In [1]:

```
# PIL - Python image liberary
# pic
# TASK >>> GRAYSCALE DETECTION
# determine whether the input image is colored or grayscale.

# Approach  ( Rough ):
#     Calculate Pixel per variance , mean color bias adjustment

from PIL import Image, ImageStat
def GRAYSCALE_DETECTION(path, size=500,colourbias=True, msecutoff=22 ):
#     open image file
    img = Image.open(path)
#     get channel
#     3 channel      4 channel (png type file)
#     R = Red        R = Red
#     G = green      G = green
#     B = blue       B = blue
#                    A = alpha -> determines how opaque each pixel is
    c = img.getbands()
    if c == ('R','G','B') or c == ('R','G','B','A'):
#         resize image to reduce complexity
        stat_resize = img.resize((size,size))

        SSE, bias = 0, [0,0,0]

        if colourbias:
            bias = ImageStat.Stat(stat_resize).mean[:3]
            bias = [b - sum(bias)/3 for b in bias ]

        for px in stat_resize.getdata():
            mu = sum(px)/3
            s = 0
            for i in [0,1,2]:
                SSE+=(px[i] - bias[i] - mu) ** 2

#         mean square error
        MSE = float(SSE)/(size*size)

        if MSE <= msecutoff:
            return "RESULT : GRAYSCALE"

        else:
            return "RESULT : COLOR"
    elif len(c)==1:
        return "RESULT : GRAYSCALE"
    else:
        return "ERROR : CAN'T IDENTIFY "

# add path of image to the function GRAYSCALE_DETECTION(<>)

# print(GRAYSCALE_DETECTION(<path>))
```

In [2]:

```
print
(
GRAYSCALE_DETECTION("dataset/1.jpg"),
GRAYSCALE_DETECTION("dataset/2.jpg"),
GRAYSCALE_DETECTION("dataset/3.jpg"),
GRAYSCALE_DETECTION("dataset/4.jpg"),
GRAYSCALE_DETECTION("dataset/5.jpg"),
GRAYSCALE_DETECTION("dataset/6.jpg"),
GRAYSCALE_DETECTION("dataset/7.jpg"),
GRAYSCALE_DETECTION("dataset/8.jpg"),
GRAYSCALE_DETECTION("dataset/9.jpg"),
GRAYSCALE_DETECTION("dataset/11.jpg"),
GRAYSCALE_DETECTION("dataset/12.jpg"),
GRAYSCALE_DETECTION("dataset/13.jpg"),
GRAYSCALE_DETECTION("dataset/14.jpg"),
GRAYSCALE_DETECTION("dataset/15.jpg"),
GRAYSCALE_DETECTION("dataset/16.jpg"),
GRAYSCALE_DETECTION("dataset/17.jpg"),
GRAYSCALE_DETECTION("dataset/19.jpg"),
GRAYSCALE_DETECTION("dataset/20.jpg"))
```

Out[2]:

```
('RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE',
 'RESULT : GRAYSCALE')
```