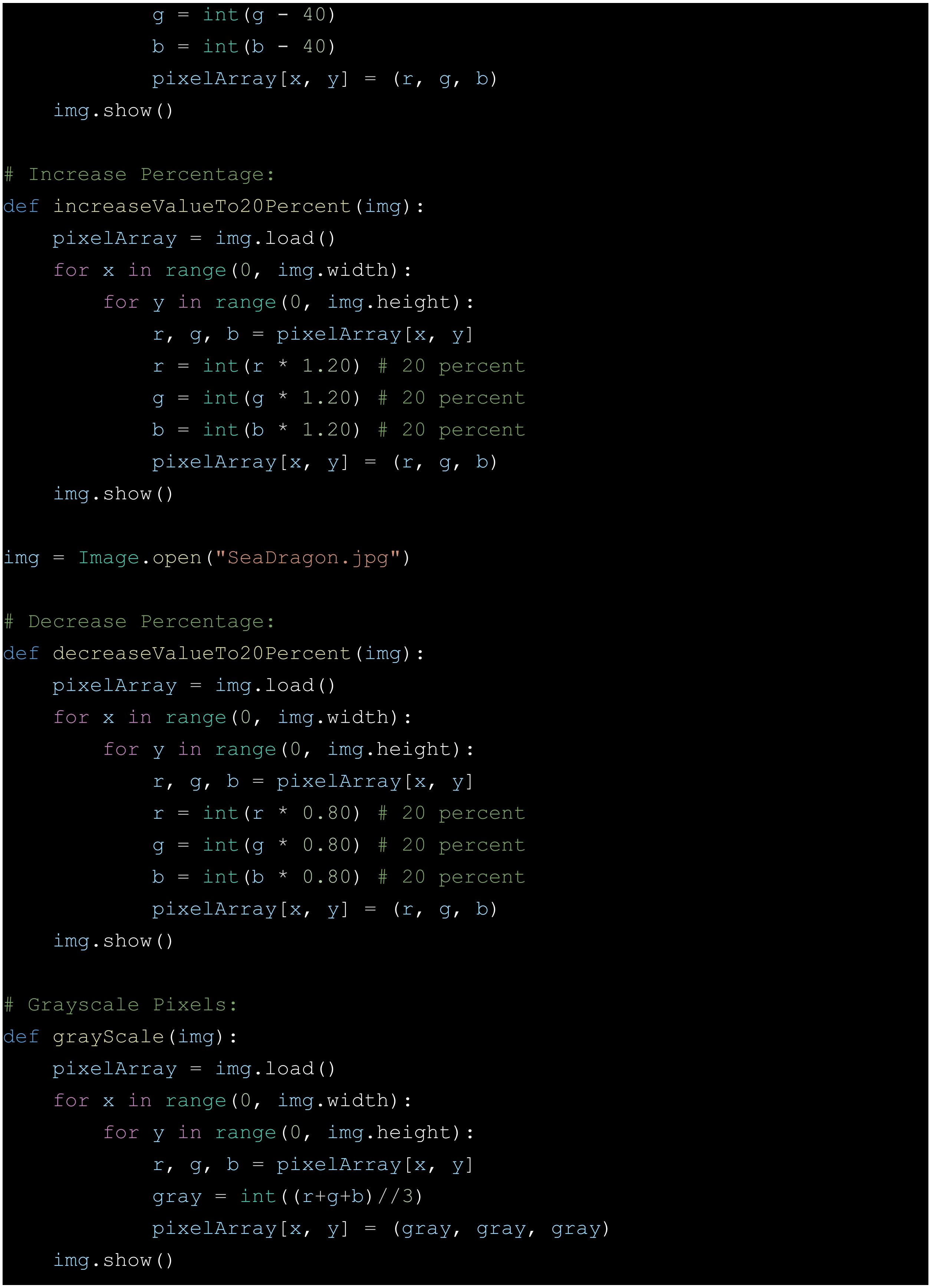
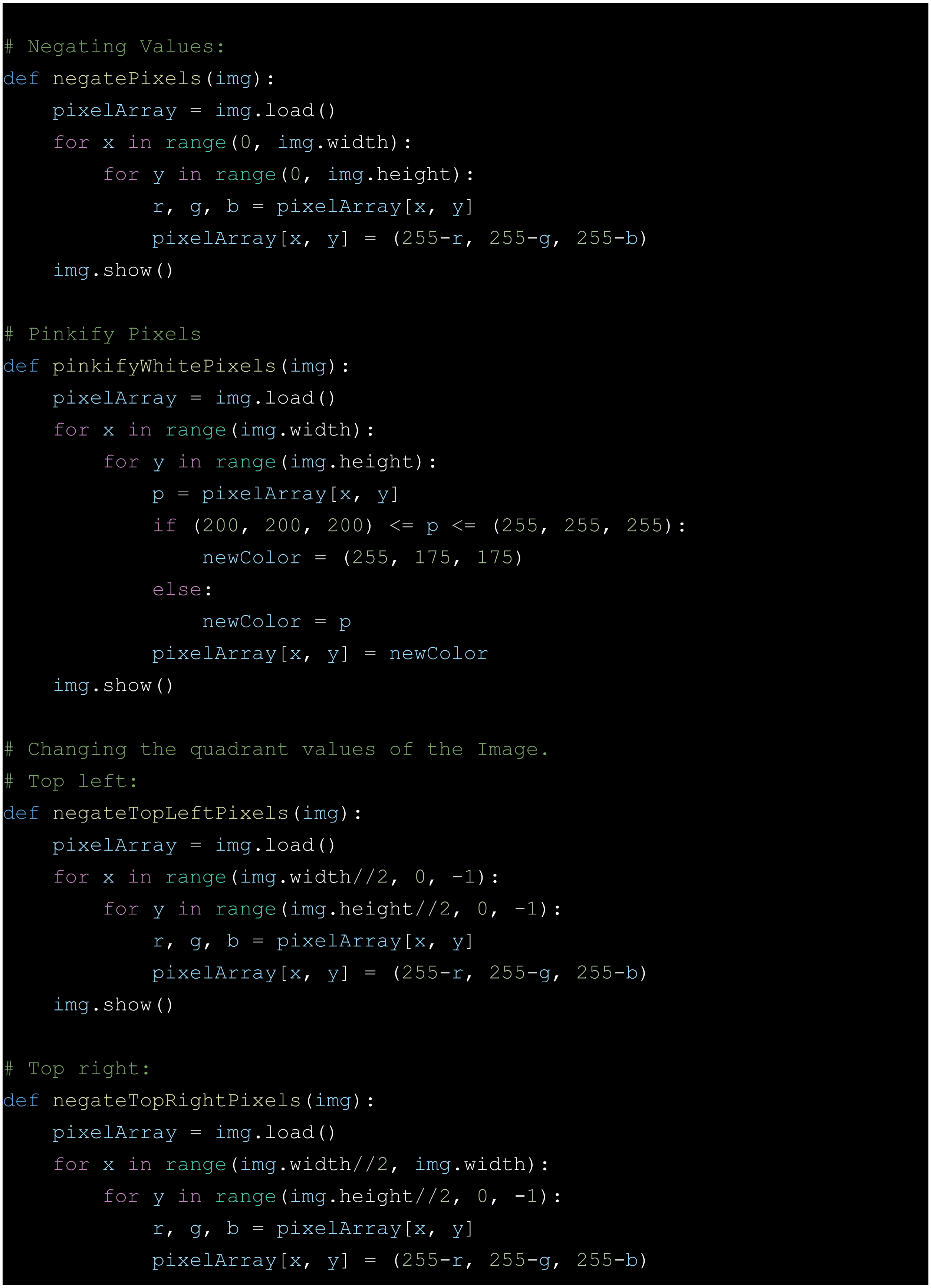
Final Exam Notes

Name: Betero Tiikana
Course: CS101
Photo Codes:

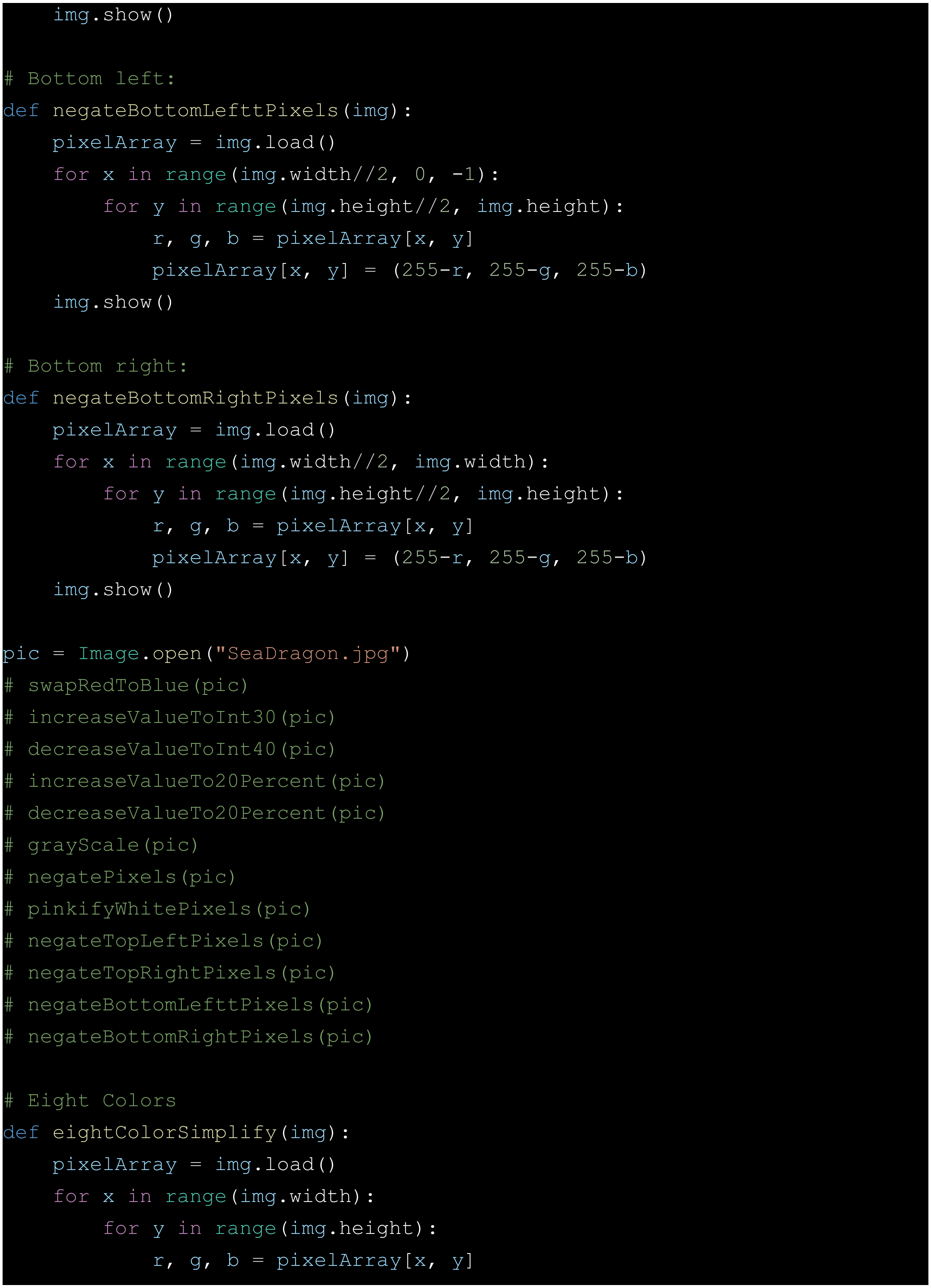
```
from PIL import Image, ImageDraw

# Formulas:
# Increase: Percentage / 100 + 1
# Decrease: Percentage / 100 - 1

# Swapping Colors
def swapRedToBlue(img):
    pixelArray = img.load()
    for x in range(0, img.width):
        for y in range(0, img.height):
            r, g, b = pixelArray[x, y]
            pixelArray[x, y] = (b, g, r)
    img.show()


# Changing the Values
# Increase Integers:
def increaseValueToInt30(img):
    pixelArray = img.load()
    for x in range(0, img.width):
        for y in range(0, img.height):
            r, g, b = pixelArray[x, y]
            r = int(r + 30)
            g = int(g + 30)
            b = int(b + 30)
            pixelArray[x, y] = (r, g, b)
    img.show()

# Decrease Integers:
def decreaseValueToInt40(img):
    pixelArray = img.load()
    for x in range(0, img.width):
