# HW01

The gray and RGB images are read and shown as below,

|  |  |
| --- | --- |
| A group of football players  Description automatically generated with low confidence  Figure 1: Gray image | A picture containing text, person, player, outdoor  Description automatically generated  Figure : RGB image |

Converted gray image from RGB is displayed,

## A group of football players Description automatically generated with low confidence

Figure : Gray image converted from RGB

## Created checker image is as follows,

## A black and white checkered floor Description automatically generated

Figure : Checker image

Attatched Python3 code:

import matplotlib.pyplot as plt

import numpy as np

from skimage import color

# ----------------------------------------------- #

filename\_grey = "buckeyes\_gray.bmp"

filename\_rgb = "buckeyes\_rgb.bmp"

# Part 1 #

img\_grey = plt.imread(filename\_grey)

plt.imshow(img\_grey, cmap="gray", aspect="equal")

plt.show()

plt.imsave("buckeyes\_gray.jpg", img\_grey, cmap="gray")

img\_rgb = plt.imread(filename\_rgb)

plt.imshow(img\_rgb, aspect="equal")

plt.show()

plt.imsave("buckeyes\_rgb.jpg", img\_rgb)

# Part 2 #

img\_grey\_converted = color.rgb2gray(img\_rgb)

plt.imshow(img\_grey\_converted, cmap="gray", aspect="equal")

plt.show()

plt.imsave("buckeyes\_gray\_convertded.bmp", img\_grey\_converted, cmap="gray")

# Part 3 #

zblock = np.zeros([10,10])

oblock = np.ones([10,10]) \* 255

r1 = np.concatenate( [zblock,oblock], axis=1 )

r2 = np.concatenate( [oblock,zblock], axis=1 )

pattern = np.concatenate( [r1,r2], axis=0 )

img\_checker = np.tile(pattern, [5,5])

plt.imshow(img\_checker, cmap="gray")

plt.show()

plt.imsave("checkerIm.bmp", img\_checker, cmap="gray")