NUST
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

Digital Image Processing

Lab 2

Submitted by:

Name:	Reg:
Furqan Ahmad	352076

Submitted to:

Course Inst:	Lab Eng:
Dr Usman Akram	Sundas Ashraf

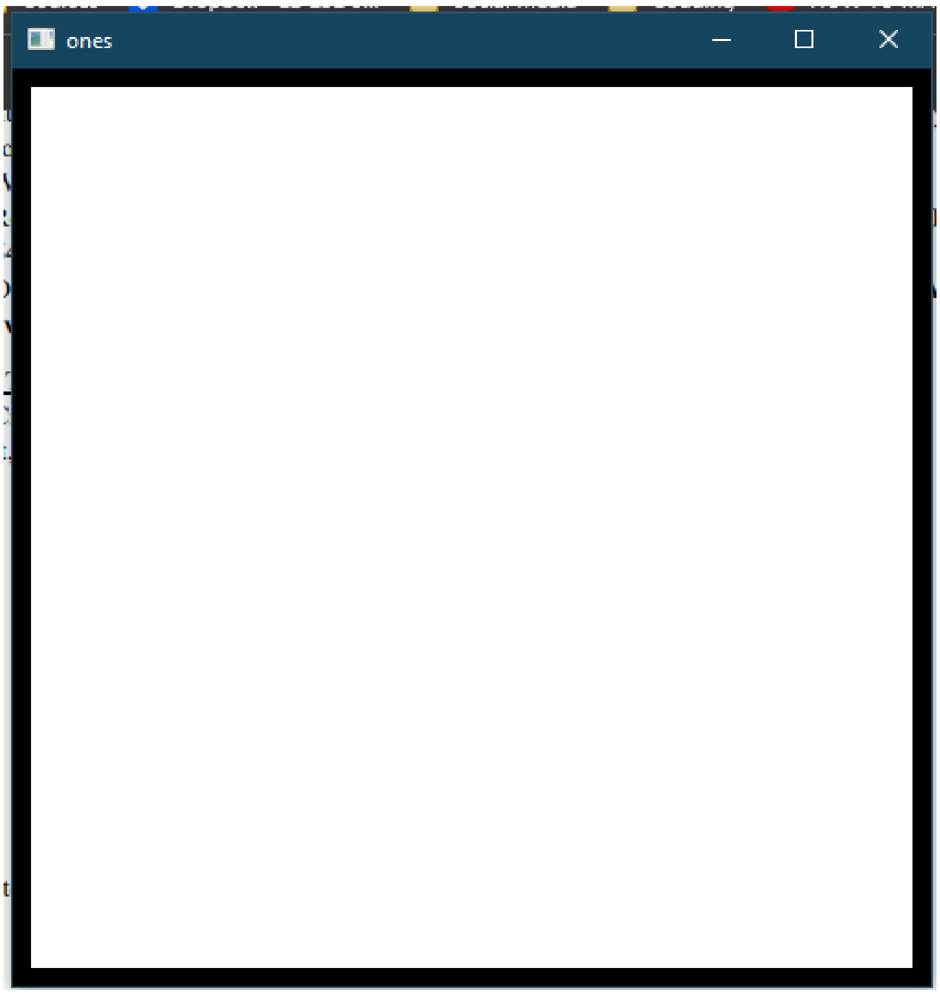
Github:Furqan3/Digital-Image-Processing (github.com)

Date:21,Feb 2023

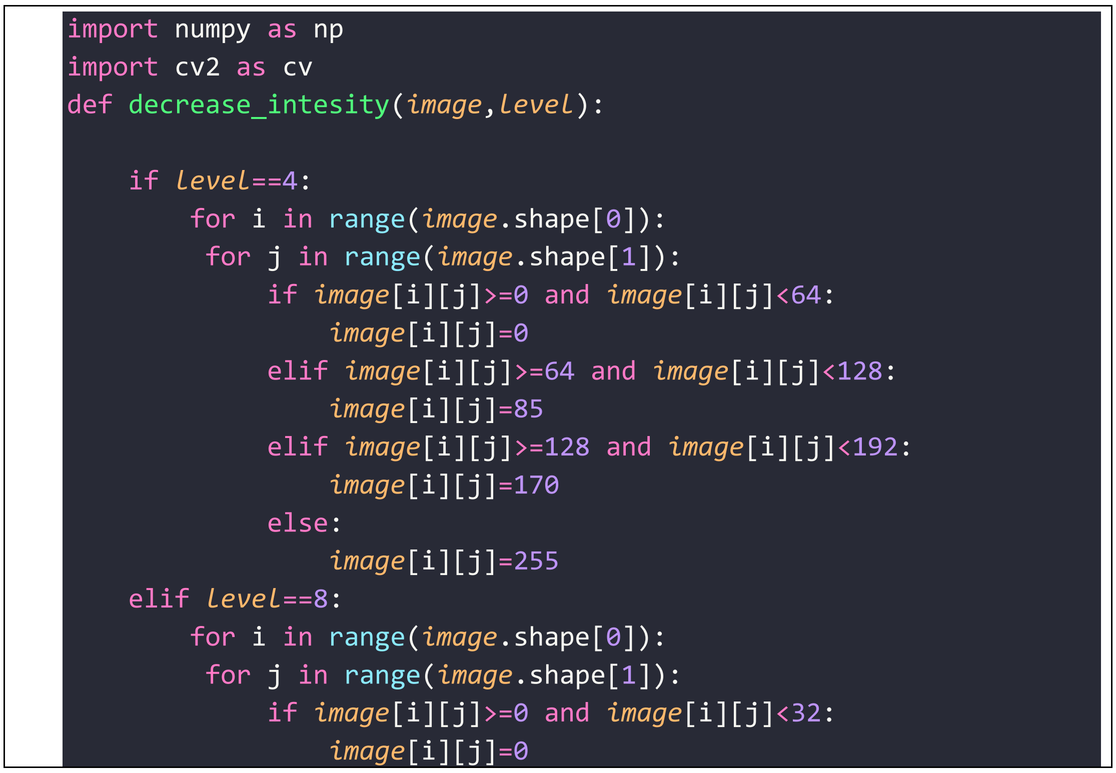
1 Code:

```python
import numpy as np
import cv2 as cv
def
createing_rectangle(size=(500,500),pading=10,background_color=(255
,255,255),stock_color=0):
    one=np.ones((size),dtype=np.uint8)
    one[:,:]=background_color
    x=size[0]
    y=size[1]
    one[:,:pading]=stock_color
    one[:,y-pading:]=stock_color
    one[:pading,:]=stock_color
    one[x-pading:,:]=stock_color
    cv.imshow('ones',one)
    cv.imwrite('myimage.png',one)
    cv.waitKey()
createing_rectangle((500,500,3),10,(255,255,255),(0,0,0))
```

