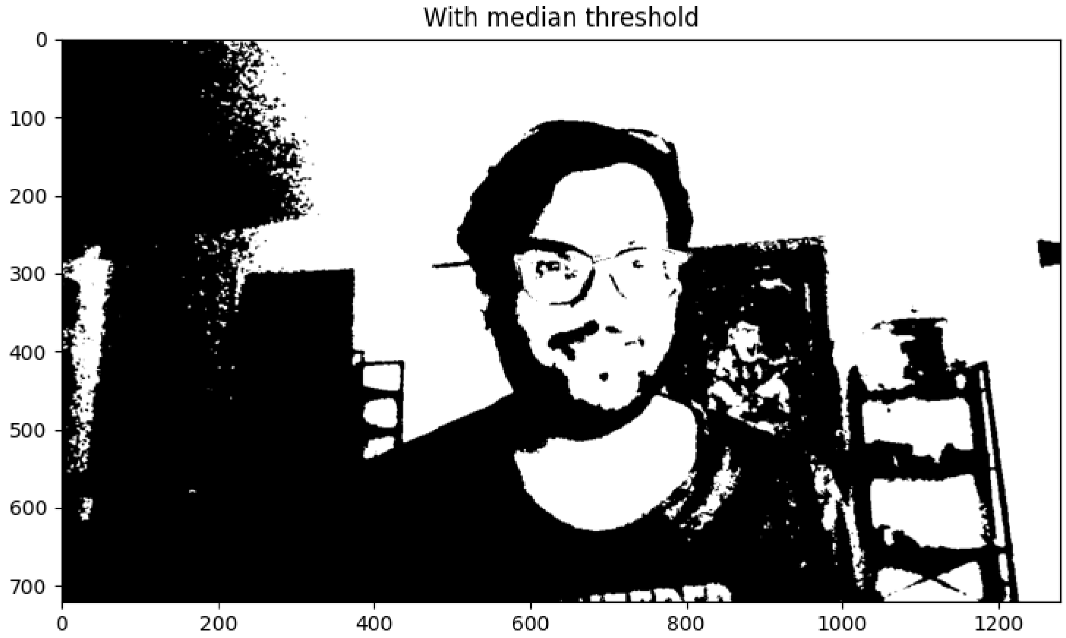
Q5 a) –

Threshold calculated using median of image. My selfie’s median is 24.0.

I used cv2.threshold function to calculate the threshold-ed image:



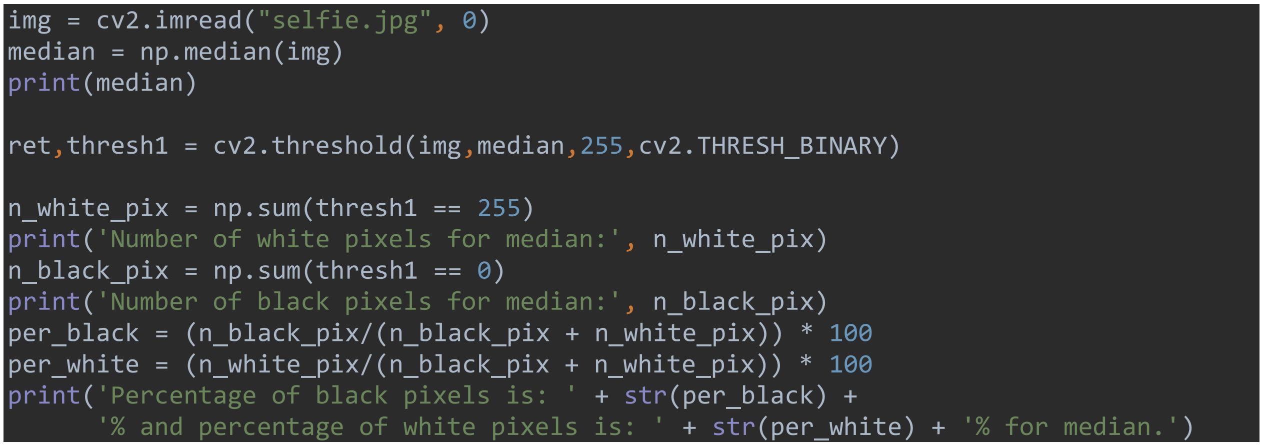
The result gives an output with the number of black and white pixels very close to each other.

Number of white pixels for median: 456309

Number of black pixels for median: 465291

Percentage of black pixels is: 50.4873046875% and percentage of white pixels is: 49.5126953125% for median.

