








- Part1. Write a program to do the following requirement.

| lena.bmp | (a) upside-down lena.bmp | (b) right-side-left lena.bmp | (c) diagonally mirrored lena.bmp |
|---|---|--|---|
|  |  |  |  |

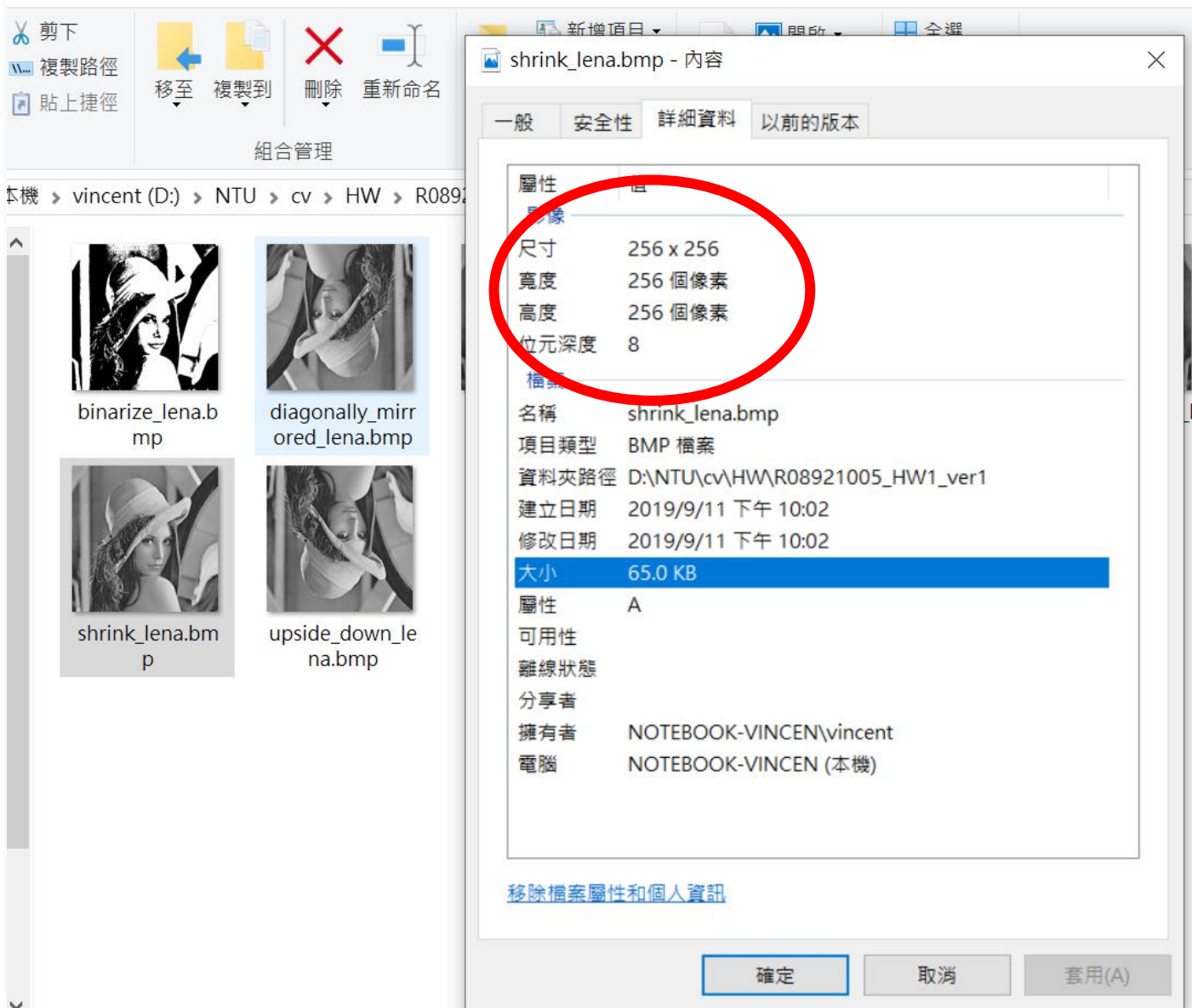
- Part2. Write a program or use software to do the following requirement.

| (d) rotate lena.bmp 45 degrees clockwise | (e) shrink lena.bmp in half | (f) binarize lena.bmp at 128 to get a binary image |
|---|--|---|
|  |  |  |

說明

1. 圖片檔可參考資料夾內圖片
2. 本部分使用 python 完成，編譯器採用 spyder。
3. 真實圖片大小可點選圖片右鍵→選取『內容』→按詳細資料即可查看大小，