Output:

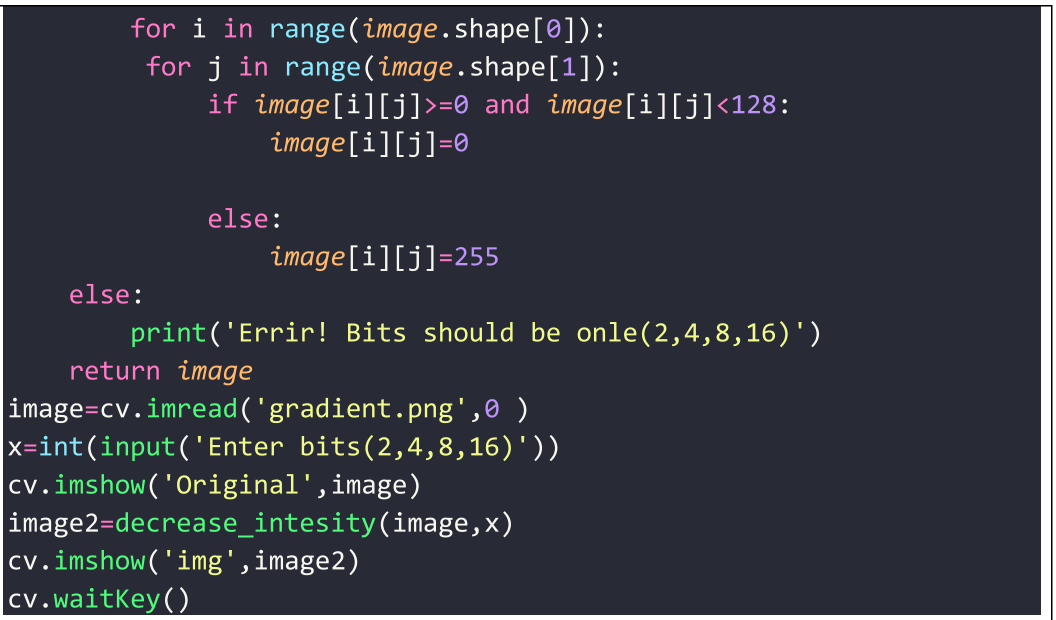


2 Code:



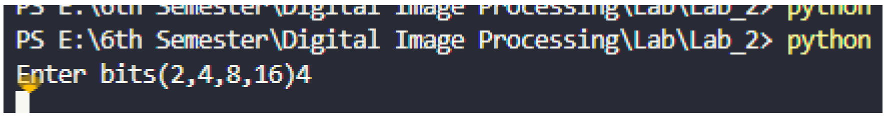
```
import numpy as np
import cv2 as cv
def decrease_intesity(image,level):

    if level==4:
        for i in range(image.shape[0]):
         for j in range(image.shape[1]):
            if image[i][j]>=0 and image[i][j]<64:
                image[i][j]=0
            elif image[i][j]>=64 and image[i][j]<128:
                image[i][j]=85
            elif image[i][j]>=128 and image[i][j]<192:
                image[i][j]=170
            else:
                image[i][j]=255
    elif level==8:
        for i in range(image.shape[0]):
         for j in range(image.shape[1]):
            if image[i][j]>=0 and image[i][j]<32:
                image[i][j]=0
```

```
             elif image[i][j]>=32 and image[i][j]<64:
                 image[i][j]=36
             elif image[i][j]>=64 and image[i][j]<64:
                 image[i][j]=72
             elif image[i][j]>=64 and image[i][j]<96:
                 image[i][j]=108
             elif image[i][j]>=96 and image[i][j]<128:
                 image[i][j]=144
             elif image[i][j]>=128 and image[i][j]<160:
                 image[i][j]=180
             elif image[i][j]>=160 and image[i][j]<192:
                 image[i][j]=216
             else:
                 image[i][j]=255
    elif level==16:
        for i in range(image.shape[0]):
         for j in range(image.shape[1]):
             if image[i][j]>=0 and image[i][j]<16:
                 image[i][j]=0
             elif image[i][j]>=16 and image[i][j]<32:
                 image[i][j]=24
             elif image[i][j]>=32 and image[i][j]<48:
                 image[i][j]=40
             elif image[i][j]>=48 and image[i][j]<64:
                 image[i][j]=56
             elif image[i][j]>=64 and image[i][j]<80:
                 image[i][j]=72
             elif image[i][j]>=80 and image[i][j]<96:
                 image[i][j]=88
             elif image[i][j]>=96 and image[i][j]<112:
                 image[i][j]=104
             elif image[i][j]>=112 and image[i][j]<128:
                 image[i][j]=120
             elif image[i][j]>=128 and image[i][j]<144:
                 image[i][j]=136
             elif image[i][j]>=144 and image[i][j]<160:
                 image[i][j]=152
             elif image[i][j]>=160 and image[i][j]<176:
                 image[i][j]=168
             elif image[i][j]>=176 and image[i][j]<192:
                 image[i][j]=184
             elif image[i][j]>=192 and image[i][j]<208:
                 image[i][j]=200
             elif image[i][j]>=208 and image[i][j]<224:
                 image[i][j]=216
             else:
                 image[i][j]=255
    elif level==2:
```

