Calculating NDVI with the Python Imaging Library (PIL)

Create a script for calculating the NDVI from two images, a regular RGB image and a near-infrared (NIR) image.

Translate the following pseudo-code into real Python script. What is your conclusion?

PSEUDO-CODE:

```
Import the Image module

open the RGB image 'imageRGB.tif'

open the NIR image 'imageNIR.tif'

split the RGB image into a RED, GREEN and BLUE image

get the number of rows and columns of the images

create a new image for the NDVI.
(it should have mode 'L' and have exactly the same size as the other images)

Loop over all the rows:

        Loop over all the columns:

                get a RED pixel at the current location

                get a NIR pixel at the current location

                if n+r equals zero then:
                        ndvi becomes zero
                else:
                        calculate NDVI (make sure the calculation is done in float): (n-r)/(n+r)
                        scale the NDVI from the range [-1..1] to the range [0..255]

                assign the calculated NDVI value to the NDVI image

save the NDVI image

(show the NDVI image)

create a new image called VEG by thresholding the NDVI at 128
(hint: you can use the 'point' method of the NDVI image)

save the VEG image

(show the VEG image)
```