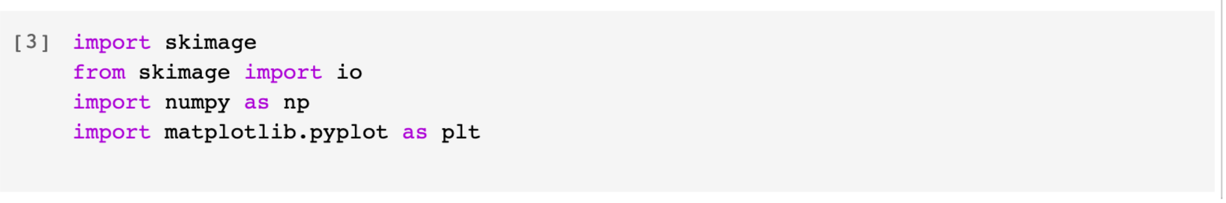
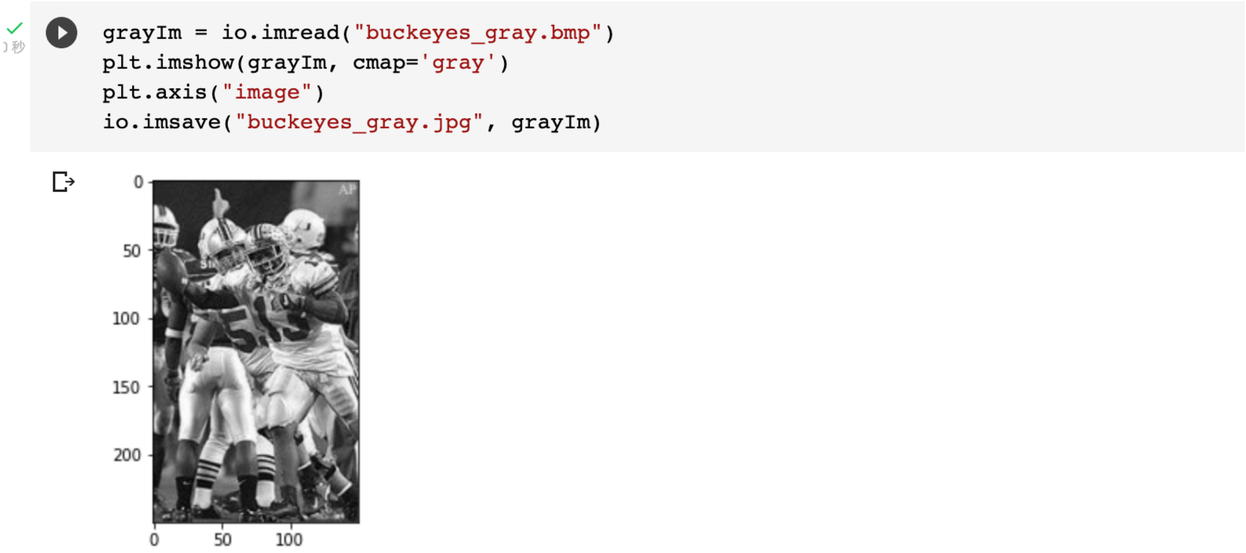
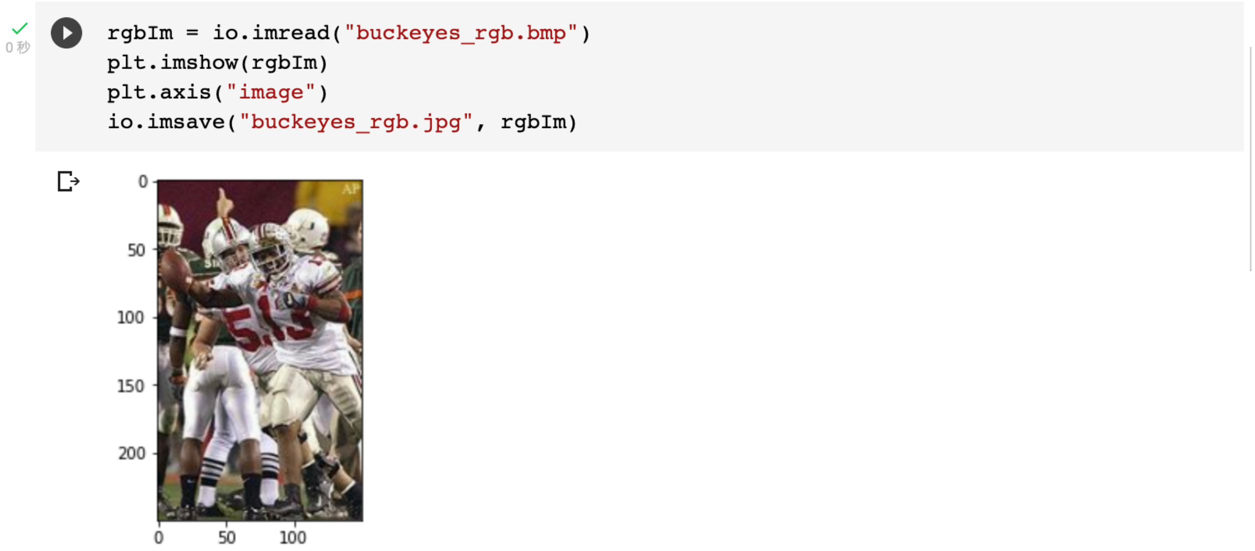
1.



```
[3] import skimage
    from skimage import io
    import numpy as np
    import matplotlib.pyplot as plt
```