        for y in range(0, img.height):
            r, g, b = pixelArray[x, y]
            r = int(r - 40)
```



```
            g = int(g - 40)
            b = int(b - 40)
            pixelArray[x, y] = (r, g, b)
    img.show()

# Increase Percentage:
def increaseValueTo20Percent(img):
    pixelArray = img.load()
    for x in range(0, img.width):
        for y in range(0, img.height):
            r, g, b = pixelArray[x, y]
            r = int(r * 1.20) # 20 percent
            g = int(g * 1.20) # 20 percent
            b = int(b * 1.20) # 20 percent
            pixelArray[x, y] = (r, g, b)
    img.show()

img = Image.open("SeaDragon.jpg")

# Decrease Percentage:
def decreaseValueTo20Percent(img):
    pixelArray = img.load()
    for x in range(0, img.width):
        for y in range(0, img.height):
            r, g, b = pixelArray[x, y]
            r = int(r * 0.80) # 20 percent
            g = int(g * 0.80) # 20 percent
            b = int(b * 0.80) # 20 percent
            pixelArray[x, y] = (r, g, b)
    img.show()

# Grayscale Pixels:
def grayScale(img):
    pixelArray = img.load()
    for x in range(0, img.width):
        for y in range(0, img.height):
            r, g, b = pixelArray[x, y]
            gray = int((r+g+b)//3)
            pixelArray[x, y] = (gray, gray, gray)
    img.show()
```



```python
# Negating Values:
def negatePixels(img):
    pixelArray = img.load()
    for x in range(0, img.width):
        for y in range(0, img.height):
            r, g, b = pixelArray[x, y]
            pixelArray[x, y] = (255-r, 255-g, 255-b)
    img.show()

# Pinkify Pixels
def pinkifyWhitePixels(img):
    pixelArray = img.load()
    for x in range(img.width):
        for y in range(img.height):
            p = pixelArray[x, y]
            if (200, 200, 200) <= p <= (255, 255, 255):
                newColor = (255, 175, 175)
            else:
                newColor = p
            pixelArray[x, y] = newColor
    img.show()

# Changing the quadrant values of the Image.
# Top left:
def negateTopLeftPixels(img):