如(e)小題需要縮小 1/2，下面圖片將可證明確實將原來 512*512 縮小為 256*256。



4. 本作業所有題目皆由 python 完成，以下將進程式碼解說

Part1

```

1 #-*-coding: utf-8 -*-|
2 """
3 Created on Wed Sep 11 19:33:55 2019
4
5 @author: vincent黃國郡
6 """
7 from PIL import Image
8 import numpy as np
9
10 lena_pic = Image.open('lena.bmp')
11 original_lena_array = np.array(lena_pic)
12 #####
13 ##### Part1 #####
14 #part(a) upside-down lena.bmp
15 upside_down_lena = original_lena_array.copy()
16 for i in range(len(original_lena_array)):
17     for j in range(len(original_lena_array[i])):
18         upside_down_lena[len(original_lena_array)-i-1][j] = original_lena_array[i][j]
19 Image.fromarray(upside_down_lena.astype(np.uint8)).save('upside_down_lena.bmp')
20
21 #part(b) right-side-left lena.bmp
22 right_side_left_lena = original_lena_array.copy()
23 for i in range(len(original_lena_array)):
24     for j in range(len(original_lena_array[i])):
25         right_side_left_lena[i][len(original_lena_array)-j-1] = original_lena_array[i][j]
26 Image.fromarray(right_side_left_lena.astype(np.uint8)).save('right_side_left_lena.bmp')
27
28 #part(c) diagonally mirrored lena.bmp
29 diagonally_mirrored_lena = original_lena_array.copy()
30 for i in range(len(original_lena_array)):
31     for j in range(len(original_lena_array[i])):
32         diagonally_mirrored_lena[len(original_lena_array)-i-1][len(original_lena_array)-j-1] = original_lena_array[i][j]
33 Image.fromarray(diagonally_mirrored_lena.astype(np.uint8)).save('diagonally_mirrored_lena.bmp')
34 #####

```

加入圖片 IO 以及 numpy 方便 array 運算

將原檔 lena.bmp 讀進來，並裝進 numpy array

先複製一個和 lena 同樣大小矩陣，並命名為 upside_down_lena

原本 lena 第一列的每個 pixel 對應到顛倒 upside_down_lena 的最後一行，利用迴圈依序上下顛倒過來整張圖，再將之存入 bmp 檔設定為 unit8 格式

先複製一個和 lena 同樣大小矩陣，並命名為 right_side_left_lena

原本 lena 第一行的每個 pixel 對應到顛倒 upside_down_lena 的最後一行，利用迴圈，做完第一行，做第二行，依序左右顛倒過來整張圖，再將之存入 bmp 檔設定為 unit8 格式

先複製一個和 lena 同樣大小矩陣，並命名為 diagonally_mirrored_lena

原本 lena 第一列第一行的每個 pixel 對應到顛倒 diagonally_mirrored_lena 的最後一列最後一行，利用迴圈，做完第一列第一行，做第二列第二行，依序左右上下顛倒過來整張圖，再將之存入 bmp 檔設定為 unit8 格式

Part2

```

36 ##### Part2 #####
37 #part(d) rotate lena.bmp 45 degrees clockwise
38 from scipy import ndimage
39 rotate_lena = ndimage.rotate(original_lena_array, 45)
40 Image.fromarray(rotate_lena.astype(np.uint8)).save('rotate_lena.bmp')
41
42 #part(e) shrink lena.bmp in half
43 half = 0.5
44 shrink_lena = lena_pic.resize([int(half * s) for s in lena_pic.size])
45 shrink_lena.save('shrink_lena.bmp')
46
47 #part(f) binarize lena.bmp at 128 to get a binary image
48 binarize_lena = original_lena_array.copy()
49 for i in range(len(original_lena_array)):
50     for j in range(len(original_lena_array[i])):
51         if (original_lena_array[i][j] >= 128):
52             binarize_lena[i][j] = 255
53         else:
54             binarize_lena[i][j] = 0
55 Image.fromarray(binarize_lena.astype(np.uint8)).save('binarize_lena.bmp')

```

由於沒有限制使用 library，因此使用 ndimage 的 rotate 幫我完成旋轉

設定旋轉 45 度，再將之存入 bmp 檔設定為 unit8 格式

取出 lena 的 size，長和寬，再將之乘 0.5 放入 resize 函數，將生成的 shrink_lena 存成 bmp 檔

先複製一個和 lena 同樣大小矩陣，並命名為 binarize_lena

從左上到右下取出每一個 pixel，如果該 pixel 大於 128 則將 binarize_lena 設為 255 反之設為 0，從左到右，從上到下依序跑完迴圈，再將 binarize_lena 這個 array 存入 bmp 檔設定為 unit8 格式