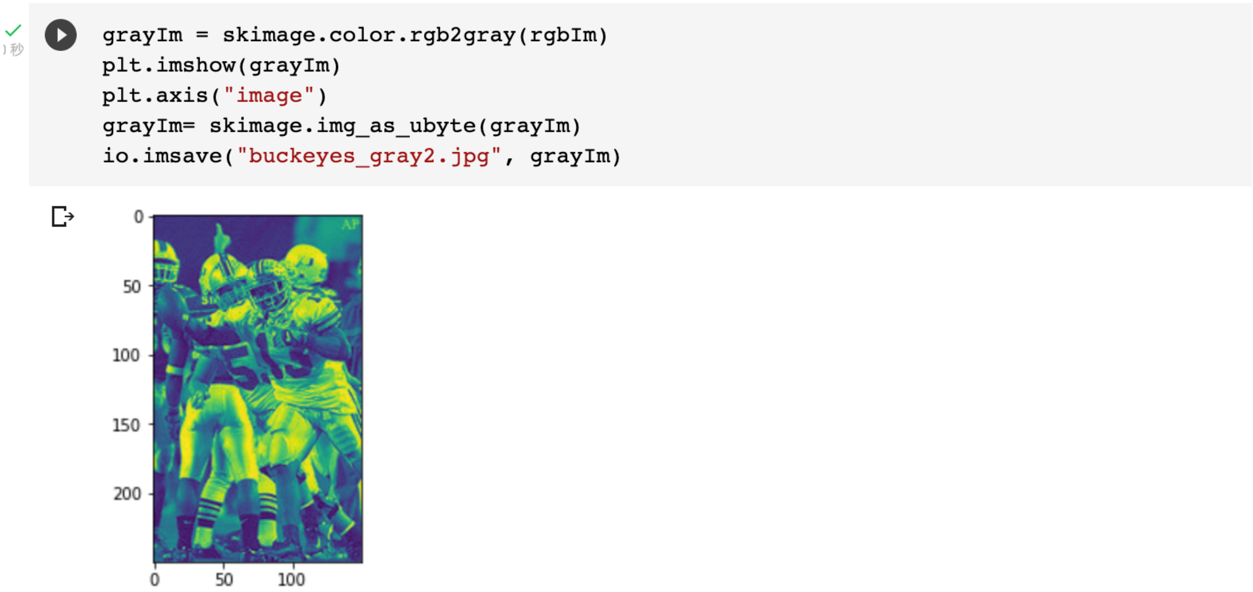
```
grayIm = io.imread("buckeyes_gray.bmp")
plt.imshow(grayIm, cmap='gray')
plt.axis("image")
io.imsave("buckeyes_gray.jpg", grayIm)
```



```
rgbIm = io.imread("buckeyes_rgb.bmp")
plt.imshow(rgbIm)
plt.axis("image")
io.imsave("buckeyes_rgb.jpg", rgbIm)
```

For question 1, I directly use the code given to us.