    pixelArray = img.load()
    for x in range(img.width//2, 0, -1):
        for y in range(img.height//2, 0, -1):
            r, g, b = pixelArray[x, y]
            pixelArray[x, y] = (255-r, 255-g, 255-b)
    img.show()

# Top right:
def negateTopRightPixels(img):
    pixelArray = img.load()
    for x in range(img.width//2, img.width):
        for y in range(img.height//2, 0, -1):
            r, g, b = pixelArray[x, y]
            pixelArray[x, y] = (255-r, 255-g, 255-b)
```



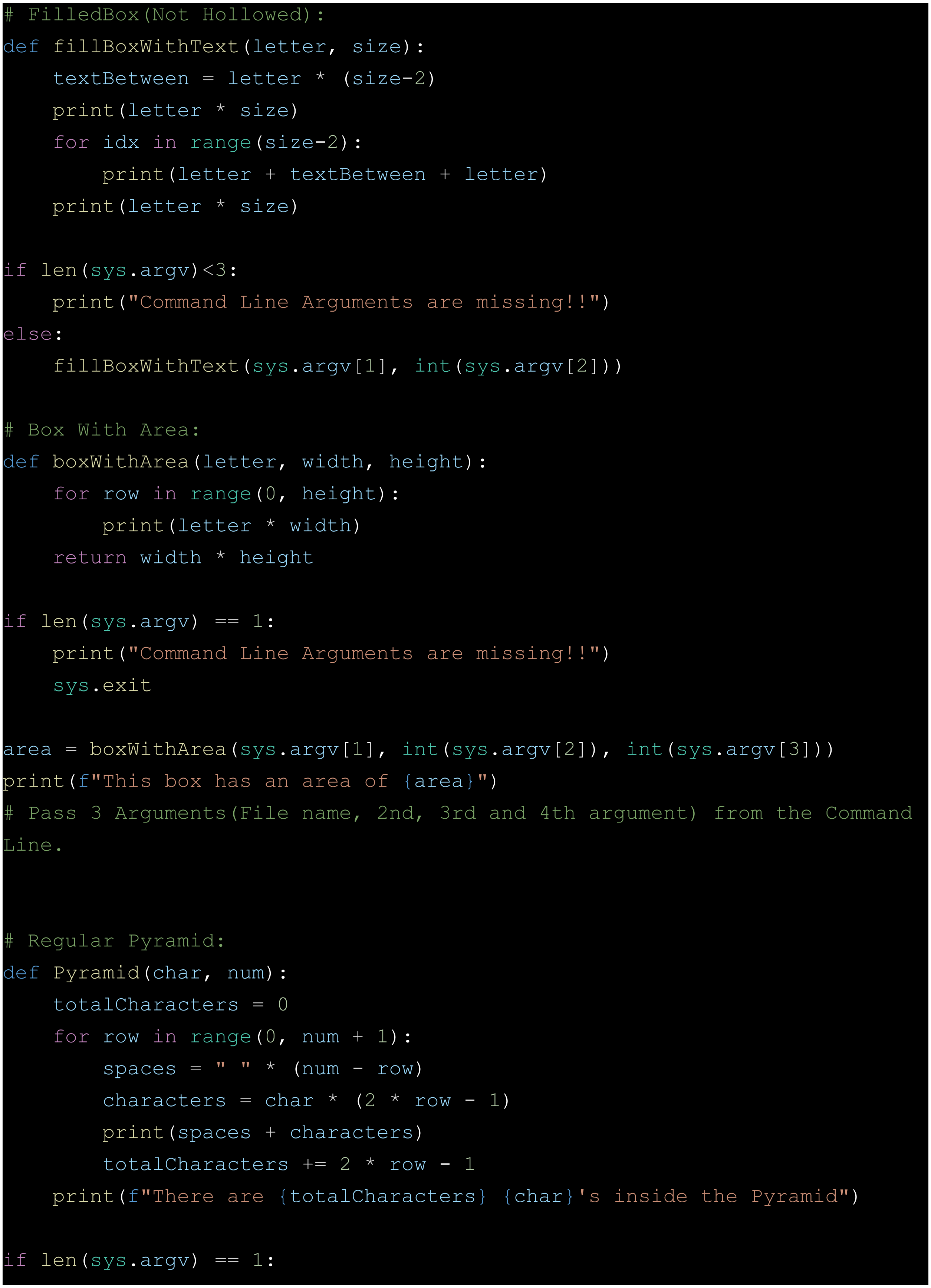
```
    img.show()

# Bottom left:
def negateBottomLefttPixels(img):
    pixelArray = img.load()
    for x in range(img.width//2, 0, -1):
        for y in range(img.height//2, img.height):
            r, g, b = pixelArray[x, y]
            pixelArray[x, y] = (255-r, 255-g, 255-b)
    img.show()

# Bottom right:
def negateBottomRightPixels(img):
    pixelArray = img.load()
    for x in range(img.width//2, img.width):
        for y in range(img.height//2, img.height):
            r, g, b = pixelArray[x, y]
            pixelArray[x, y] = (255-r, 255-g, 255-b)
    img.show()

pic = Image.open("SeaDragon.jpg")
# swapRedToBlue(pic)
# increaseValueToInt30(pic)
# decreaseValueToInt40(pic)
# increaseValueTo20Percent(pic)
# decreaseValueTo20Percent(pic)
# grayScale(pic)
# negatePixels(pic)
# pinkifyWhitePixels(pic)
# negateTopLeftPixels(pic)
# negateTopRightPixels(pic)
# negateBottomLefttPixels(pic)
# negateBottomRightPixels(pic)

# Eight Colors
def eightColorSimplify(img):
    pixelArray = img.load()
    for x in range(img.width):
        for y in range(img.height):
            r, g, b = pixelArray[x, y]
            if r < 100:
                r = 0
            else:
                r = 255

            if g < 100:
                g = 0
            else:
                g = 255

            if b < 100:
                b = 0
            else:
                b = 255

            pixelArray[x, y] = (r, g, b)
    img.show()

photo = Image.open("Charles_Babbage_.jpg")
eightColorSimplify(photo)

