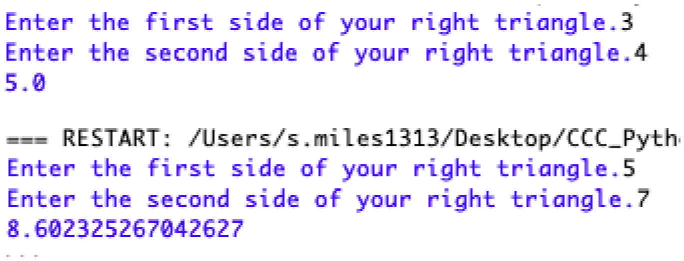
Exercise 7.10.6

```
# Exercise_7.10.6

import math


def findHypot(a,b):

    side_a = int(a)

    side_b = int(b)

    hypot_squared = (side_a**2) + (side_b**2)

    hypotenuse = math.sqrt(hypot_squared)

    return hypotenuse




side1 = input("Enter the first side of your right triangle.")

side2 = input("Enter the second side of your right triangle.")

print (findHypot(side1, side2))
```

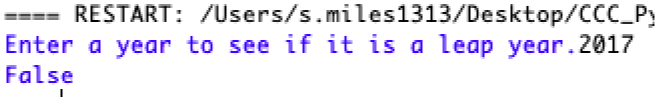


```
Enter the first side of your right triangle.3
Enter the second side of your right triangle.4
5.0

=== RESTART: /Users/s.miles1313/Desktop/CCC_Pyth
Enter the first side of your right triangle.5
Enter the second side of your right triangle.7
8.602325267042627
```

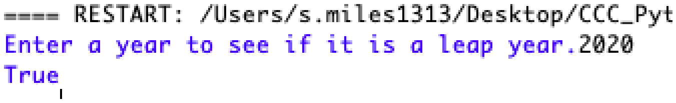
Exercise 7.10.12

```
# Exercise_7.10.12


def isLeap(year):

    year = int(year)


    if year % 4 == 0:
```

```
    if year % 100 != 0:
      year = True
    else:
      if year % 400 == 0:
        year = True
      else:
        year = False
  else:
    year = False

  return year

the_year = input("Enter a year to see if it is a leap year.")
print (isLeap(the_year))
```



```
==== RESTART: /Users/s.miles1313/Desktop/CCC_Py
Enter a year to see if it is a leap year.2017
False
```



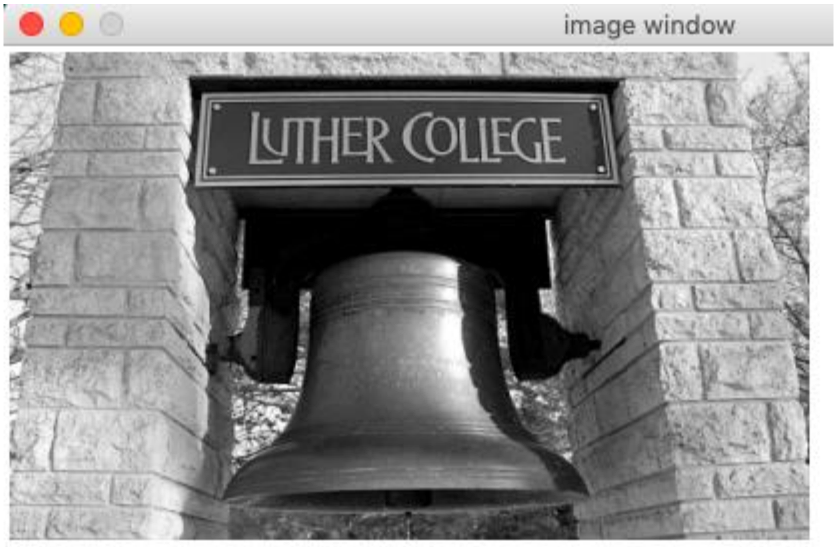
```
==== RESTART: /Users/s.miles1313/Desktop/CCC_Pyt
Enter a year to see if it is a leap year.2020
True
```

Exercise 8.14.8

```
# Exercise_8.14.8

import image

def grayscale(the_image):
  grayscale_image = image.EmptyImage(the_image.getWidth(), the_image.getHeight())
```

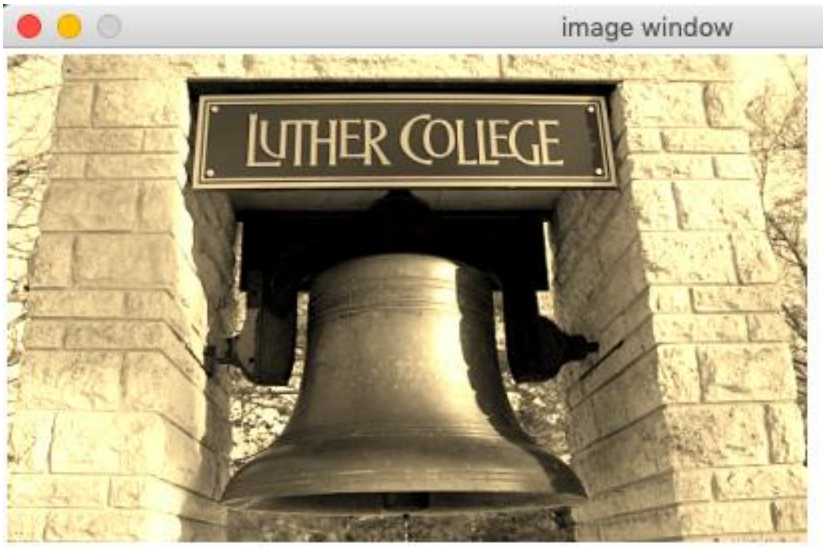
```
    for col in range(the_image.getWidth()):
        for row in range(the_image.getHeight()):
            pixel = the_image.getPixel(col, row)

            red = pixel.getRed()
            green = pixel.getGreen()
            blue = pixel.getBlue()

            average = (red + green + blue)/ 3
            if average - int(average) < 0.5:
                average = int(average)
            else:
                average = int(average) + 1

            grayPixel = image.Pixel(average, average, average)
            grayscale_image.setPixel(col, row, grayPixel)
    return grayscale_image


win = image.ImageWin()
img = image.Image("luther.jpg")

new_img = grayscale(img)
new_img.draw(win)

win.exitonclick()
```

Sophia Miles



Exercise 8.14.10

```
# Exercise_8.14.10


import image


def sepia_tone(the_image):
    sepia_image = image.EmptyImage(the_image.getWidth(), the_image.getHeight())


    for col in range(the_image.getWidth()):
        for row in range(the_image.getHeight()):
            pixel = the_image.getPixel(col, row)


            R = pixel.getRed()
            G = pixel.getGreen()
            B = pixel.getBlue()


            newR = int(R * 0.393 + G * 0.769 + B * 0.189)
            newG = int(R * 0.349 + G * 0.686 + B * 0.168)
            newB = int(R * 0.272 + G * 0.534 + B * 0.131)

            if newR > 255:
                newR = 255
            if newG > 255:
                newG = 255
            if newB > 255:
                newB = 255

            sepiaPixel = image.Pixel(newR, newG, newB)
            sepia_image.setPixel(col, row, sepiaPixel)
    return sepia_image

win = image.ImageWin()
img = image.Image("luther.jpg")

new_img = sepia_tone(img)
new_img.draw(win)

win.exitonclick()
```



Sophia Miles