1. 아래의 코드에서 빈칸을 채워서 거리구하기 코드를 작성해 보세요

a_거리구하기.py

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# 1. 빈 이미지 생성
img = np.zeros((____, ____), dtype="uint8")

# 2. 원 그리기
cv2.circle(img, (____, ____), ____, (____, ____, ____), (-1))
cv2.circle(img, (____, ____), ____, (____, ____, ____), (-1))

# 3. 거리 변환 수행
dist_transform = cv2.distanceTransform(img, cv2.____, ____)

# 4. 결과 시각화
plt.imshow(____)
plt.show()
```

[키워드]

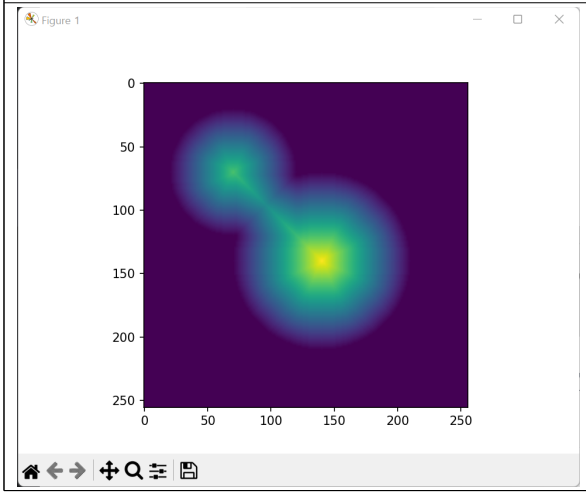
256: 이미지의 크기,,,70, 70: 첫 번째 원의 중심 좌표,,,50: 첫 번째 원의 반지름

255, 255, 255: 원의 색깔 (흰색), 140, 140: 두 번째 원의 중심 좌표

70: 두 번째 원의 반지름, DIST_L2: 거리 변환의 유형 (L2 유클리디안 거리)

3: 거리 변환에 사용되는 마스크 크기 ,

dist_transform: 거리 변환 결과를 저장할 변수 이름



2. Python과 OpenCV를 사용하여 동전의 개수를 찾는 코드가 있습니다. 아래의 코드에서 빈 칸을 채워서 동전의 개수를 정확히 찾아보세요. b_동전개수 확인 .py

```
# 필요한 라이브러리를 불러옵니다.
import cv2
import numpy as np
from matplotlib import pyplot as plt

# 이미지를 로드합니다.
img = cv2.imread('coin.jpg', 0)

# OTSU 이진화를 수행합니다. (빈 칸 1)
ret, bin_img = cv2.threshold(img, 0, 255, cv2._____ + cv2.THRESH_OTSU)

# 작은 노이즈를 제거합니다. (빈 칸 2)
kernel = np.ones((3,3), np.uint8)
opening = cv2._____(bin_img, cv2.MORPH_OPEN, kernel, iterations=2)