Code (with cv2, numpy and matplotlib imports) –

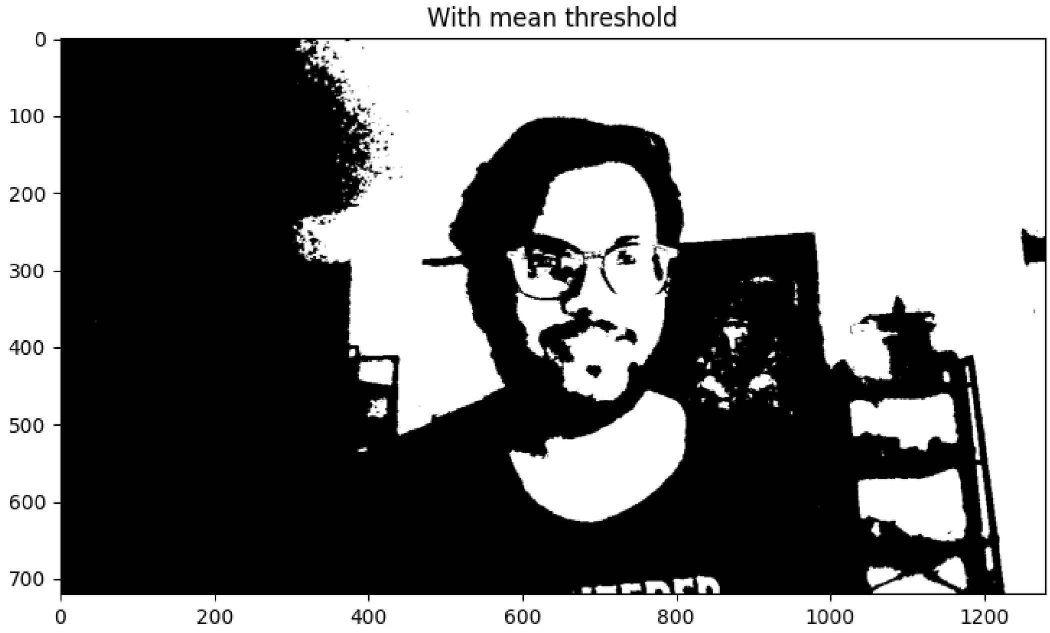


```
img = cv2.imread("selfie.jpg", 0)
median = np.median(img)
print(median)

ret,thresh1 = cv2.threshold(img,median,255,cv2.THRESH_BINARY)

n_white_pix = np.sum(thresh1 == 255)
print('Number of white pixels for median:', n_white_pix)
n_black_pix = np.sum(thresh1 == 0)
print('Number of black pixels for median:', n_black_pix)
per_black = (n_black_pix/(n_black_pix + n_white_pix)) * 100
per_white = (n_white_pix/(n_black_pix + n_white_pix)) * 100
print('Percentage of black pixels is: ' + str(per_black) +
      '% and percentage of white pixels is: ' + str(per_white) + '% for median.')
```

Q5 b) Threshold calculated using mean. The value is 32.18.

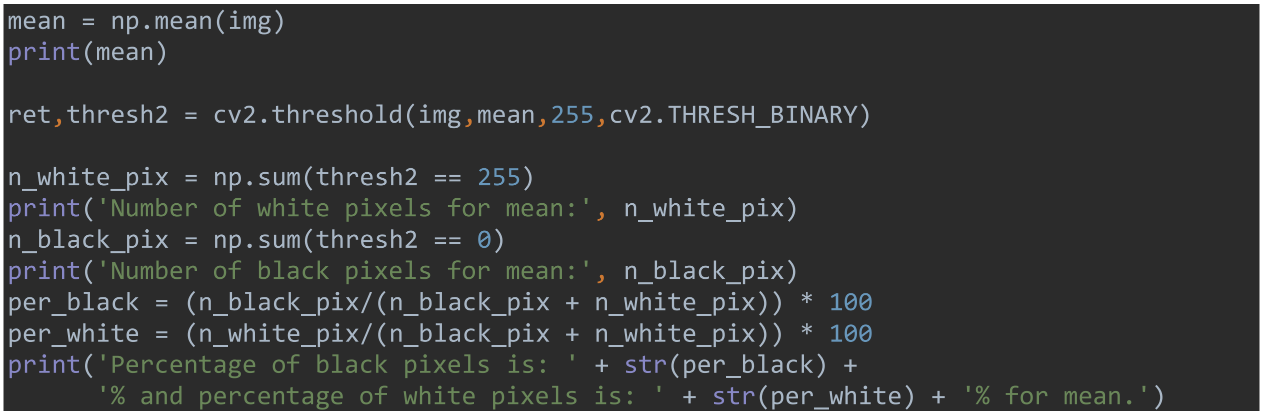


Number of white pixels for mean: 384220

Number of black pixels for mean: 537380

Percentage of black pixels is: 58.30946180555555% and percentage of white pixels is: 41.69053819444444% for mean. Median is a better way for binarization with ~50 % black & white pixels.

Code (with cv2, numpy and matplotlib imports) –



```
mean = np.mean(img)
print(mean)

ret,thresh2 = cv2.threshold(img,mean,255,cv2.THRESH_BINARY)

n_white_pix = np.sum(thresh2 == 255)
print('Number of white pixels for mean:', n_white_pix)
n_black_pix = np.sum(thresh2 == 0)
print('Number of black pixels for mean:', n_black_pix)
per_black = (n_black_pix/(n_black_pix + n_white_pix)) * 100
per_white = (n_white_pix/(n_black_pix + n_white_pix)) * 100
print('Percentage of black pixels is: ' + str(per_black) +
      '% and percentage of white pixels is: ' + str(per_white) + '% for mean.')
```