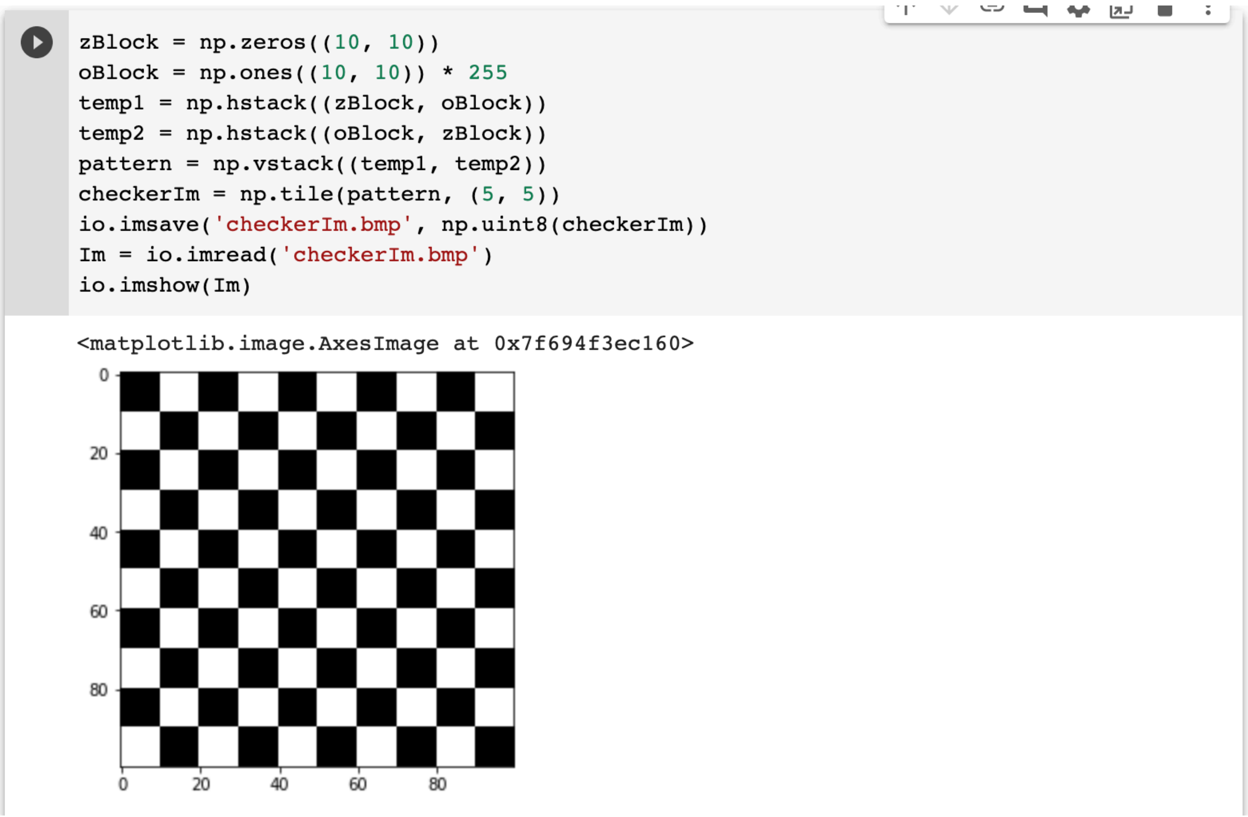
2.



```
grayIm = skimage.color.rgb2gray(rgbIm)
plt.imshow(grayIm)
plt.axis("image")
grayIm= skimage.img_as_ubyte(grayIm)
io.imsave("buckeyes_gray2.jpg", grayIm)
```

For question 2, I first use the code from the HW description, then, I display the image and convert the from float64 to type unite8 to save it without warning.

3.



```
zBlock = np.zeros((10, 10))
oBlock = np.ones((10, 10)) * 255
temp1 = np.hstack((zBlock, oBlock))
temp2 = np.hstack((oBlock, zBlock))
pattern = np.vstack((temp1, temp2))
checkerIm = np.tile(pattern, (5, 5))
io.imsave('checkerIm.bmp', np.uint8(checkerIm))
Im = io.imread('checkerIm.bmp')
io.imshow(Im)
```

```
<matplotlib.image.AxesImage at 0x7f694f3ec160>
```

For question 3, I directly use the code given to us.