

# Homework1

## Report

Use B\_PIX to write a program to generate

**(a)upside-down lena.im**

直接把圖像矩陣做倒序即可



**(b)right-side-left lena.im**

把圖像矩陣中的每一 row 做倒序



**(c)diagonally mirrored lena.im**

有兩種結果

一. 將圖像矩陣中的上三角矩陣的值對應複製到下三角矩陣中



二. 將圖像矩陣中的下三角矩陣的值對應複製到上三角矩陣中



2. Use Photoshop to  
(a) rotate lena.im 45 degrees clockwise



**(b)shrink lena.im in half**



**(c)binarize lena.im at 128 to get a binary image**



## Source code

```
# origin image
import numpy as np # the library to deal with array and matrix
from PIL import Image
img = Image.open('lena.bmp')
img_array = np.array(img)
Image.fromarray(img_array)

# The function to display the image
from IPython.display import display
def show(img_array):
    display(Image.fromarray(img_array))

# upside down
img_ud_array = img_array[::-1]
show(img_ud_array)

# right-side-left
img_rsf_array = np.zeros(img_array.shape, dtype = img_array.dtype)

for i in range(img_array.shape[0]):
    img_rsf_array[i] = img_array[i][::-1]

show(img_rsf_array)

# diagonally mirrored
img_dm1_array = np.zeros(img_array.shape, dtype = img_array.dtype)
img_dm2_array = np.zeros(img_array.shape, dtype = img_array.dtype)

for i in range(img_array.shape[0]):
    for j in range(img_array.shape[1]):
        if (i <= j):
            img_dm1_array[i][j] = img_array[i][j]
            img_dm2_array[i][j] = img_array[j][i]
        else:
```

```
img_dm1_array[i][j] = img_array[j][i]  
img_dm2_array[i][j] = img_array[i][j]  
show(img_dm1_array)  
show(img_dm2_array)
```