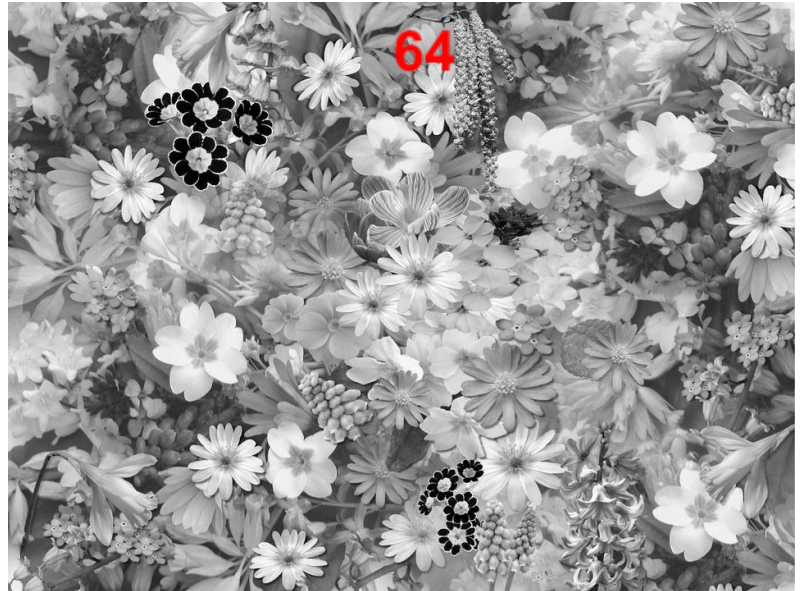


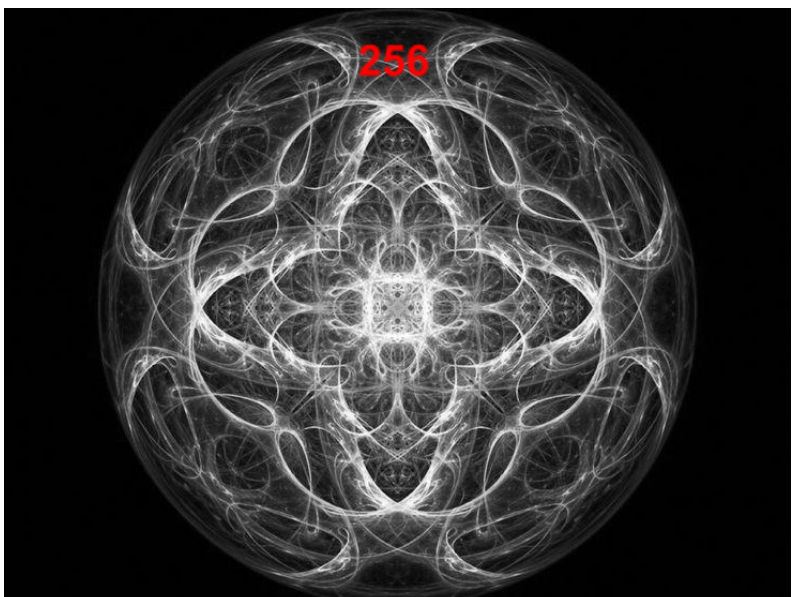
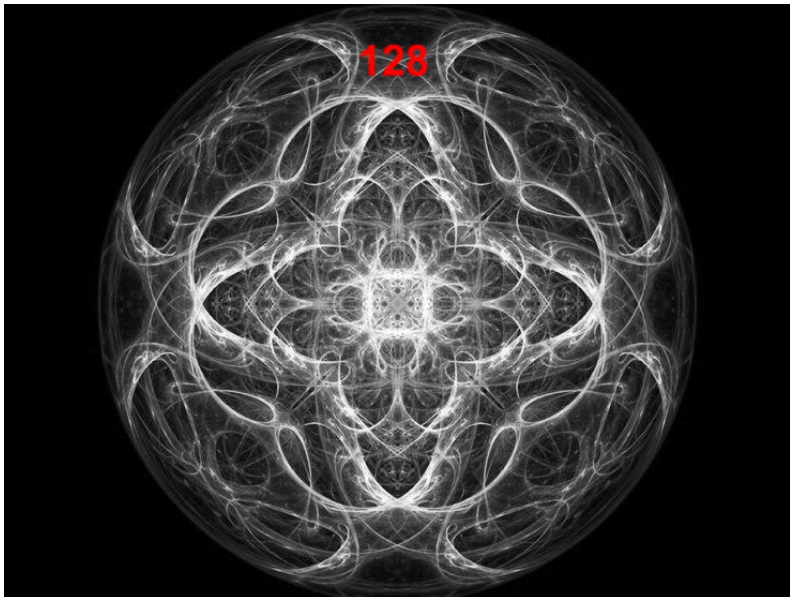
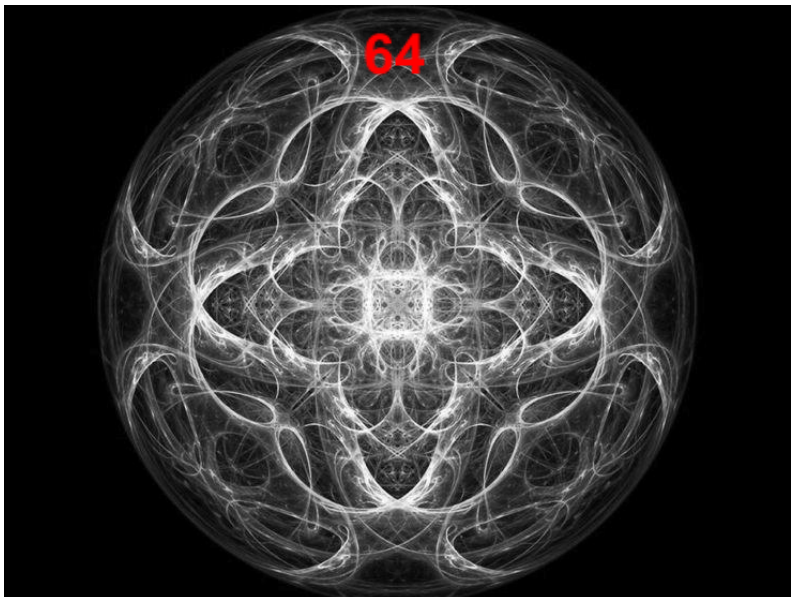
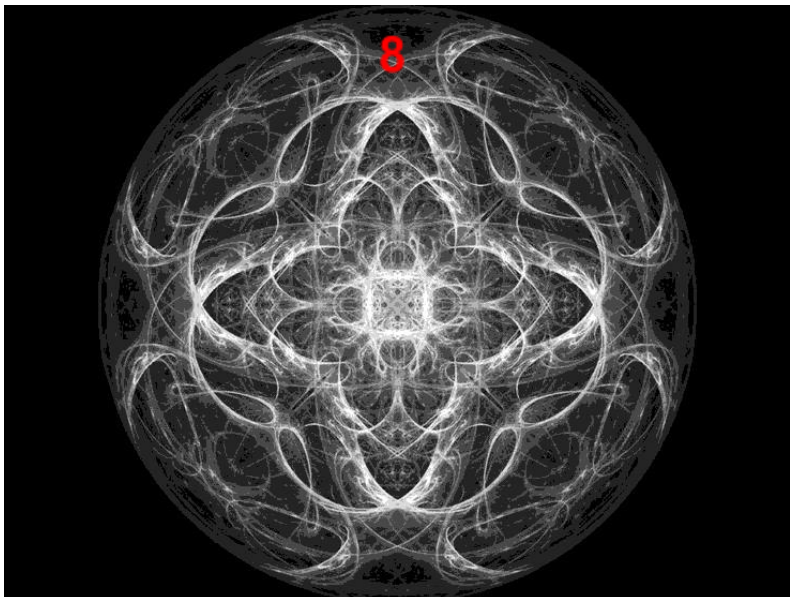
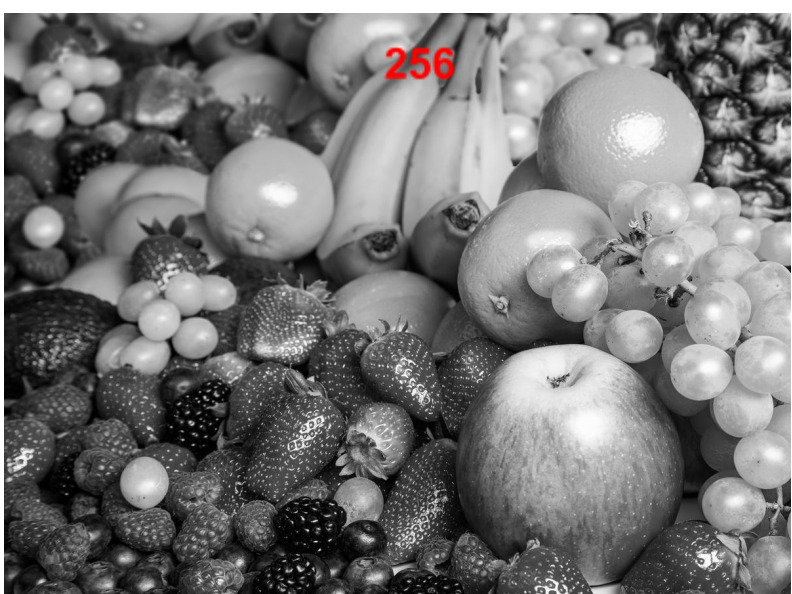
**Q1:** ภาพที่มี gray level น้อยๆรายละเอียดของสีจะหายไป สีที่เคยแยกออกจากกันจะเริ่มกลืนกัน

```
from PIL import Image
from PIL.Image import Image as IMG
def displayCustomGrayLevel(image:IMG, levels:int):
    image = image.convert("L") # convert to grayscale
    max_level = 255 # max gray scale value of this image (255 for 8 bit)
    new_level = levels - 1 # new max gray scale value
    for x in range(image.height): # height
        for y in range(image.width): # width
            old = image.getpixel((y,x))
            new = (round(old/max_level * new_level))/new_level*max_level
            image.putpixel((y,x), round(new))
    image.save(f"ImageProcessing/assign1/q1/fruit{levels}levels.jpg")
