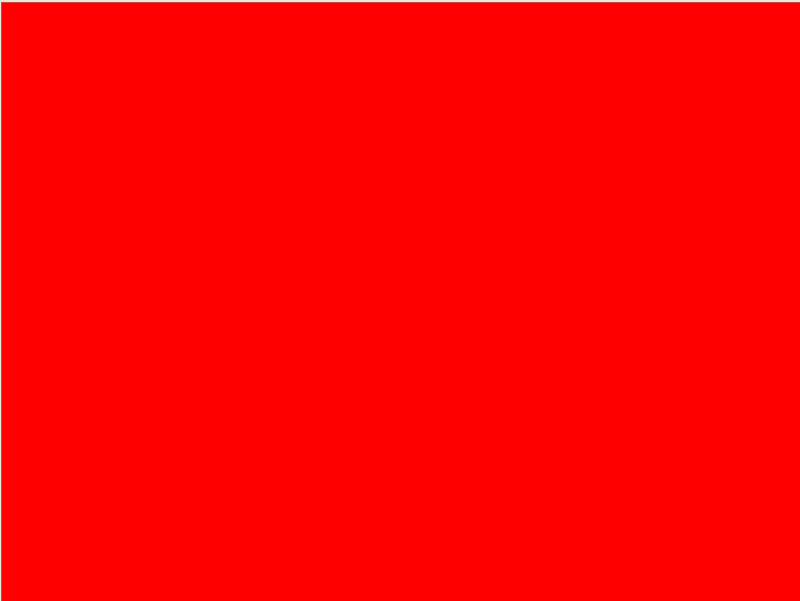
# 一

### 实验内容

创建一个800\*600的图像，全部为红色

图1 800\*600全红图像



### 实验效果：



### 实现过程：

首先创建一个800\*600的黑色窗口，然后将所有的红色通道值改为255

img = np.zeros((600, 800, 3), np.uint8)  
img[:, :, 2] = 255

# 二

### 实验内容：

创建一个800\*600的图像，以40\*30的方框红蓝交替，如图2所示：

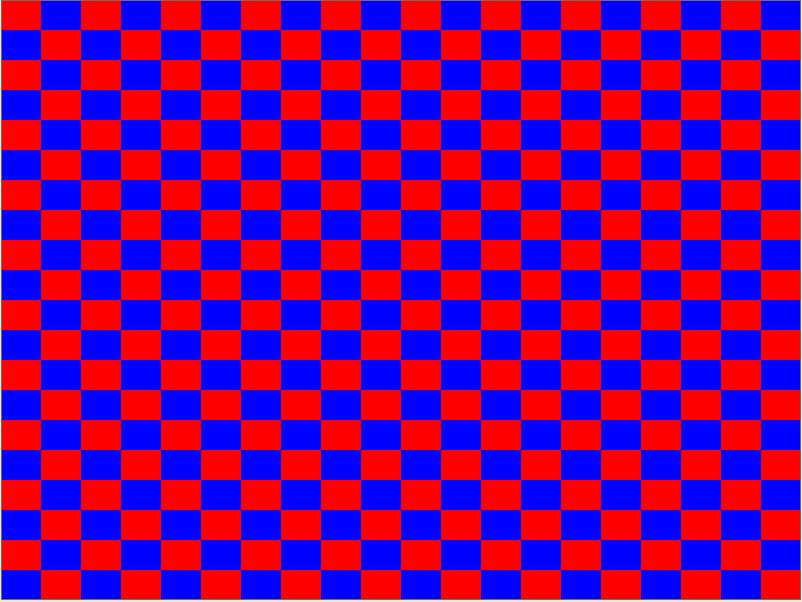


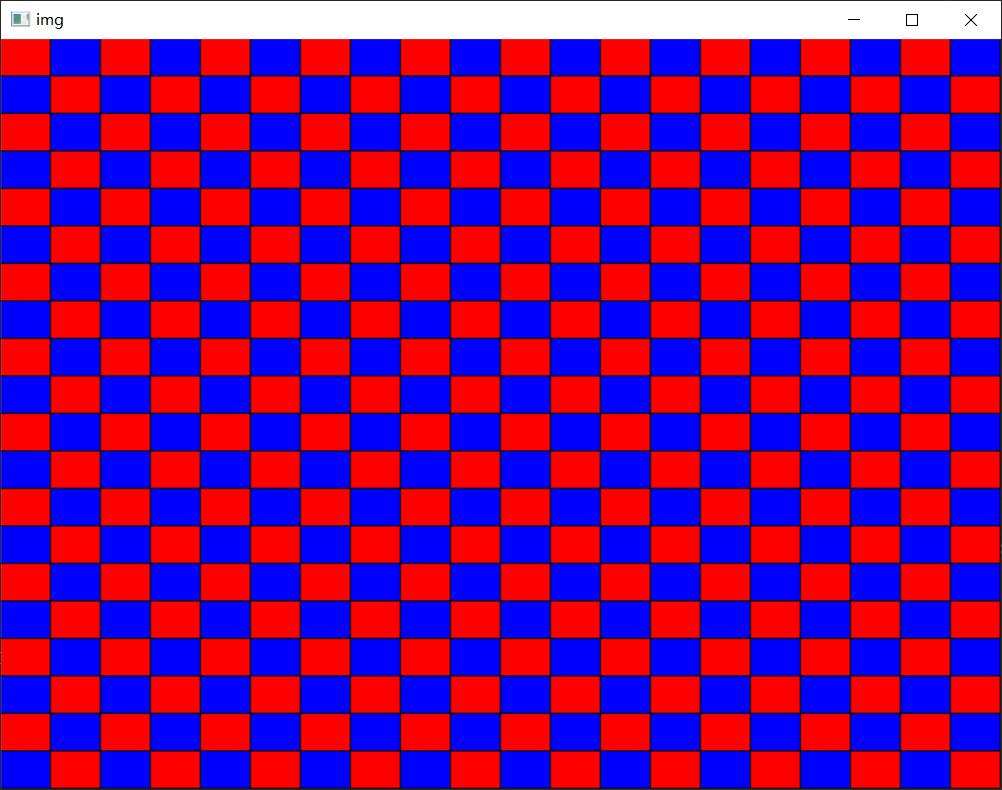
图2 800\*600红蓝交替图像

要求：

图像大小：800\*600

图像中方格大小：40\*30

### 实验效果：



### 实现过程：

首先创建一个800\*600黑色窗口，利用循环判断修改对应ROI区域的RGB值

while i < 600:  
 while j < 800:  
 if k:  
 img[i:i + 29, j:j + 39, 2] = 255  
 else:  
 img[i:i + 29, j:j + 39, 0] = 255  
 k = not k  
 j += 40  
 k = not k  
 j = 0  
 i += 30

# 三

### 实验内容：

完成从彩色图像到灰度图像的转换



图3 从彩色图像到灰度图像的转换

### 实现效果：

### 实现过程：

方法1:

首先读取彩色图片，获取每个像素点对应RGB值，利用公式gray = 0.299 \* r + 0.587 \* g + 0.114 \* b将对应值修改为灰度值。

方法2：

利用函数