# Draw a crowd of faces.
def drawFace(img, x, y):
    img.ellipse([x, y, 100 + x, 100 + y], fill = "yellow")
    img.ellipse([30 + x, 20 + y, 40 + x, 30 + y], fill = "black") # Right eye
    img.ellipse([60 + x, 20 + y, 70 + x, 30 + y], fill = "black") # Left eye
    img.ellipse([30 + x, 65 + y, 70 + x, 80 + y], fill = "red") # Mouth

def drawFaces():
    canvas = Image.new("RGB", (800, 800), "black")
    img = ImageDraw.Draw(canvas)
    for x in range(0, 800, 100):
        for y in range(0, 800, 100):
            drawFace(img, x, y)
    canvas.show()

canvas = Image.new("RGB", (800, 800), "white")
```

```
img = ImageDraw.Draw(canvas)
# drawFace(img, 350, 350)
# drawFaces()


# Copy Image pixels to a Canvas
def cropAndPaste(photo):
    # Get the coordinates using (pixspy.com)
    x0, y0 = 177, 50
    x1, y1 = 277, 100

    croppedPixels = photo.crop((x0, y0, x1, y1))
    canvas = Image.new("RGB", (800, 800), "white")
    canvas.paste(croppedPixels, (350, 350))
    canvas.show()

img = Image.open("Charlesbabbage.jpg")
cropAndPaste(img)

# Copy Image to Image
def replaceBaby():
    img = Image.open("babyGreenScreen.png")
    pixelList = img.getdata()
    bgImage = Image.open("GreenTreeBG.jpg")
    bgList = bgImage.getdata()
    modifiedPixelList = []

    pos = 0
    for pixel in pixelList:
        if (132, 238, 39) <= pixel < (147, 255, 40):
            newPixel = bgList[pos]
        else:
            newPixel = pixel