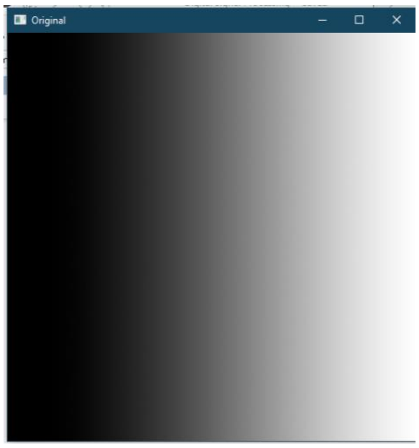


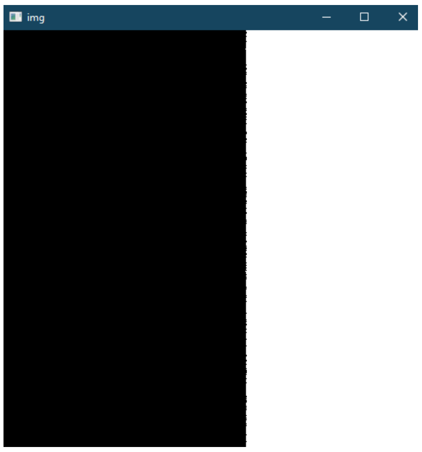
```python
        for i in range(image.shape[0]):
         for j in range(image.shape[1]):
             if image[i][j]>=0 and image[i][j]<128:
                 image[i][j]=0

             else:
                 image[i][j]=255
    else:
        print('Errir! Bits should be onle(2,4,8,16)')
    return image
image=cv.imread('gradient.png',0 )
x=int(input('Enter bits(2,4,8,16)'))
cv.imshow('Original',image)
image2=decrease_intesity(image,x)
cv.imshow('img',image2)
cv.waitKey()
```

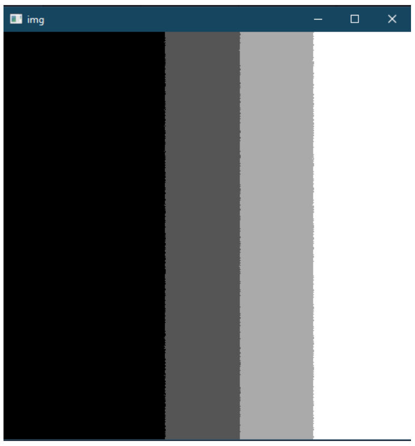
Output:



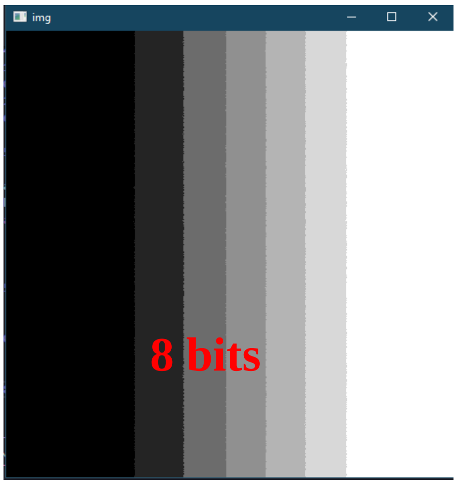
```
PS E:\6th Semester\Digital Image Processing\Lab\Lab_2> python
PS E:\6th Semester\Digital Image Processing\Lab\Lab_2> python
Enter bits(2,4,8,16)4
```