img\_gray = cv.cvtColor(img, cv.COLOR\_BGR2GRAY)

直接进行颜色转化

# 四：

### 实验内容：

完成从灰度图像到黑白图的转换



图4从灰度图像到黑白图像的转换（阈值取100）



### 实现效果：

### 实现过程：

取灰度图每个像素点的灰度值进行比较，大于100则将值改为255，反之改为0

# 五

### 实验内容：

在4的基础上，实现原来的灰度图像，大于阈值的，设为红色；小于阈值的，设为蓝色

### 实现效果：



### 实现过程：

取灰度图每个像素点的灰度值进行比较，大于100则将红色通道值改为255，其余通道为0，反之将蓝色通道值改为255，其余为0

# 六

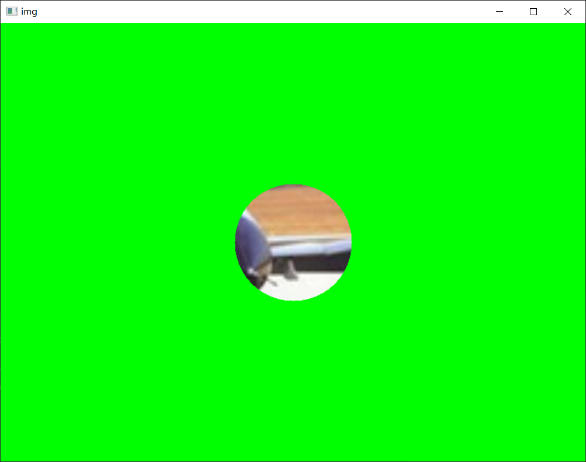
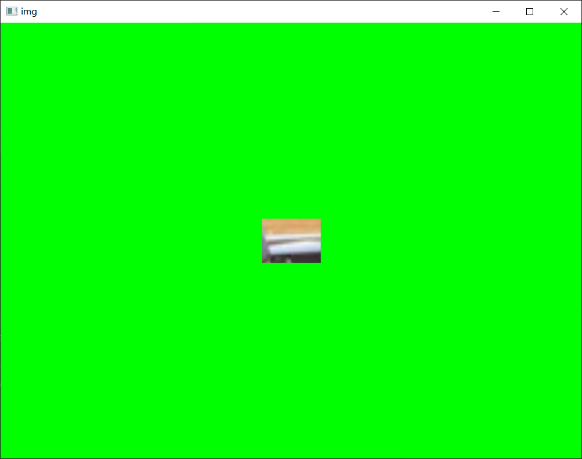
### 实验内容：

打开一副图像，取出以原图像为中心，原图像大小的十分之一的子图像。原来图像：800\*600-->80\*60的图像。

a）以方框的形式显示

b）以源图像为中心，半径为原图像十分之一的图像(原图像大小取宽度和高度中的最小值)。显示一个圆，圆的外面用绿色的背景

### 实现效果：



### 实现过程：

首先创建800\*600的黑色图像，将中间80\*60区域改为白色，在与原图像进行与操作，最后遍历图像将所有黑色像素点值改为绿色。

mask = np.zeros((600, 800, 3), np.uint8)  
cv.circle(mask, (400, 300), 80, (255, 255, 255), -1)  
img = cv.imread("picture/pic1.jpg")  
img = cv.resize(img, (800, 600))  
img = cv.bitwise\_and(img, mask)