

## Environment

- python3.9
- Pillow 10.0.0, numpy 1.25.2

## I/O

```
from PIL import Image
import numpy as np
import copy

img = Image.open('./lena.bmp') # load lena.bmp
img_array = np.array(img) # pixel content saved in np.array
width, height = img_array.shape # get `width` and `height`
img_list = img_array.tolist() # transform pixel content into list
```

## Part 1

### a. upside down

```
result = copy.deepcopy(img_list)

for y in range(height//2):
    rowy = copy.deepcopy(result[y])
    result[y] = copy.deepcopy(result[(height-1)-y])
    result[(height-1)-y] = copy.deepcopy(rowy)

img_ = Image.fromarray(np.array(result, dtype='uint8'), mode='L')
img_.save('./lena_upside_down.bmp')
```

1. remove mutability of nested list
2. loop through half the rows

3. for each row `y`, swap it with row `(height-1) - y`



## b. rightside left (horizontal flip)

```
result = copy.deepcopy(img_list)

for y in range(height):
    for x in range(width//2):
        elmx = result[y][x]
        result[y][x] = result[y][(width-1)-x]
        result[y][(width-1)-x] = elmx

img_ = Image.fromarray(np.array(result, dtype='uint8'), mode='L')
img_.save('./lena_rightside_left.bmp')
```

1. remove mutability of nested list
2. loop through the rows
3. in each row `y`, loop through half the columns

4. for each column  $x$  (of the row  $y$ ), swap the column  $x$  with column  $(width-1)-x$



### 3. diagonal mirrored

```
result = copy.deepcopy(img_list)

for y in range(height//2):
    rowy = copy.deepcopy(result[y])
    result[y] = copy.deepcopy(result[(height-1)-y])
    result[(height-1)-y] = copy.deepcopy(rowy)

for y in range(height):
    for x in range(width//2):
        elmx = result[y][x]
        result[y][x] = result[y][(width-1)-x]
        result[y][(width-1)-x] = elmx

img_ = Image.fromarray(np.array(result, dtype='uint8'), mode='L')
img_.save('./lena_diagonal_mirrored.bmp')
```