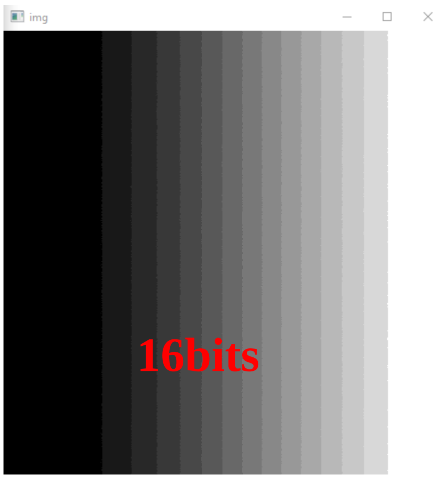




Original **2 bits** **4 bits**



8 bits

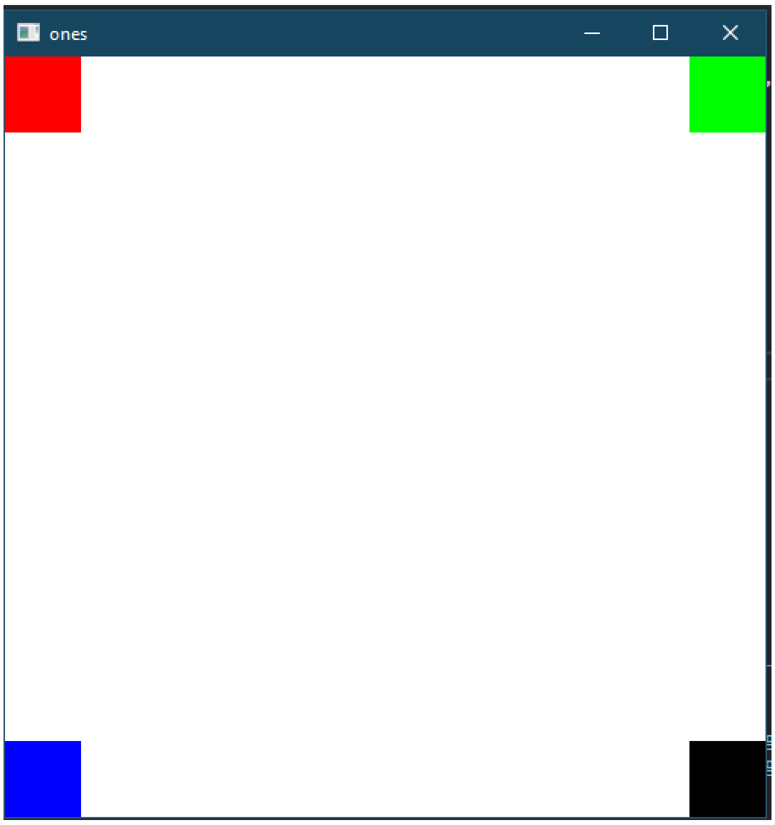


16bits

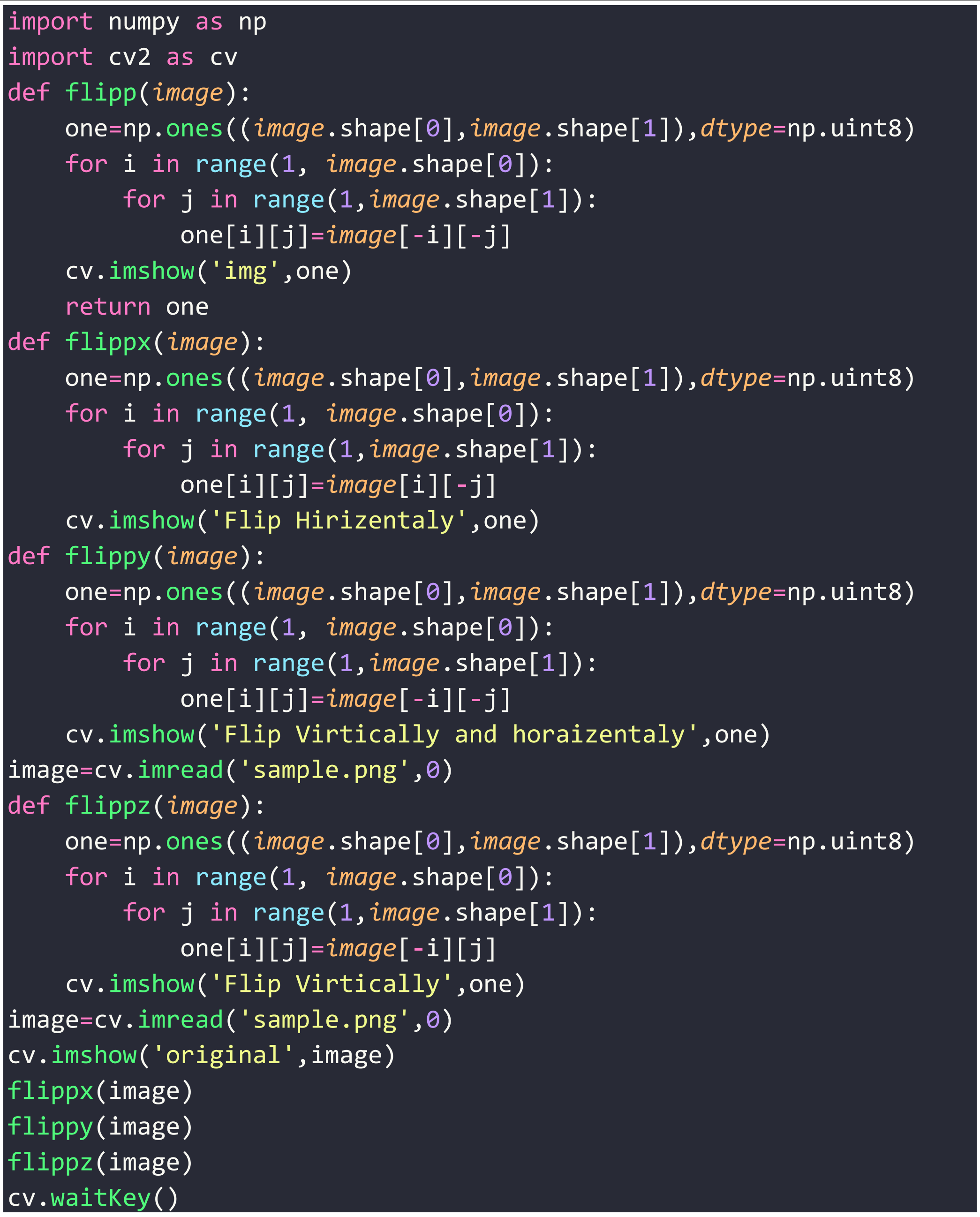
3 Code:

```
import numpy as np
import cv2 as cv
def
createing_rectangle(size=(500,500),pading=10,background_color=(255
,255,255)):
    one=np.ones((size),dtype=np.uint8)
    one[:,:]=background_color
    x=size[0]
    y=size[1]
    one[:pading,:pading]=(0,0,255)
    one[:pading,y-pading:]=(0,255,0)
    one[x-pading:,:pading]=(255,0,0)
    one[x-pading:,y-pading:]=(0,0,0)
    cv.imshow('ones',one)
    cv.imwrite('myimage.png',one)
    cv.waitKey()
createing_rectangle((500,500,3),50,(255,255,255))
```

Output:

