

original image



Part1.(a)	Part1.(b)	Part1.(c)
A grayscale photograph of the same woman and hat, rotated 90 degrees clockwise. Her head is now at the bottom left, and her hat is at the top right.	A grayscale photograph of the same woman and hat, rotated 90 degrees counter-clockwise. Her head is now at the top left, and her hat is at the bottom right.	A grayscale photograph of the same woman and hat, rotated 180 degrees. She is upside down, with her head at the bottom and her hat at the top.
Part2.(d)	Part2.(e)	Part2.(f)
A grayscale photograph of the same woman and hat, rotated 45 degrees clockwise. The image is tilted to the right.	A small, square grayscale photograph of the same woman and hat, centered within a larger white square frame.	A high-contrast, binary (black and white) version of the original image. The woman's face and hat are white against a black background.

**The difference between version 1 and 2 is the rotating direction in part2.(d)**  
“python R11922150\_HW1.py” to run the program.

In this program, I define 7 functions, 6 of which are corresponding to 6 tasks, and the last one is a show function to show an image. The program will first show the six images which are respectively generated by each function, then write them into six image files.

Among these six tasks, (a),(b),(c), and (f) tasks can all be done by easy for loops and computation. In (d) task, sin and cos 45 degrees are not integers which makes the task hard to implement, so I choose to use scipy.ndimage function - rotate to solve it. As for the (e) task, I'm not sure which pixel we should choose during sampling the pixels (max/min/average/random/or else?), so I choose to use the resize function in cv2 to solve it.