        modifiedPixelList.append(newPixel)
        pos = pos + 1
    img.putdata(modifiedPixelList)
    img.show()

replaceBaby()
```

```
# Targeting Pixels by Zones
def blackToRedByZone(img):
    pixelArray = img.load()
    x0, y0 = 177, 50
    x1, y1 = 277, 100
    for x in range(0, img.width):
        for y in range(0, img.height):
            if x0 <= x < x1 and y0 <= y <= y1:
                r, g, b = pixelArray[x, y]
                if (0, 0, 0) <= pixelArray[x, y] < (45, 45, 45):
                    r = 255
                    pixelArray[x, y] = (r, g, b)
    img.show()

pix = Image.open("Charlesbabbage.jpg")
# blackToRedByZone(pix)
```

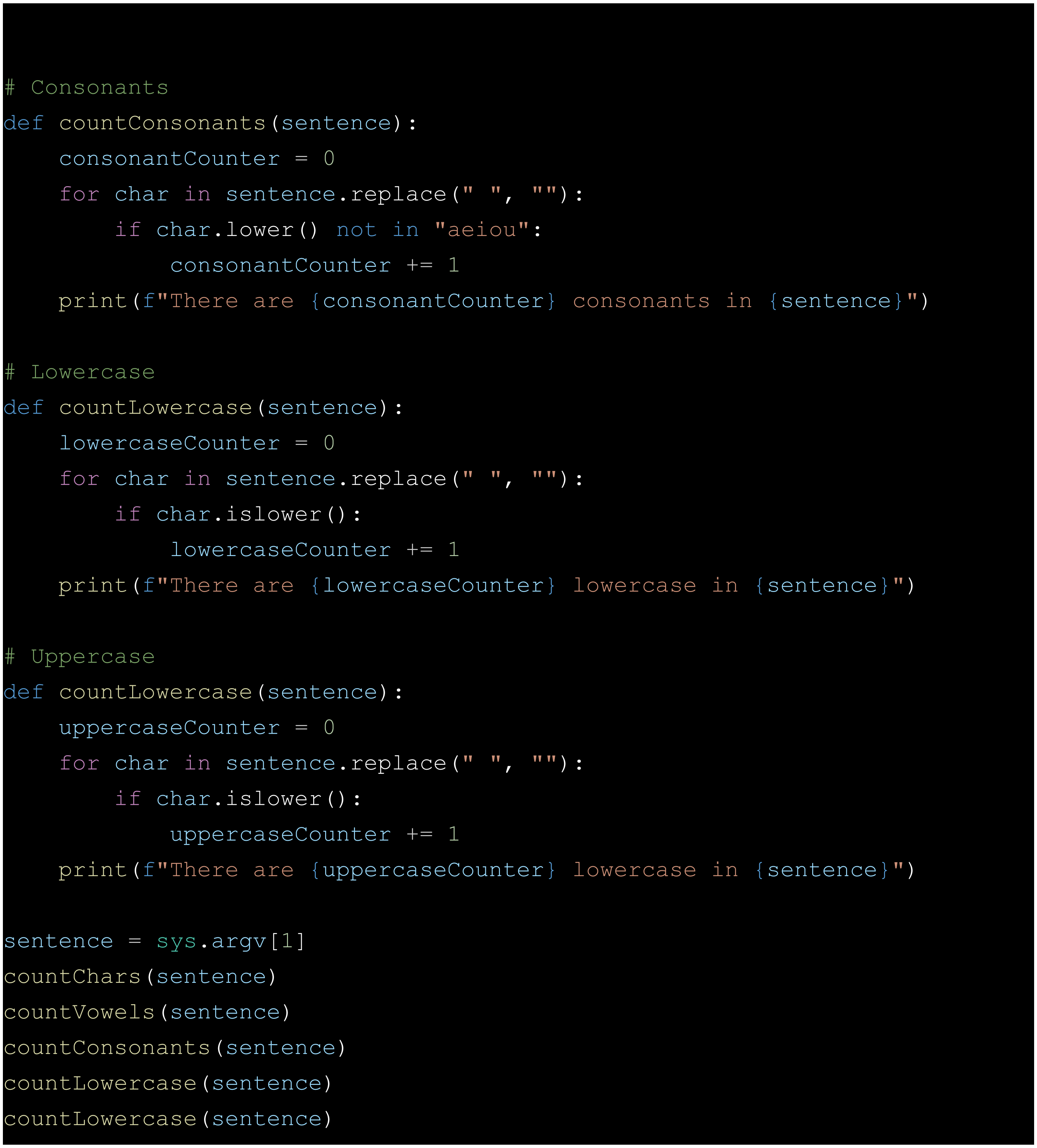
String Codes:

```
import sys


# Command Line Inputs
# Hollowed Box:
def textHollowedBox(text, size):
    spacing = " " * (size-2)
    print("Hollowed Box:")
    print(text * size)
    for count in range(size-2):
        print(text + spacing + text)
    print(text * size)

if len(sys.argv)<3:
    print("You need to add a letter and integer after the program name to complete the box.")
else:
    textHollowedBox(sys.argv[1], int(sys.argv[2]))
```



```python
# FilledBox(Not Hollowed):
def fillBoxWithText(letter, size):
    textBetween = letter * (size-2)
    print(letter * size)
    for idx in range(size-2):
        print(letter + textBetween + letter)
    print(letter * size)

if len(sys.argv)<3:
    print("Command Line Arguments are missing!!")