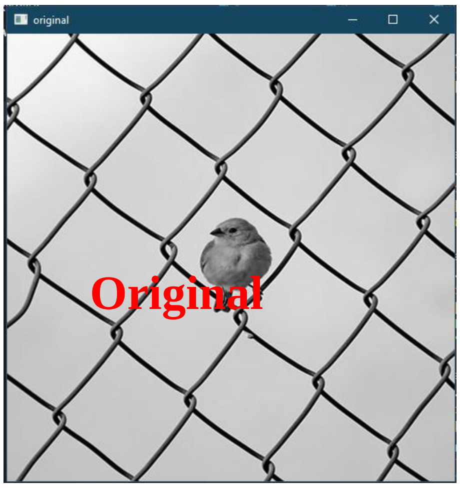


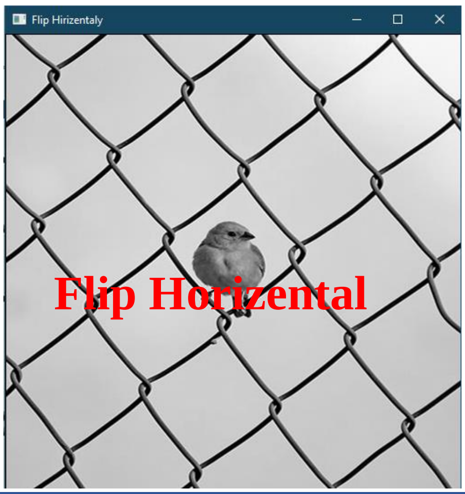
4 Code:

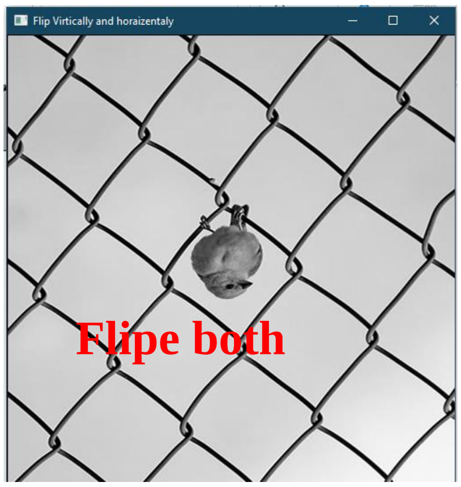


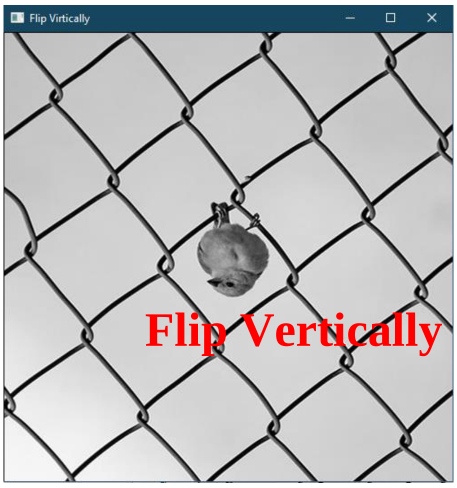
```python
import numpy as np
import cv2 as cv
def flipp(image):
    one=np.ones((image.shape[0],image.shape[1]),dtype=np.uint8)
    for i in range(1, image.shape[0]):
        for j in range(1,image.shape[1]):
            one[i][j]=image[-i][-j]
    cv.imshow('img',one)
    return one
def flippx(image):
    one=np.ones((image.shape[0],image.shape[1]),dtype=np.uint8)
    for i in range(1, image.shape[0]):
        for j in range(1,image.shape[1]):
            one[i][j]=image[i][-j]
    cv.imshow('Flip Hirizentaly',one)
def flippy(image):
    one=np.ones((image.shape[0],image.shape[1]),dtype=np.uint8)
    for i in range(1, image.shape[0]):
        for j in range(1,image.shape[1]):
            one[i][j]=image[-i][-j]
    cv.imshow('Flip Virtically and horaizentaly',one)
image=cv.imread('sample.png',0)
def flippz(image):
    one=np.ones((image.shape[0],image.shape[1]),dtype=np.uint8)
    for i in range(1, image.shape[0]):
        for j in range(1,image.shape[1]):
            one[i][j]=image[-i][j]
    cv.imshow('Flip Virtically',one)
image=cv.imread('sample.png',0)
cv.imshow('original',image)
flippx(image)
flippy(image)
flippz(image)
cv.waitKey()
```

Output:





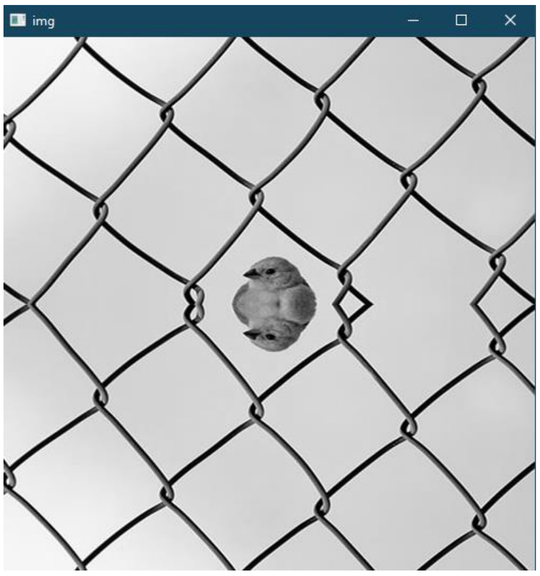




5 Code:

```
import numpy as np
import cv2 as cv
def flipp(image):
    for i in range(0, image.shape[0]):
        image[-i,:],image[i,:]=image[i,:],image[-i,:]
    return image
image=cv.imread('sample.png',0)
cv.imshow('original',image)
cv.waitKey()
flipimg=flipp(image)
cv.imshow('img',flipimg)
cv.waitKey()
```

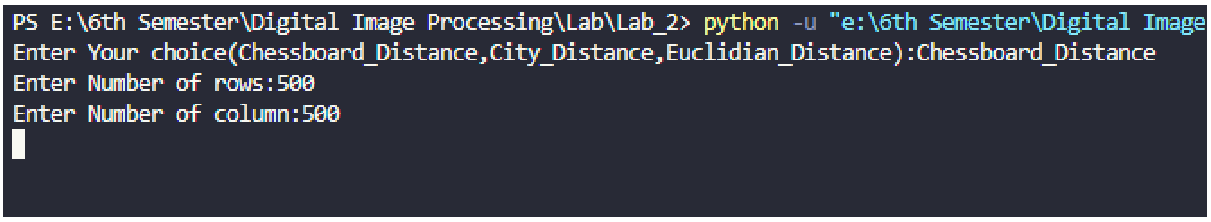
Output:



