**Digital Image Processing**

**Lab-05**

**Name:** Ahmad Amjad Mughal

**Reg No:** 121672

**Class**: BSCS-6C

**Task 1**

**Code**

from PIL import Image

import PIL.ImageOps

#Defines Function that invert RGB Image

def RGBImage(image):

converted = image.convert('RGB')

converted.save('RGBImage.jpg')

im = converted.load()

#traverseing through each loaction and do 255 - pixel for each R, G, B value which is negative of a pixel

for row in range(converted.size[0]):

for column in range(converted.size[1]):

red,blue,green = im[row,column][0],im[row,column][1],im[row,column][2]

red,blue,green = abs(255-red),abs(255-blue),abs(255-green)

im[row,column] = (red,blue,green)

converted.show();

converted.save('RGBInvertImage.jpg')

#Defines Function that invert Binary Image

def BinaryImage(image):

converted = image.convert('1')

converted.save('BinaryImage.jpg')

im = converted.load()

#traversing through each location and invert the pixel's value

for row in range(converted.size[0]):

for column in range(converted.size[1]):

if im[row,column] == 1:

im[row,column] = 0

else:

im[row,column] = 1 - im[row,column]

converted.show();

converted.save('BinaryInvertImage.jpg')

#Defines Function that invert Grey Image

def GreyImage(image):

converted = image.convert('L')

converted.save('GreyImage.jpg')

im = converted.load()

#traversing through each location and invert each grey sacle pixel's value

for row in range(converted.size[0]):

for column in range(converted.size[1]):

currentPixel = im[row,column]

im[row,column] = 255- currentPixel

converted.show();

converted.save('GreyInvertImage.jpg')

image = Image.open('lena\_color.gif')

invertedImage1 = RGBImage(image)

invertedImage2 = GreyImage(image)

invertedImage3 = BinaryImage(image)

**Screenshot**

**RGB Original Image RGB Inverted Image**



**Binary Original Image binary Inverted Image**



**Grey Original Image Inverted Grey Image**



**Task 2**

**Code**

from PIL import Image

import numpy as np

#Read original image. Convert it to grayscale

originalImage = Image.open("lena\_color.gif")

convertedImage = originalImage.convert('L')

#Load the image pixels in the form of (x,y) coardinates

pixels = convertedImage.load()

#Assiging these coardinates to width and height

width, height = convertedImage.size

#traversing through each pixel and when there's some edge formula for gradient applies to it

for row in range(width):

for column in range(height):

if( row < width - 1):

pixels[row,column] = pixels[row + 1,column] - pixels[row,column]

else:

pixels[row,column] - pixels[row,column]

#Save and show the converted image

convertedImage.save('lenna\_gradient.png')

convertedImage.show()

**Screenshot**

****

**Task 3**

**Code**

#Including necessary libraries

from PIL import Image

#load image pixels for each bitslice operation

image1 = Image.open("dollar.tif")

pix1 = image1.load()

image2 = Image.open("dollar.tif ")

pix2 = image2.load()

image3 = Image.open("dollar.tif")

pix3 = image3.load()

image4 = Image.open("dollar.tif")

pix4 = image4.load()

image5 = Image.open("dollar.tif")

pix5 = image5.load()

image6 = Image.open("dollar.tif")

pix6 = image6.load()

image7 = Image.open("dollar.tif")

pix7 = image7.load()

image8 = Image.open("dollar.tif")

pix8 = image8.load()

#Assigning coardinates (x,y) to height and width where xx determines w and y determines h

w, height = image1.size

#For BitSlice 1

for x in range(0, width):

for y in range(0, height):

a = image1.getpixel((x, y))

a = a & 1

pix1[x, y] = a

image1.save("bitslicing1.png")

#For BitSlice 2

for x in range(0, width):

for y in range(0, height):

a = image2.getpixel((x, y))

a = a & 2

pix2[x, y] = a

image2.save("bitslicing2.png")

#For BitSlice 3

for x in range(0, width):

for y in range(0, height):

a = image3.getpixel((x, y))

a = a & 4

pix3[x, y] = a

image3.save("bitslicing3.png")

#For BitSlice 4

for x in range(0, width):

for y in range(0, height):

a = image4.getpixel((x, y))

a = a & 8

pix4[x, y] = a

image4.save("bitslicing4.png")

#For BitSlice 5

for x in range(0, width):

for y in range(0, height):

a = image5.getpixel((x, y))

a = a & 16

pix5[x, y] = a

image5.save("bitslicing5.png")

#For BitSlice 6

for x in range(0, width):

for y in range(0, height):

a = image6.getpixel((x, y))

a = a & 32

pix6[x, y] = a

image6.save("bitslicing6.png")

#For BitSlice 7

for x in range(0, width):

for y in range(0, height):

a = image7.getpixel((x, y))

a = a & 64

pix7[x, y] = a

image7.save("bitslicing7.png")

#For BitSlice 8

for x in range(0, width):

for y in range(0, height):

