```python
# Title: Programming Assignment 2
# Due date: Wednesday, September 9, 2021 at 11:59pm
# Author: Sotheanith Sok
# Description:
#  1. Segment the given rocks in “colorful rocks 2.jpg” image
#  2. Plot the result and then save the resulting image as png.
#  3. Count the total number of the gray rocks in the image and print the result.
#  4. Calculate the area of each gray rock and save the result in a file. Explain how you did that.
#  5. Estimate the center of each gray rock and plot the image with red stars on the calculated centers. Explain how you found the centers.
#  6. Upload a pdf file of your code, your answers to question 4 and 5, and the resulting images.

# ---------------------------------------------------------------------------
# Imports
from skimage import io
import matplotlib.pyplot as plt
import numpy as np
from im2bw import im2bw
from bwareaopen import bwareaopen
from bwlabeln import bwlabeln


# 1. Segment the given rocks in “colorful rocks 2.jpg” image
# Load the image and normalize it between 0 and 1
image = io.imread("./colorful rocks 2.jpg")
image = image / 255.0

# Convert image to binary image
image = im2bw(image, 0.72)

# Inverse 0 and 1 with each other for bwareopen and bwlabeln functions
image = np.subtract(1, image)

# Remove all connected components that has less than 800 pixels
image = bwareaopen(image, 800)

# Label connect components
image = bwlabeln(image)


# 2. Plot the result and then save the resulting image as png.
# Setting pyplot settings
fig = plt.figure()
fig.suptitle("Segmented Color Rock 2")
plt.xlabel("Columns")
plt.ylabel("Rows")

# Plot image
plt.imshow(image, cmap="gray")

# Save image to file
print('2. Segmented image has been saved to "segmented_colorful_rock_2.png"')
plt.savefig("segmented_colorful_rock_2.png")

# Show figure
# plt.show()


# 3. Count the total number of the gray rocks in the image and print the result.
nums_gray_rock = image.max()
print("3. Number of gray rocks is %d" % nums_gray_rock)


# 4. Calculate the area of each gray rock and save the result in a file. Explain how you did that
# We can calculate the area of each gray rock by couting the number of pixels belong to each gray rocks based on the label that we generate with bwlabeln function
labels, areas = np.unique(image, return_counts=True)
print("4. Calculate the area of each gray rock.")
for i in range(1, len(labels)):
    print("Gray rock %d has area %d pixels" % (labels[i], areas[i]))


# 5. Estimate the center of each gray rock and plot the image with red stars on the calculated centers. Explain how you found the centers
# We can caluclate the center of each gray rock by suming each pixels coordinate seperately (sum of row indexs and sum of coloumn indexs) and divide the result of the number of point.

# Calculate the center for each connected components
labels = np.unique(image)[1:]
centers = []
for label in labels:
    rows, columns = np.where(image == label)
    center_row_index = np.mean(rows)
    center_col_index = np.mean(columns)
    centers.append((center_row_index, center_col_index))

# Plot the center
for center in centers:
    plt.plot(center[1], center[0], "r*")

# Save figure to files
print('5. Result has been saved to "center_plotted_segmented_colorful_rock_2.png"')
plt.savefig("center_plotted_segmented_colorful_rock_2.png")
```