

(a) upside-down lena.bmp

- description:
要將 lena 的影像進行垂直翻轉
- algorithm:
每次固定要交換的 row，並且每個 column 都要交換一次，依序逐次移動一對 row 直到要交換的 row 小於等於自己才停下來。這個停止機制是要避免已交換過的 pixel 又交換一次。
- principal code fragment:

```
h, w, _ = img.shape

for r in range(h):
    swap_r = h - r - 1
    if swap_r > r:
        for c in range(w):
            tmp = np.array(img[r, c, :])
            img[r, c, :] = img[swap_r, c, :]
            img[swap_r, c, :] = tmp
```

- result:



(b) right-side-left lena.bmp

- description:
要將 lena 的影像進行水平翻轉
- algorithm:
每次固定要交換的 column，並且每個 row 都要交換一次，依序逐次移動一對 column 直到要交換的 column 小於等於自己才停下來。這個停止機制是要避免已交換過的 pixel 又交換一次。

- principal code fragment:

```
h, w, _ = img.shape

for c in range(w):
    swap_c = w - c - 1
    if swap_c > c:
        for r in range(h):
            tmp = np.array(img[r, c, :])
            img[r, c, :] = img[r, swap_c, :]
            img[r, swap_c, :] = tmp
```

- result:



(c) diagonally flip lena.bmp

- description:
要將 lena 的影像進行對角翻轉
- algorithm:
逐 row 對每個 pixel 做交換，而 row 的 index 為 k，交換方式是 img[i,j]和 img[j,i]的值交換，下一個 row 則從 img[k+1, k+2]開始交換，因為 img[k+1, k+1]之前的 pixel 接交換過。
- principal code fragment:

```
h, w, _ = img.shape

for c in range(w):
    for r in range(c+1, h):
        tmp = np.array(img[r, c, :])
        img[r, c, :] = img[c, r, :]
        img[c, r, :] = tmp
```

- result:



(d) rotate lena.bmp 45 degrees clockwise

利用 `cv2.getRotationMatrix2D` 得到順時針旋轉 45 的 Matrix，之後再用 `cv2.warpAffine` 對圖片旋轉

```
center = (w / 2, h / 2)

M = cv2.getRotationMatrix2D(center, -45, 1.0)
img_rot = cv2.warpAffine(img, M, (w, h))
```



(e) shrink lena.bmp in half

利用 `cv2.resize` 將邊長縮小一半，然後貼到原本圖片大小的中心

```
result_resize = cv2.resize(img, (w//2, h//2))
img_result = np.zeros(img.shape)
img_result[h//2-h//4:h//2+h//4, w//2-w//4:w//2+w//4, :] = result_resize
```



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

利用 for loop 拜訪每個 pixel，若值大於 180 則改成 255，反之改為 1

```
for c in range(w):
    for r in range(h):
        if img[r, c, 0] > 128:
            img[r, c, 0] = 255
            img[r, c, 1] = 255
            img[r, c, 2] = 255
        else:
            img[r, c, 0] = 0
            img[r, c, 1] = 0
            img[r, c, 2] = 0
```