img = Image.open("ImageProcessing/assign1/sample/fruit.jpg")
displayCustomGrayLevel(img, 4)
displayCustomGrayLevel(img, 64)
displayCustomGrayLevel(img, 128)
displayCustomGrayLevel(img, 256)
```

**Output**









## Q2: คล้ายเป็นการ Invert ภาพจากสีดํากลายเป็นขาว ขาวกลายเป็นดำ

```
from PIL import Image

from PIL.Image import Image as IMG

def enhance(image:IMG):
    image = image.convert("L") # convert to grayscale
    max_level = 255 # max gray scale value of this image (255 for 8 bit)
    for x in range(image.height): # height
        for y in range(image.width): # width
            r = image.getpixel((y,x))
            if r < (max_level/3):
                s = 5*max_level/6
            elif r < (2*max_level/3):
                s = (-2 * r) + (9 * max_level /6)
            else:
                s = max_level/6
            image.putpixel((y,x), round(s))
    image.save(f"ImageProcessing/assign1/q2/tram-ENHANCE.jpg")

img = Image.open("ImageProcessing/assign1/sample/tram.jpg")
enhance(img)
```





Q3: ทำให้เห็นรายละเอียดภาพได้ดีขึ้นหากปรับค่า parameter ให้พอดี

```
from PIL import Image
from PIL.Image import Image as IMG
def enhancePowerLaw(image:IMG, c, gamma):
    image = image.convert("L") # convert to gray scale
    max_level = 255 # max gray scale value of this image (255 for 8 bit)
    max_transformed_value = c*pow(max_level, gamma)
    for x in range(image.height): # height
        for y in range(image.width): # width
            r = image.getpixel((y,x))
            s = c*pow(r, gamma)
            s = s/max_transformed_value * max_level # convert value back to 0-255 range
            image.putpixel((y,x), round(s))
    image.save(f"ImageProcessing/assign1/q3/scenery2-PowerLaw-c{c}-y{gamma}.jpg")

img = Image.open("ImageProcessing/assign1/sample/scenery2.jpg")
c_arr = [0.5, 1, 2]
y_arr = [0.4, 2.5]
for c in c_arr:
    for y in y_arr:
        enhancePowerLaw(img, c, y)
```