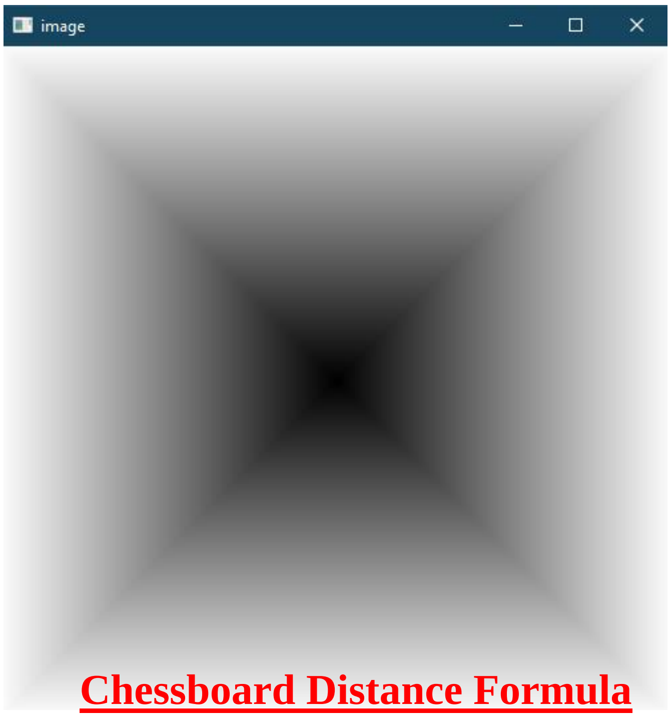
6 Code:

```
import numpy as np
import cv2 as cv

def distance_map(image,formula):
    centre=[int(image.shape[0]/2)+1,int(image.shape[1]/2)+1]
    if formula=='Euclidian_Distance':
        for i in range(image.shape[0]):
            for j in range(image.shape[1]):
               image[i][j]=((centre[0]-i)**2+(centre[1]-j)**2)**.5
               if ((centre[0]-i)**2+(centre[1]-j)**2)**.5>255:
                   image[i][j]=255

    elif formula=='City_Distance':
        for i in range(image.shape[0]):
            for j in range(image.shape[1]):
               image[i][j]=np.abs((centre[0]-
i))+np.abs((centre[1]-j))
               if np.abs((centre[0]-i))+np.abs((centre[1]-j))>255:
                   image[i][j]=255

    elif formula=='Chessboard_Distance':
        for i in range(image.shape[0]):
            for j in range(image.shape[1]):
                if np.abs((centre[0]-i))>np.abs((centre[1]-j)):
                    image[i][j]=np.abs((centre[0]-i))
                    if np.abs((centre[0]-i))>255:
                        image[i][j]=255
                else:
                    image[i][j]=np.abs((centre[1]-j))
                    if np.abs((centre[1]-j))>255:
                        image[i][j]=255
    else:
        print('Error! You are allowed to select the given
choices!')
        return None
    cv.imshow('image',image)
    cv.waitKey()
choice=input('Enter Your
choice(Chessboard_Distance,City_Distance,Euclidian_Distance):')
x=int(input('Enter Number of rows:'))
y=int(input('Enter Number of column:'))
image=np.zeros((x,y),np.uint8)
distance_map(image,choice)
```

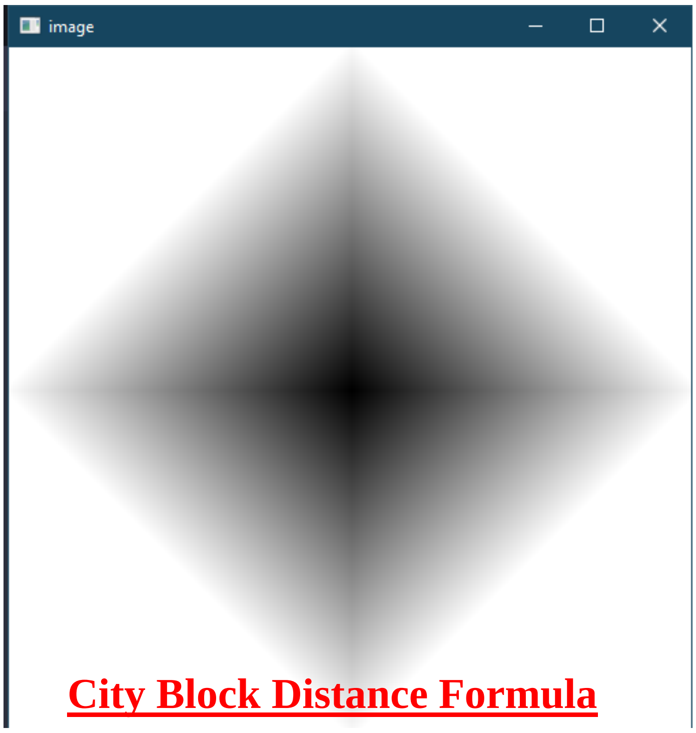
Output:



