In [11]:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

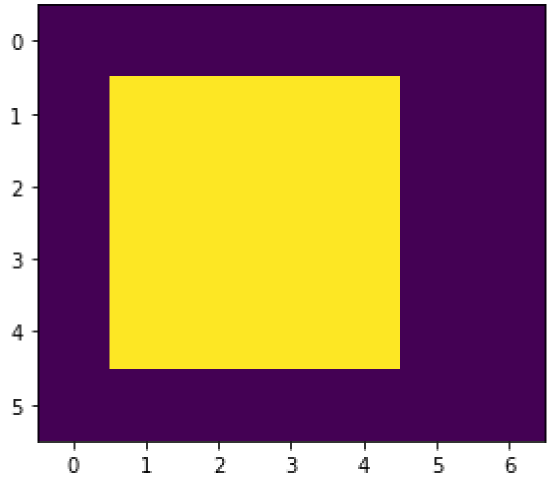
In [21]:

```
def skeletonize(img, gray_then_thres=False, debug=False):
    """ OpenCV function to return a skeletonized version of img, a Mat o
bject"""
#     img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
#     if gray_then_thres:
#         _, img = cv2.threshold(img,250,255,cv2.THRESH_BINARY)
#     img = cv2.bitwise_not(img)

#     #  hat tip to http://felix.abecassis.me/2011/09/opencv-morphologic
al-skeleton/
#     ret,img = cv2.threshold(img,10,255,0)

    img = img.copy() # don't clobber original
    skel = img.copy()

    skel[:,:] = 0
    kernel = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))

    count = 0
    while True:
        print("Iteration={}".format(count))
        eroded = cv2.morphologyEx(img, cv2.MORPH_ERODE, kernel)
        temp = cv2.morphologyEx(eroded, cv2.MORPH_DILATE, kernel) # Open
ing

        print("eroded")
        plt.imshow(eroded)
        plt.show()

        print("opening")
        plt.imshow(temp)
        plt.show()

        temp  = cv2.subtract(img, temp)
        skel = cv2.bitwise_or(skel, temp)   # add onto skel

        print("skel")
        plt.imshow(skel)
        plt.show()

        img[:,:] = eroded[:,:]
        count += 1
        if debug:
            print("count=", count)
        if cv2.countNonZero(img) == 0:
            break


    return skel
```

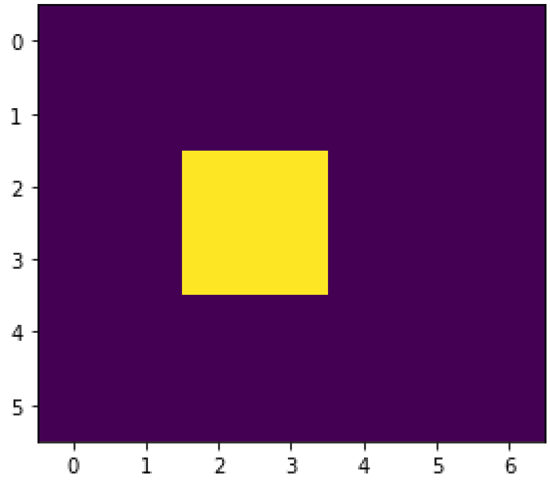
In [26]:

```
a = np.array([[0,0,0,0,0,0,0],
              [0,1,1,1,1,0,0],
              [0,1,1,1,1,0,0],
              [0,1,1,1,1,0,0],
              [0,1,1,1,1,0,0],
              [0,0,0,0,0,0,0],
             ]).astype(np.uint8)

print(">Original Image:")
plt.imshow(a)
plt.show()

print(">Start Skeletonize:")
skel = skeletonize(a)

print(">Result:")
plt.imshow(skel)
plt.show()
```

