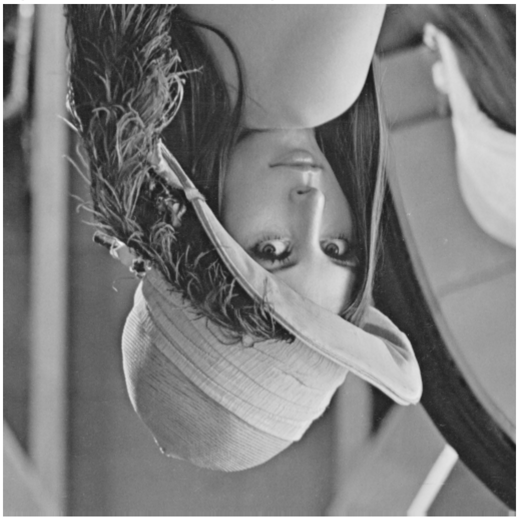
Computer Vision HW1

by b06902034 黃柏諭

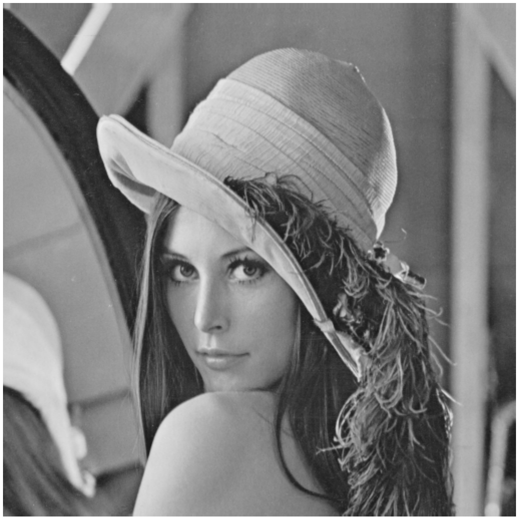
Part 1

```python
import cv2
import numpy as np

image = cv2.imread("lena.bmp")
upsideDown = np.zeros(image.shape)
rightsideLeft = np.zeros(image.shape)
diagonallyMirrored = np.zeros(image.shape)

lenR = image.shape[0]
lenC = image.shape[1]
for r in range(lenR):
    for c in range(lenC):
        upsideDown[lenR - r - 1, c, :] = image[r, c, :]
        rightsideLeft[r, lenC - c - 1, :] = image[r, c, :]
        diagonallyMirrored[c, r, :] = image[r, c, :]

cv2.imwrite("1_upside-down lena.bmp", upsideDown)
cv2.imwrite("2_right-side-left lena.bmp", rightsideLeft)
cv2.imwrite("3_diagonally mirrored lena.bmp", diagonallyMirrored)
```

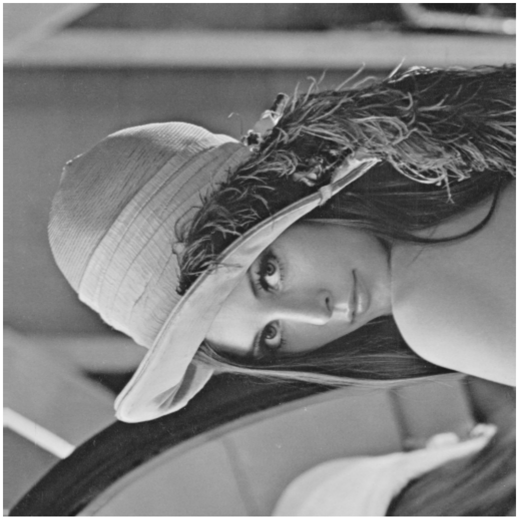
- upside-down lena.bmp



- right-side-left lena.bmp



- diagonally mirrored lena.bmp

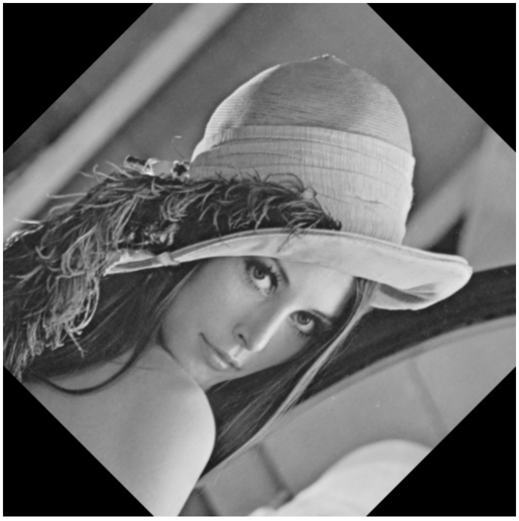


Part 2

Use the following code to get the results

```
import cv2
import numpy as np

image = cv2.imread("lena.bmp")
h, w = image.shape[:2]
center = (h / 2, w / 2)
Mat = cv2.getRotationMatrix2D(center, 45, 1)
cv2.imwrite("4_rotate45.bmp", cv2.warpAffine(image, Mat, (h, w)))
cv2.imwrite("5_shrink.bmp", cv2.resize(image, (h >> 1, w >> 1)))
cv2.imwrite("6_binary.bmp", cv2.threshold(image, 127, 255, cv2.THRESH_BINARY)[1])
```

- rotate 45 degrees clockwise



- shrink lena.bmp in half

- binarize lena.bmp at 128 to get a binary image