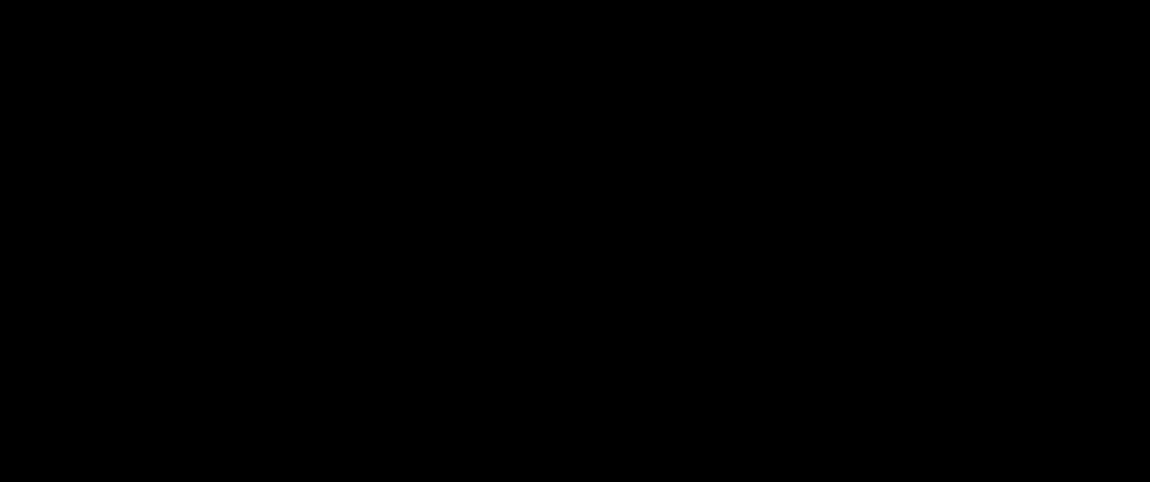
a = image8.getpixel((x, y))

a = a & 128

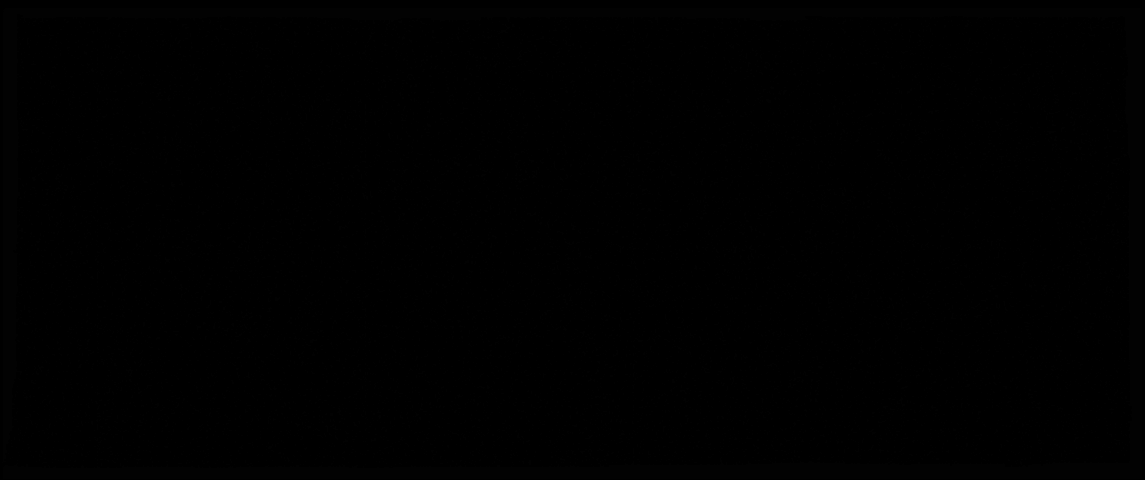
pix8[x, y] = a

image8.save("bitslicing8.png")