1. remove mutability of nested list

2. first `for` loop: upside down
3. second (nested) `for` loop: rightside left



---

## Part 2

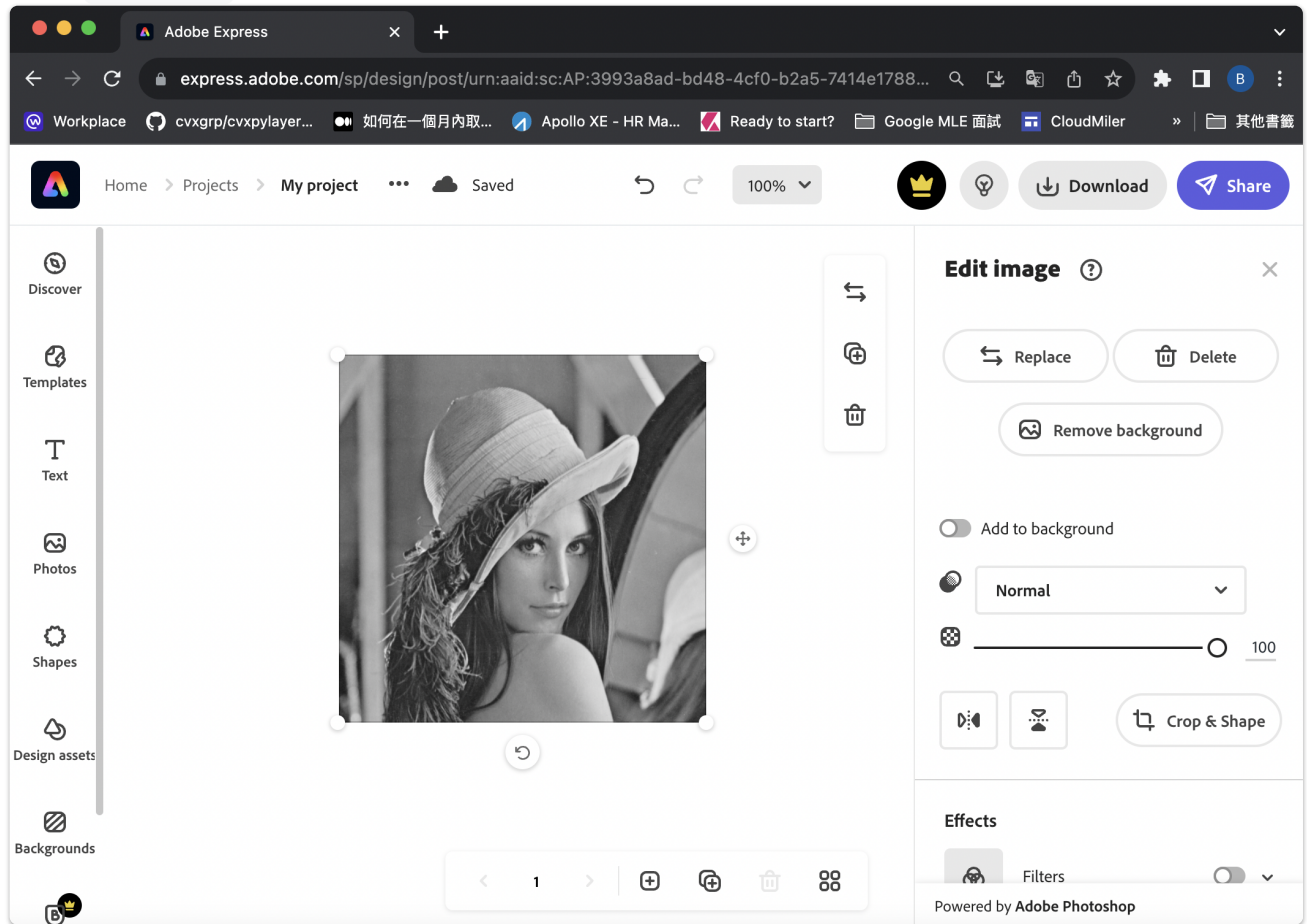
### d. rotate 45° clockwise

use adobe photoshop online

<https://www.adobe.com/tw/express/feature/image/editor>

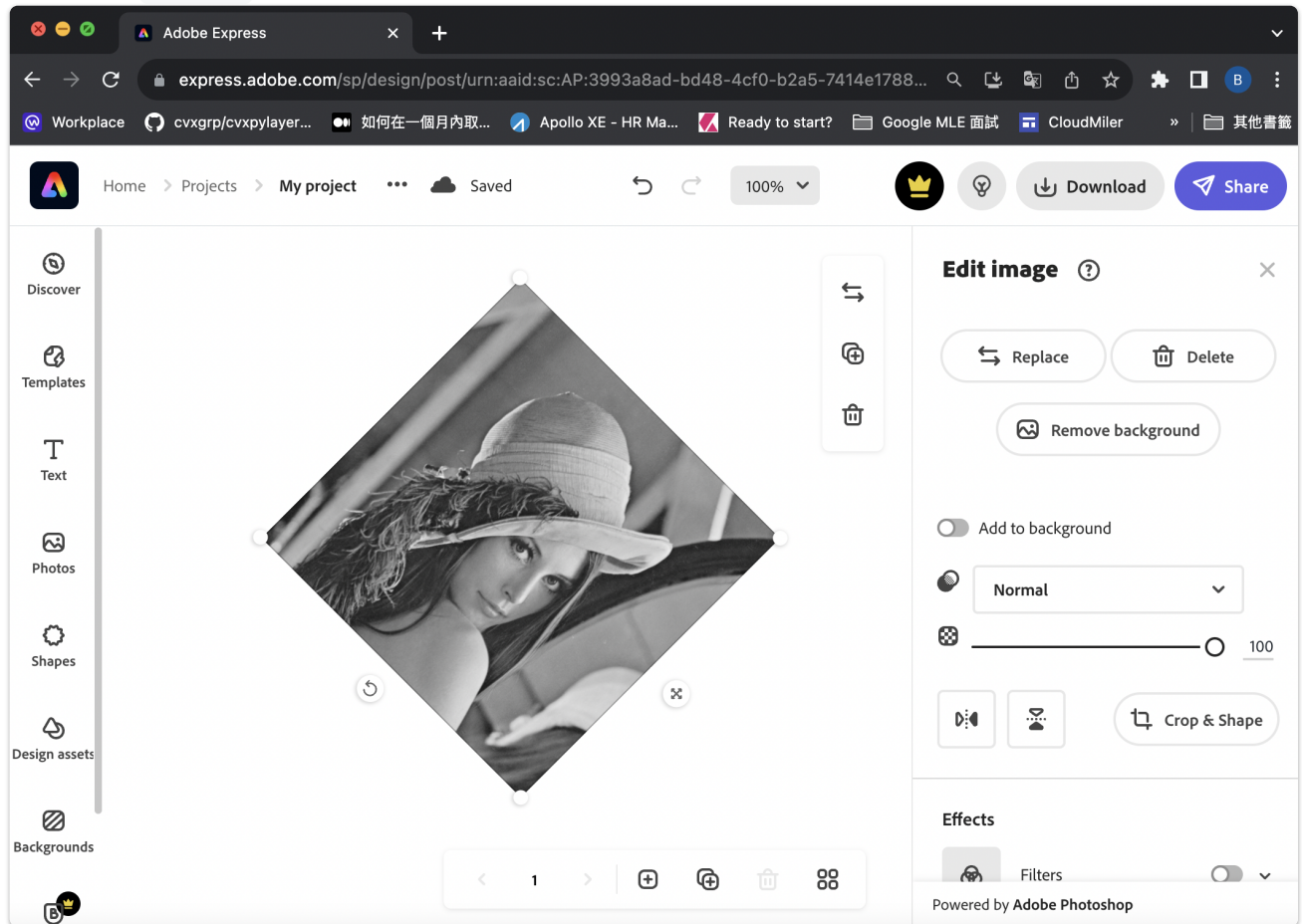
1. open a new project on adobe photoshop online
2. change `lena.bmp` to `lena.png` (it only accepts png, jpeg, and jpg)

### 3. drag lena.png to adobe photoshop online





4. click the rotate icon, and rotate 45° clockwise



## e. shrink

```
result = copy.deepcopy(img_list)

for y in range(0, height, 2):
    for x in range(0, width, 2):
        elm = result[y][x]
        result[y//2][x//2] = elm

img_ = Image.fromarray(np.array(result, dtype='uint8')[0:height//2,
0:width//2], mode='L')
img_.save('./lena_shrink.bmp')
```

1. remove mutability of nested list
2. loop through every 2 row
3. in each even row `y`, loop through every 2 column of the row
4. for each column `x` (of the row `y`), save the content to row `y//2`, column `x//2`. which will make the left upper corner a 256 x 256 shrunk image



### 3. binarize at 128

```
result = copy.deepcopy(img_list)

for y in range(height):
    for x in range(width):
        elm = 255 if result[y][x] >= 128 else 0
        result[y][x] = elm

img_ = Image.fromarray(np.array(result, dtype='uint8'), mode='L')
img_.save('./lena_binarize.bmp')
```

1. remove mutability of nested list
2. loop through every row
3. in each even row `y`, loop through every column of the row
4. for each column `x` (of the row `y`),  
+ set 255 if pixel value  $\geq 128$

+ set 0 if pixel value < 128