else:
    fillBoxWithText(sys.argv[1], int(sys.argv[2]))

# Box With Area:
def boxWithArea(letter, width, height):
    for row in range(0, height):
        print(letter * width)
    return width * height

if len(sys.argv) == 1:
    print("Command Line Arguments are missing!!")
    sys.exit

area = boxWithArea(sys.argv[1], int(sys.argv[2]), int(sys.argv[3]))
print(f"This box has an area of {area}")
# Pass 3 Arguments(File name, 2nd, 3rd and 4th argument) from the Command Line.


# Regular Pyramid:
def Pyramid(char, num):
    totalCharacters = 0
    for row in range(0, num + 1):
        spaces = " " * (num - row)
        characters = char * (2 * row - 1)
        print(spaces + characters)
        totalCharacters += 2 * row - 1
    print(f"There are {totalCharacters} {char}'s inside the Pyramid")

if len(sys.argv) == 1:
    print("Command Line Arguments are missing!!")
    sys.exit()

Pyramid(sys.argv[1], int(sys.argv[2]))
# Pass 2 Arguments from the Command Line.


# Inverted Pyramid:
def invertedPyramid(char, num):
    totalCharacters = 0
    for row in range(num, 0, -1):
        spaces = " " * (num - row)
        characters = char * (2 * row - 1)
        print(spaces + characters)
        totalCharacters += 2 * row - 1
    print(f"There are {totalCharacters} {char}'s inside the Pyramid")

if len(sys.argv) == 1:
    print("Command Line Arguments are missing!!")
    sys.exit

invertedPyramid(sys.argv[1], int(sys.argv[2]))
# Pass 2 Arguments from the Command Line.

# Find and Remove Occurances
import sys

def findOccurence(word, match):
    cnt = 0
    for char in word:
        if char.lower() in match.lower():
            cnt = cnt + 1
    print(f"There's {cnt} occurence of {match} in '{word}'")

if len(sys.argv) == 1:
    print("Command Line Arguments are missing!!")
    sys.exit

# Find how many "e" are in a word
findOccurence(sys.argv[1], sys.argv[2])
```

```
# (Home Work Material) Removing Occurences.
def removeOccurence(word, match):
    print(f" Letter '{match}' has been removed from the word
'{word.lower().replace(match.lower(), "")}'")

if len(sys.argv) == 1:
    print("Command Line Arguments are missing!!")
    sys.exit

removeOccurence(sys.argv[1], sys.argv[2])
# Pass 2 Arguments from the Command Line.

#
--------------------------------------------------------------------------

# String Manipulation
# Counting the type of characters in a sentence.
    # char, vowels, consonants, lowercase, uppercase.

# Characters
def countChars(sentence):
    charCounter = 0
    # Replace Space with "" to count no Spaces
    # for char in sentence.replace(" ", ""):
    # Count everything even with Spaces
    for char in sentence:
        charCounter += 1
    print(f"There are {charCounter} characters in {sentence}")

# Vowels
def countVowels(sentence):
    vowelCounter = 0
    for char in sentence:
        if char.lower() in "aeiou":
            vowelCounter += 1
    print(f"There are {vowelCounter} vowels in {sentence}")
```



```python
# Consonants
def countConsonants(sentence):
    consonantCounter = 0
    for char in sentence.replace(" ", ""):
        if char.lower() not in "aeiou":
            consonantCounter += 1
    print(f"There are {consonantCounter} consonants in {sentence}")

# Lowercase
def countLowercase(sentence):
    lowercaseCounter = 0
    for char in sentence.replace(" ", ""):
        if char.islower():
            lowercaseCounter += 1
    print(f"There are {lowercaseCounter} lowercase in {sentence}")

# Uppercase
def countLowercase(sentence):
    uppercaseCounter = 0
    for char in sentence.replace(" ", ""):
        if char.islower():
            uppercaseCounter += 1
    print(f"There are {uppercaseCounter} lowercase in {sentence}")

sentence = sys.argv[1]
countChars(sentence)
countVowels(sentence)
countConsonants(sentence)
countLowercase(sentence)
countLowercase(sentence)
```