```python
import cv2
import numpy as np

B = np.array([[0, 1, 0], [1, 1, 1], [0, 1, 0]], dtype=np.uint8)

img = cv2.imread(r"D:\FCI\The 3 Level\The Second Term\New\Digital Image Processing\imags\BonusTask\close.png", cv2.IMREAD_GRAYSCALE)

pad_size = B.shape[0] // 2
img_padded = cv2.copyMakeBorder(img, pad_size, pad_size, pad_size, pad_size, cv2.BORDER_CONSTANT, value=0)

for i in range(15): # Increase this value to increase Closing

    # Dilation step
    img_dilated = np.zeros_like(img_padded)
    for i in range(pad_size, img_padded.shape[0] - pad_size):
        for j in range(pad_size, img_padded.shape[1] - pad_size):
            if np.any(np.logical_and(B, img_padded[i-pad_size:i+pad_size+1, j-pad_size:j+pad_size+1])):
                img_dilated[i, j] = 255

    img_padded = img_dilated[pad_size:-pad_size, pad_size:-pad_size]

    # Erosion step
    img_eroded = np.zeros_like(img_padded)
    for i in range(pad_size, img_padded.shape[0] - pad_size):
        for j in range(pad_size, img_padded.shape[1] - pad_size):
            if np.all(np.logical_or(np.logical_not(B), img_padded[i-pad_size:i+pad_size+1, j-pad_size:j+pad_size+1])):
                img_eroded[i, j] = 255

    img_padded = img_eroded[pad_size:-pad_size, pad_size:-pad_size]

img_closed = img_eroded[pad_size:-pad_size, pad_size:-pad_size]

# Display the images
cv2.imshow('Input Image', img)
cv2.imshow('Closed Image', img_closed)
cv2.waitKey(0)
cv2.destroyAllWindows()
```