# 배경을 추출합니다. (빈 칸 3)
sure_bg = cv2._____(opening, kernel, iterations=2)

# 거리 변환을 수행합니다. (빈 칸 4)
dist_transform = cv2._____(opening, cv2.DIST_L2, 5)

# 거리 변환 결과를 사용하여 전경을 추출합니다. (빈 칸 5)
ret, sure_fg = cv2._____(dist_transform, 0.5 * dist_transform.max(), 255, 0)

# 전경과 배경이 아닌 것을 추출합니다.
sure_fg = np.uint8(sure_fg)
res = cv2.subtract(sure_bg, sure_fg)

# 영역 라벨링을 수행하고 라벨의 개수를 구합니다. (빈 칸 6)
ret, markers = __________

# 실제 동전의 개수를 출력합니다. (빈 칸 7)
__________(f"동전의 개수는 _______ 입니다.")
```

[키워드]

THRESH_BINARY_INV: 이진화의 유형을 결정
morphologyEx: 모폴로지 연산을 수행하는 함수
dilate: 팽창 연산을 수행하는 함수
distanceTransform: 거리 변환을 수행하는 함수
threshold: 임계값 처리를 하는 함수
cv2.connectedComponents: 영역 라벨링을 수행하는 함수
print: 출력을 위한 함수

[출력결과]
동전의 개수는 24 입니다.

3. 이미지 처리를 이용해 동전을 세는 문제를 해결하려고 합니다. 아래 키워드 힌트를 사용해서 코드를 완성 하세요

① 이미지 로딩: 원본 'coin.jpg' 이미지를 로드합니다.
② OTSU 이진화: 원본 이미지를 이진화하여 배경과 객체(여기서는 동전)를 구분합니다.
③ 노이즈 제거: 작은 노이즈를 제거하기 위해 모폴로지 연산을 사용합니다.
④ 배경 추출: 배경을 확실하게 추출합니다.
⑤ 거리 변환: 각 픽셀에서 가장 가까운 영역까지의 거리를 계산합니다.
⑥ 전경 추출: 거리 변환을 통해 얻은 정보를 사용하여 전경(동전)을 확실히 추출합니다.
⑦ 영역 라벨링: 동전 각각을 라벨링합니다.
⑧ Watershed 알고리즘: 라벨링한 정보를 이용해 최종적으로 동전을 구분합니다.

[키워드 힌트]c_동전Segmentation.py

imread, THRESH_BINARY_INV, morphologyEx, dilate,
distanceTransform , threshold, subtract

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Create a 2x5 subplot array
fig, axes = plt.subplots(2, 5, figsize=(20, 8))
axes = axes.ravel()

# 1. 이미지 로딩
img = cv2._____('coin.jpg', 0)  # 빈 칸 1
axes[0].imshow(img, cmap='gray')
axes[0].set_title('1. Original Image')

# 2. OTSU 이진화
ret, bin_img = cv2.threshold(img, 0, 255, cv2._____ + cv2.THRESH_OTSU)  # 빈 칸 2
axes[1].imshow(bin_img, cmap='gray')
axes[1].set_title('2. OTSU Binary Image')

# 4. 이미지에 포함된 작은 노이즈 삭제
kernel = np.ones((3, 3), np.uint8)
opening = cv2._____(bin_img, cv2.MORPH_OPEN, kernel, iterations=2)  # 빈 칸 3
axes[2].imshow(opening, cmap='gray')
```

```
axes[2].set_title('4. Noise Removal')

# 5. 배경 추출
sure_bg = cv2._____(opening, kernel, iterations=2)  # 빈 칸 4
axes[3].imshow(sure_bg, cmap='gray')
axes[3].set_title('5. Background Extraction')

# 6. 거리 변환
dist_transform = cv2._____(opening, cv2.DIST_L2, 5)  # 빈 칸 5
axes[4].imshow(dist_transform, cmap='gray')
axes[4].set_title('6. Distance Transform')

# 7. 전경 추출
ret, sure_fg = cv2.____(dist_transform, 0.5 * dist_transform.max(), 255, 0)  # 빈 칸 6
axes[5].imshow(sure_fg, cmap='gray')
axes[5].set_title('7. Foreground Extraction')

# 8. 배경도 전경도 아닌 값 추출
sure_fg = np.uint8(sure_fg)
res = cv2._____(sure_bg, sure_fg)  # 빈 칸 7
axes[6].imshow(res, cmap='gray')
axes[6].set_title('8. Unknown Region')

# 9. 영역 라벨
ret, markers = cv2.connectedComponents(sure_fg)
axes[7].imshow(markers)
axes[7].set_title(f'9. Connected Components (Coins: {ret-1})')
print(f"동전의 개수는 {ret-1}개 입니다.")

# 10. Watershed
markers = markers + 1
markers[res == 255] = 0
img_colored = cv2.cvtColor(img, cv2.COLOR_GRAY2BGR)
markers = cv2.watershed(img_colored, markers)
img_colored[markers == -1] = [255, 0, 0]
axes[8].imshow(cv2.cvtColor(img_colored, cv2.COLOR_BGR2RGB))
axes[8].set_title('10. Watershed')
```

```
for ax in axes:
    ax.axis('off')

plt.show()
```

[실행결과]

동전의 개수는 24 입니다.