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

In [18]:
```
img = cv2.imread('lena.png',1)
kernel = np.ones((5,5), np.uint8)
imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# bluring the image   ( src, kernel, sigma )
imgBlur = cv2.GaussianBlur(imgGray, (7,7), 0)
# edge detection using carry
imgCanny = cv2.Canny(img, 150,200)
# to increase the thickness of the edges we need to design a kernel
imgDialation = cv2.dilate(imgCanny, kernel, iterations = 1 )
# to decrease the thickness of the edges we need to use erosion
imgEroded = cv2.erode(imgDialation, kernel, iterations = 1)
cv2.imshow("Gray image", imgGray)
cv2.imshow("Blur image", imgBlur)
cv2.imshow("Canny image", imgCanny)
cv2.imshow("dialate image", imgDialation)
cv2.imshow("erode image", imgEroded)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [22]:
```
# perspective transform
import cv2
import numpy as np

img = cv2.imread('cards.png',1)
width,height = 450, 350
pts1 = np.float32([[162,29],[672,2], [0,254],[543,257]])
pts2 = np.float32([[0,0],[width,0], [0,height],[width, height]])
matrix = cv2.getPerspectiveTransform(pts1, pts2)
imgout = cv2.warpPerspective(img, matrix,(width, height))

cv2.imshow("image", img)
cv2.imshow("Output", imgout)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [24]:
```
#joining images
import cv2
import numpy as np
img = cv2.imread('lena.png')
#horizontal
imghor  = np.hstack((img, img))
#vertical
imgver  = np.vstack((img, img))
cv2.imshow("Output", imghor)
cv2.imshow("Output", imgver)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

In [ ]:
```

```