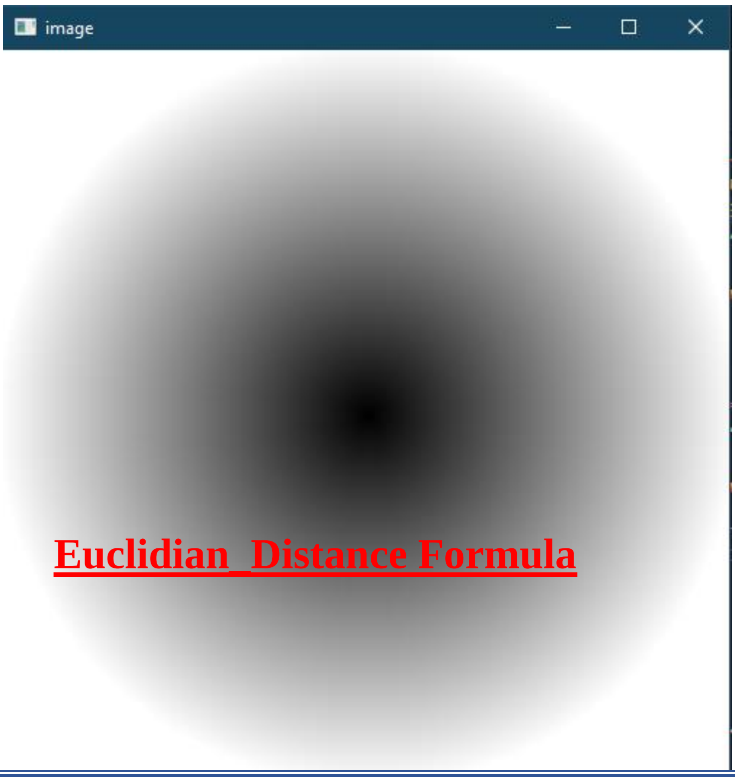
```
PS E:\6th Semester\Digital Image Processing\Lab\Lab_2> python -u "e:\6th Semester\Digital Image
Enter Your choice(Chessboard_Distance,City_Distance,Euclidian_Distance):Chessboard_Distance
Enter Number of rows:500
Enter Number of column:500
```