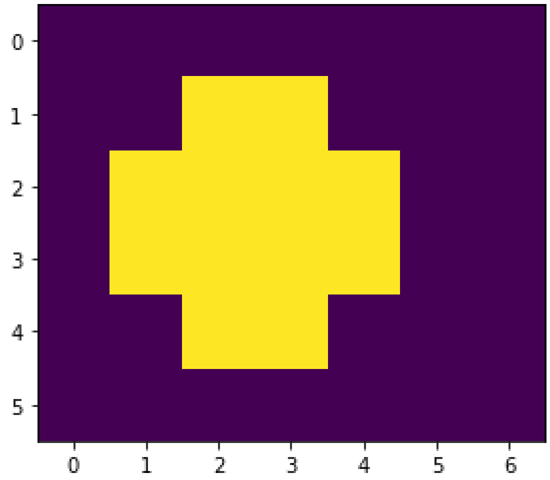
```
>Original Image:
```



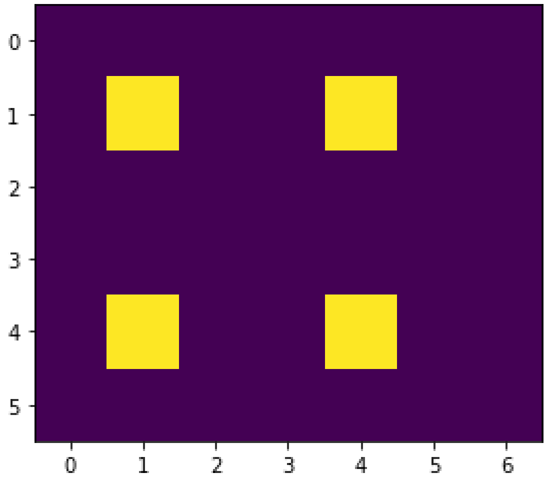
```
>Start Skeletonize:
Iteration=0
eroded
```



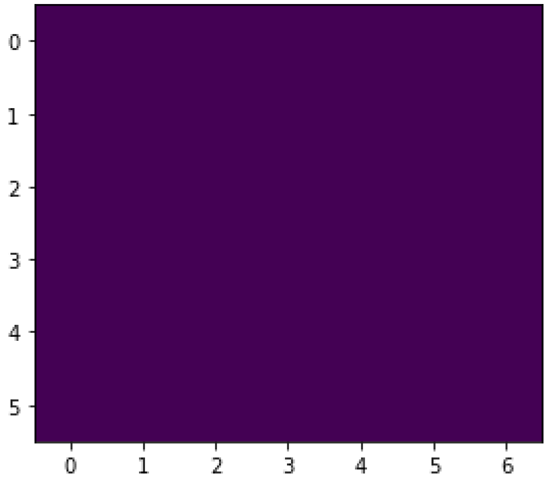
```
opening
```



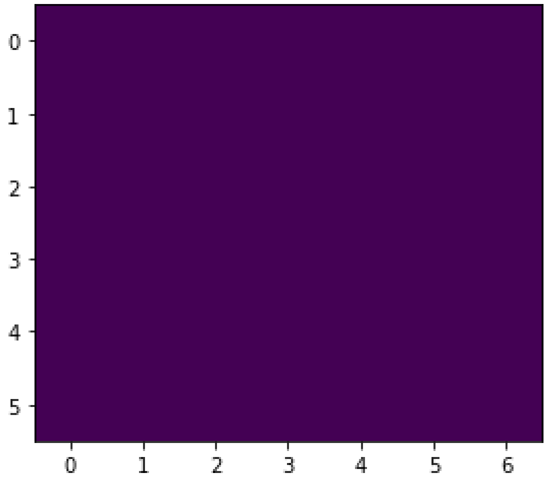
```
skel
```



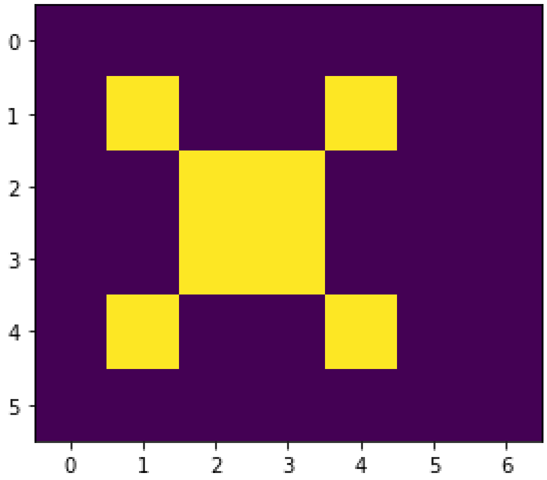
```
Iteration=1
eroded
```



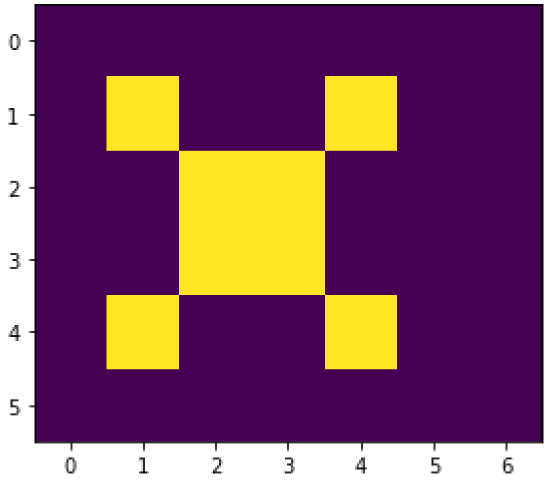
```
opening
```



```
skel
```



```
>Result:
```


0
1
2
3
4
5
0
1
2
3
4
5
6