HW1

R10944013 馮啟倫

Part1:

(a) upside-down lena.bmp

演算法：每個像素的Col不動，Row則上下顛倒
Code fragment：

```python
def upside_down(img):
    temp_img = np.zeros((512,512,3))
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            temp_img[512 - 1 - row][col] = img[row][col]
    return temp_img
```

(b) right-side-left lena.bmp

演算法：每個像素的Row不動，Col則左右顛倒
Code fragment：

```python
def rightside_left(img):
    temp_img = np.zeros((512,512,3))
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            temp_img[row][512 - 1 - col] = img[row][col]
    return temp_img
```

(c) diagonally flip lena.bmp

演算法：把每一個像素的Row跟Col互換，例如img[a][b] 這個像素點就讓他出現在(b,a)這個位置上

Code fragment：

```python
def diagonally_flip(img):
    temp_img = np.zeros((512,512,3))
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            temp_img[col][row] = img[row][col]
    return temp_img
```

Part2:

(d) rotate lena.bmp 45 degrees clockwise

方法：使用OpenCV的getRotationMatrix2D與warpAffine函數，首先先利用getrotation函數產生一個旋轉矩陣，再用warpaffine函數將原矩陣(影像)映射到旋轉矩陣對應的座標，就產生了Rotate的效果
Code fragment:

```
def rotate(img, angle):
    (h, w) = img.shape[:2]
    center = (h/2, w/2)
    M = cv2.getRotationMatrix2D(center, angle, 1)
    rotated = cv2.warpAffine(img, M, (w, h))

    return rotated
```

(e) shrink lena.bmp in half

方法：使用OpenCV裡面的resize函數，把長，寬都設為一半，並且使用Opencv內建的interpolation補值，即可以做到shrink的事情

Code fragment:

```
def shrink(img):
    (h, w) = img.shape[:2]
    img_modified = cv2.resize(img, (int(h/2), int(w/2)), interpolation=cv2.INTER_AREA)
    return img_modified
```

(f) binarize lena.bmp at 128 to get a binary image

方法：由於照片有512*512*3個數值，每一個數值都去檢驗是否大於等於128，若條件符合則設為255, 反之則設為0

Code fragment:

```
def binarize(img):
    temp_img = np.zeros((512,512,3))
    for row in range(img.shape[0]):
        for col in range(img.shape[1]):
            for channel in range(3):
                if img[row][col][channel] > 127:
                    temp_img[row][col][channel] = 255
                else:
                    temp_img[row][col][channel] = 0
    return temp_img
```