindows()
```

In [ ]:
```
```