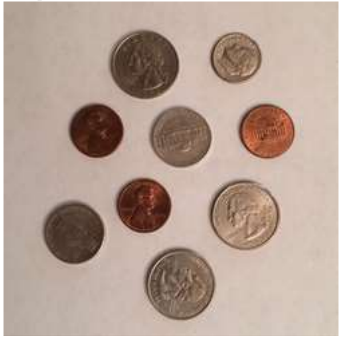
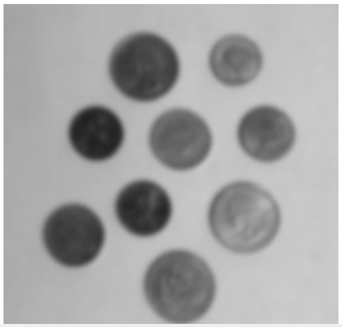
```
# USAGE
# python counting_coins.py --image ../images/coins.png

# Import the necessary packages
from __future__ import print_function
import numpy as np
import imutils
import cv2
from google.colab.patches import cv2_imshow


# Load the image, convert it to grayscale, and blur it slightly
image = cv2.imread("coins.png")
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
blurred = cv2.GaussianBlur(gray, (11, 11), 0)
print("Image")
cv2_imshow(image)
print("blurred")
cv2_imshow(blurred)
```

```
Image
```

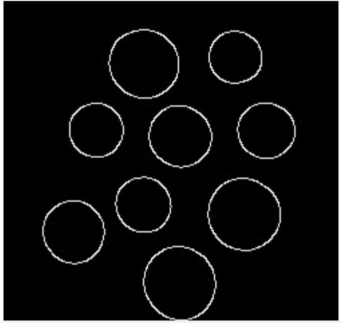


```
blurred
```



```
# The first thing we are going to do is apply edge detection to
# the image to reveal the outlines of the coins
edged = cv2.Canny(blurred, 30, 150)
print("Edges")
cv2_imshow(edged)
```

```
Edges
```

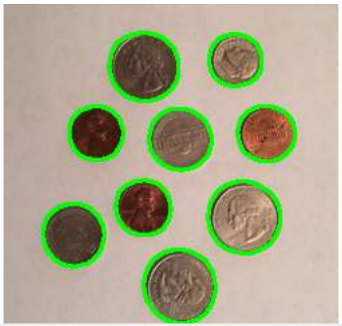


```
# Find contours in the edged image.
# NOTE: The cv2.findContours method is DESTRUCTIVE to the image
# you pass in. If you intend on reusing your edged image, be
# sure to copy it before calling cv2.findContours
cnts = cv2.findContours(edged.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
# cnts = cv2.findContours(edged.copy(), cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
cnts = imutils.grab_contours(cnts)


# How many contours did we find?
print("I count {} coins in this image".format(len(cnts)))
```

```
I count 9 coins in this image
```

```
# Let's highlight the coins in the original image by drawing a
# green circle around them
coins = image.copy()
cv2.drawContours(coins, cnts, -1, (0, 255, 0), 2)
cv2.drawContours(coins, cnts, 0, (0, 255, 0), 2)
cv2.drawContours(coins, cnts, 1, (0, 255, 0), 2)
cv2.drawContours(coins, cnts, 2, (0, 255, 0), 2)
cv2.drawContours(coins, cnts, -1, (0, 255, 0), 2)
print("Coins")
cv2_imshow(coins)
```

```
Coins
```



```
# Now, let's loop over each contour
for (i, c) in enumerate(cnts):
  # We can compute the 'bounding box' for each contour, which is
  # the rectangle that encloses the contour
  (x, y, w, h) = cv2.boundingRect(c)

  # Now that we have the contour, let's extract it using array
  # slices
  print("Coin #{}".format(i + 1))
  coin = image[y:y + h, x:x + w]
cv2_imshow(coin)

  # Just for fun, let's construct a mask for the coin by finding
  # The minumum enclosing circle of the contour
mask = np.zeros(image.shape[:2], dtype = "uint8")
((centerX, centerY), radius) = cv2.minEnclosingCircle(c)
cv2.circle(mask, (int(centerX), int(centerY)), int(radius), 255, -1)
mask = mask[y:y + h, x:x + w]
print("Masked Coin")
cv2_imshow(cv2.bitwise_and(coin, coin, mask = mask))
```

```
Coin #1
Coin #2
Coin #3
Coin #4
Coin #5
Coin #6
Coin #7
Coin #8
Coin #9
```

```
Masked Coin
```