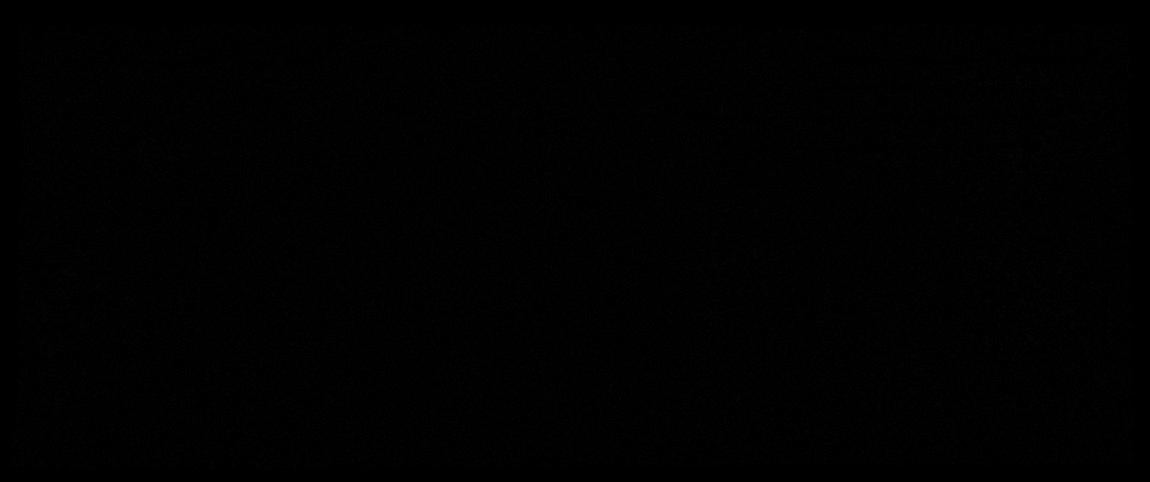
**BitSlicing 1**



**BitSlicing2**



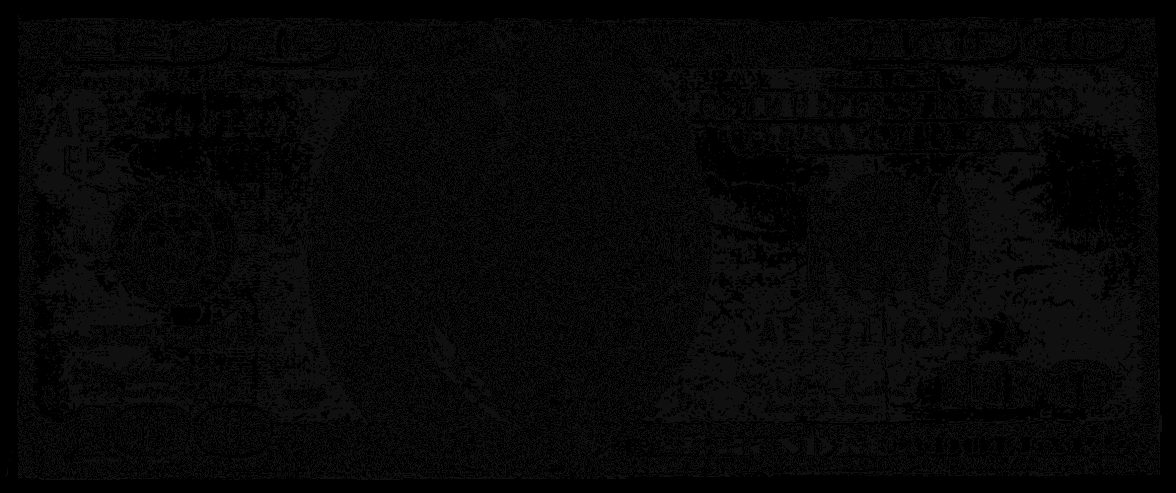
**BitSlicing3**

l

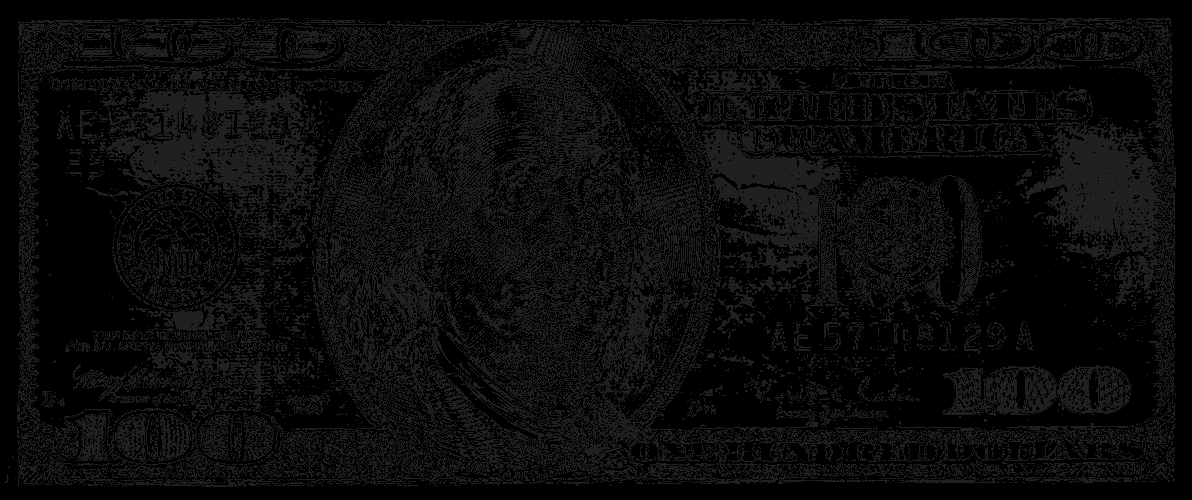
**BitSlicing4**

****

**BitSlicing5**

****

**BitSlicing6**

****

**BitSlicing7**



**BitSlicing8**