Chessboard Distance Formula



City Block Distance Formula

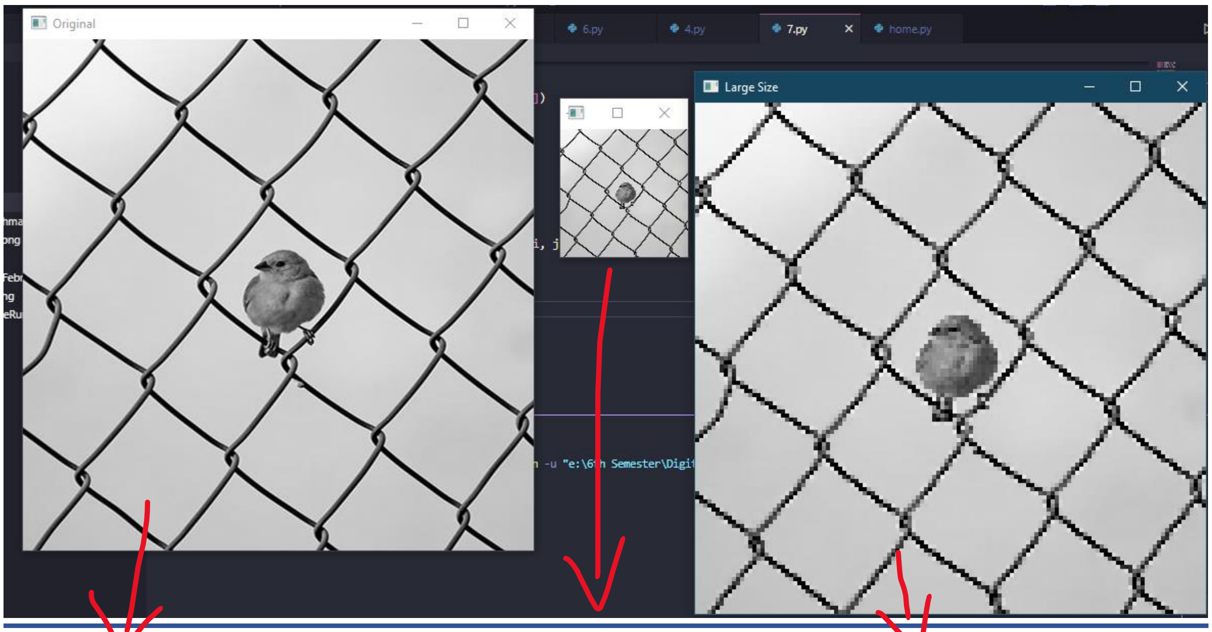


Euclidian_Distance Formula

7 Code:

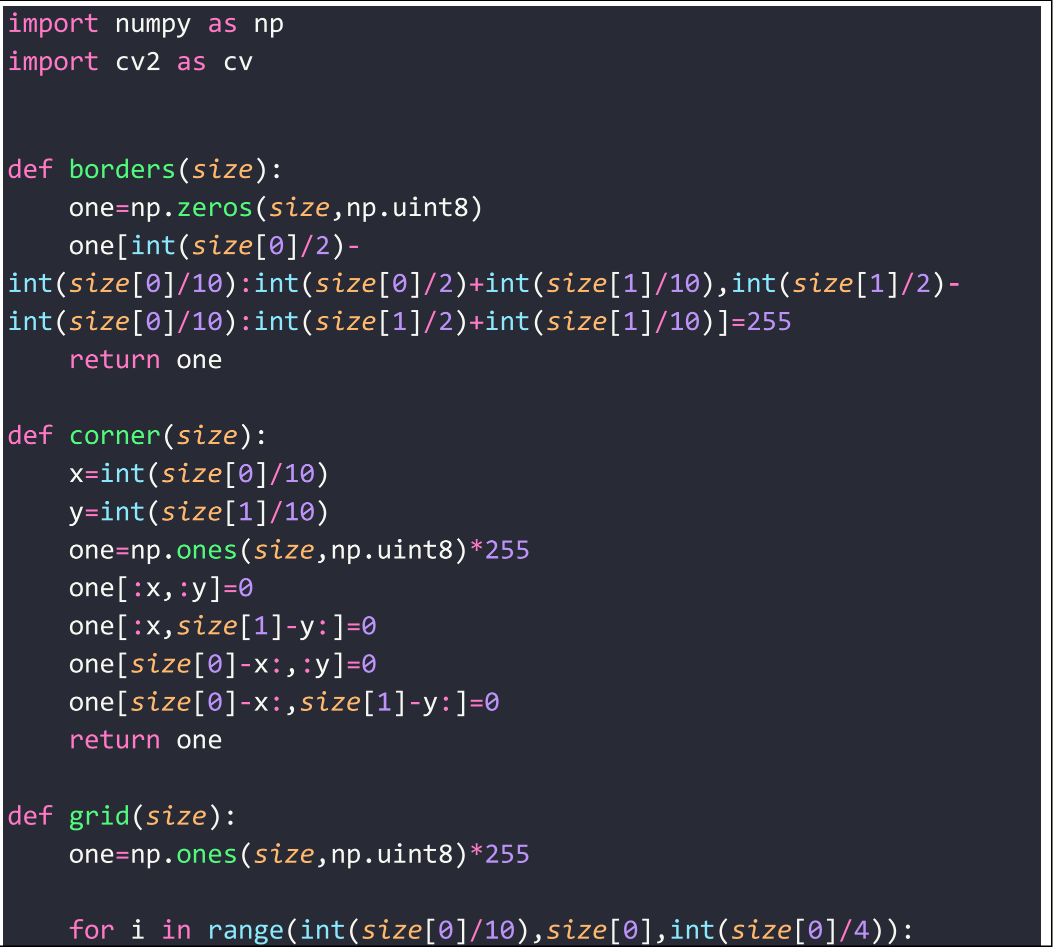
```python
import numpy as np
import cv2 as cv

def resize(image):
    x=int(image.shape[0]/4)
    y=int(image.shape[1]/4)
    one=np.ones((x,y),np.uint8)
    for i in range(0,image.shape[0],4):
        for j in range(0,image.shape[1],4):
            one[int(i/4)][int(j/4)]=int(image[i][j])
    return one

def increase(image):
    x=int(image.shape[0]*4)
    y=int(image.shape[1]*4)
    one=np.ones((x,y),np.uint8)
    for i in range(0,image.shape[0]):
        for j in range(0,image.shape[1]-1):
            one[i*4][j*4]=image[i][j]
            a=image[i][j+1]
            b=image[i][j]
            x=int((a+b)/2)
            one[(i*4)+3][(j*4)+3]=a
            one[(i*4)+2][(j*4)+2]=x
            one[(i*4)+1][(j*4)+1]=b
    return one

image=cv.imread('sample2.png',0)
cv.imshow('Original',image)
image2=resize(image)
cv.imshow('Small Size',image2)
image3=increase(image2)
cv.imshow('Large Size',image3)
cv.waitKey()
```

Output:



Original Size **Small Size** **Resize Large**

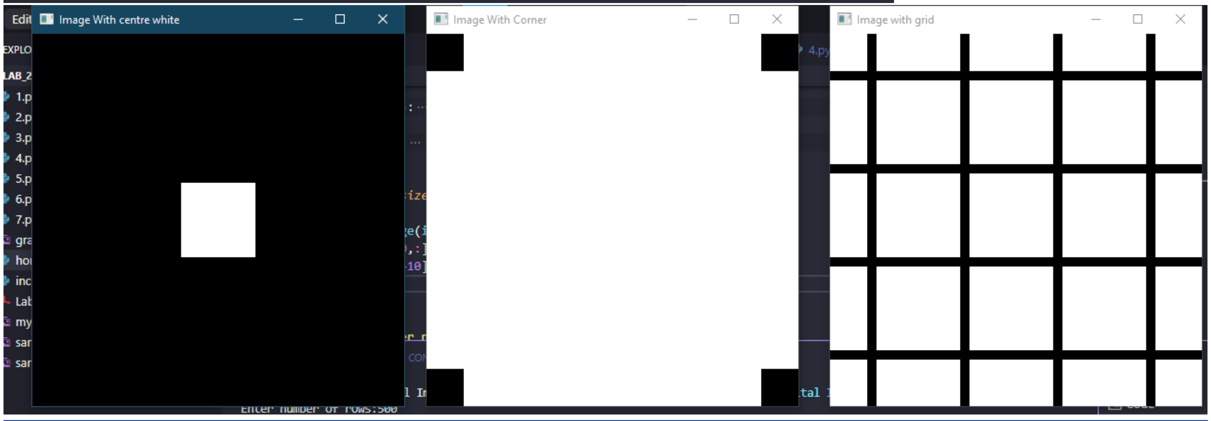
8 Code:



```python
import numpy as np
import cv2 as cv


def borders(size):
    one=np.zeros(size,np.uint8)
    one[int(size[0]/2)-
int(size[0]/10):int(size[0]/2)+int(size[1]/10),int(size[1]/2)-
int(size[0]/10):int(size[1]/2)+int(size[1]/10)]=255
    return one

def corner(size):
    x=int(size[0]/10)
    y=int(size[1]/10)
    one=np.ones(size,np.uint8)*255
    one[:x,:y]=0
    one[:x,size[1]-y:]=0
    one[size[0]-x:,:y]=0
    one[size[0]-x:,size[1]-y:]=0
    return one

def grid(size):
    one=np.ones(size,np.uint8)*255

    for i in range(int(size[0]/10),size[0],int(size[0]/4)):
        one[i:i+10,:]=0
        one[:,i:i+10]=0

    return one

x=int(input('Enter number of rows:'))
y=int(input('Enter number of colums:'))

cv.imshow('Image With centre white',borders((x,y)))
cv.imshow('Image With Corner',corner((x,y)))
cv.imshow('Image with grid',grid((x,y)))
cv.waitKey()
```

Output:

```
PS E:\6th Semester\Digital Image Processing\Lab\Lab_2> python
Enter number of rows:400
Enter number of colums:400
```



White center Black Corner Grid