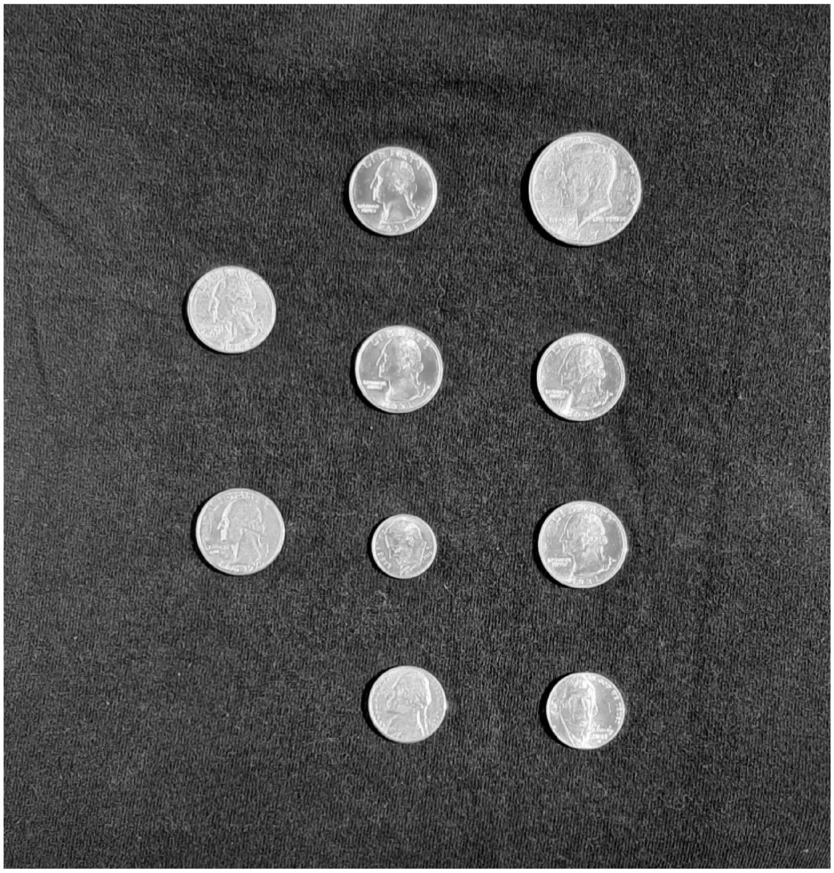
Coin Counting

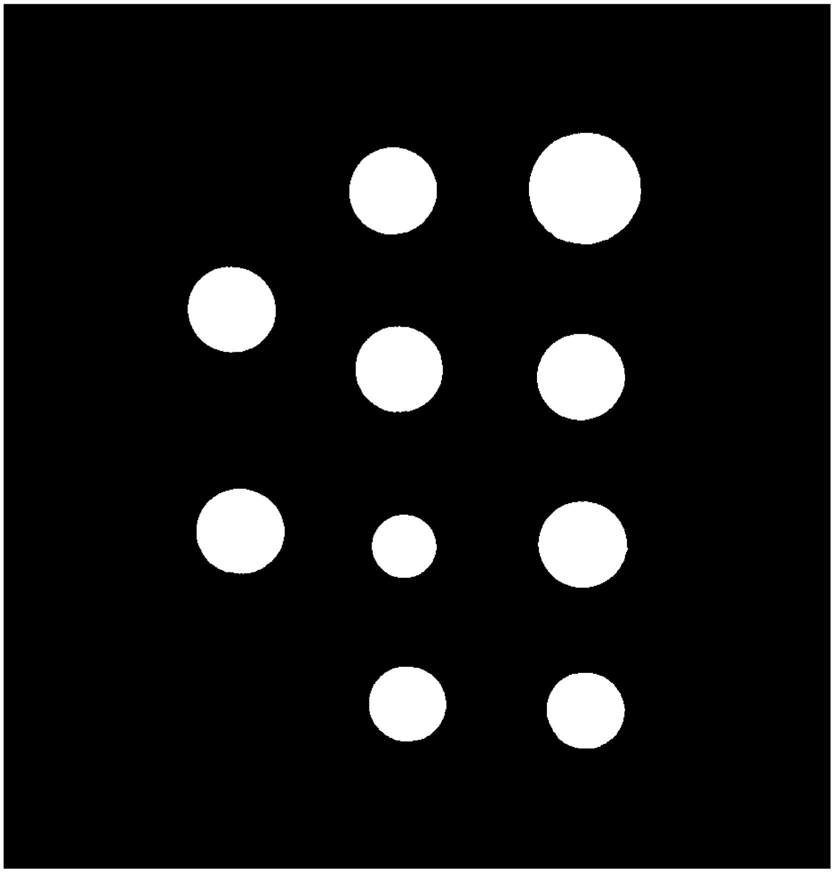
For this problem, your code will need to do the following:

- Provide a mask to accurately segment the coins from the background. Use variable name **`coinMask`**.
- Create the masked image. Use variable name **`maskedCoinImage`**.
- Calculate the area and perimeter of each coin. Provide the results in a table variable **`coinSizes`**. **Hint:** Remember, to get a table output directly from the **`regionprops`** function, use the syntax shown in this documentation example.
- Analyze the mask region properties to determine the *number of each coin type* present. Use variable names **`nDimes`**, **`nNickels`**, **`nQuarters`**, and **`nFiftyCents`**. **Note:** Although you can see the correct numbers here by looking at the image, and could hard code the answers, in your final project you'll need to determine the coin types again for an image you will not know ahead of time.
- Calculate the total $ value of coins present. Use variable name **`USD`**.

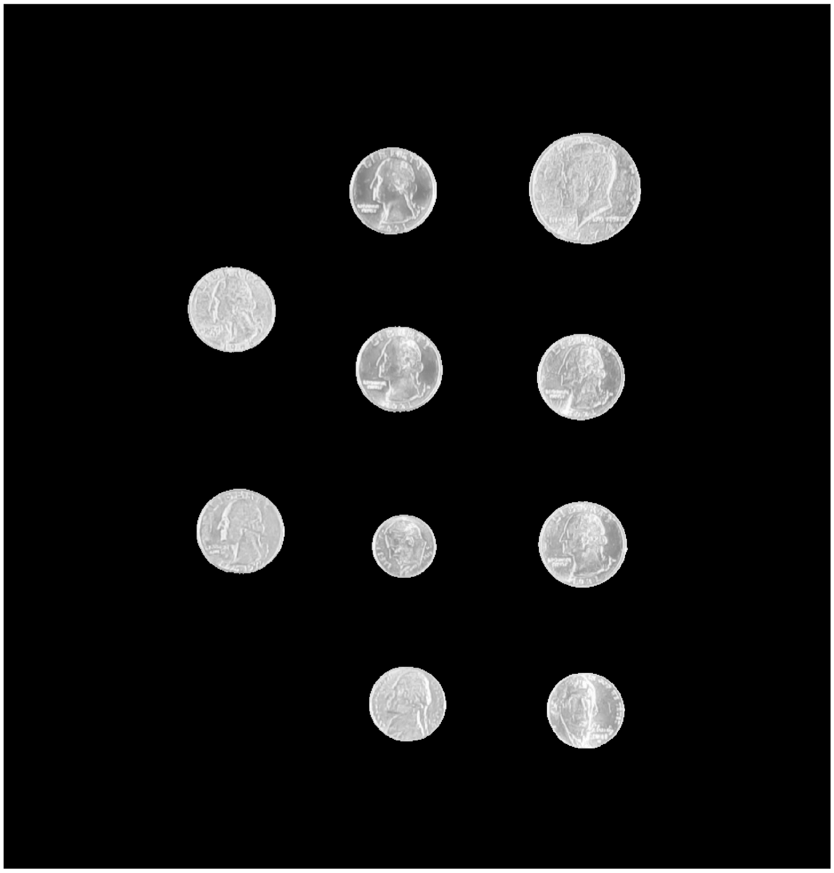
```
coinImage = imread("coinImage.png");
imshow(coinImage);
```



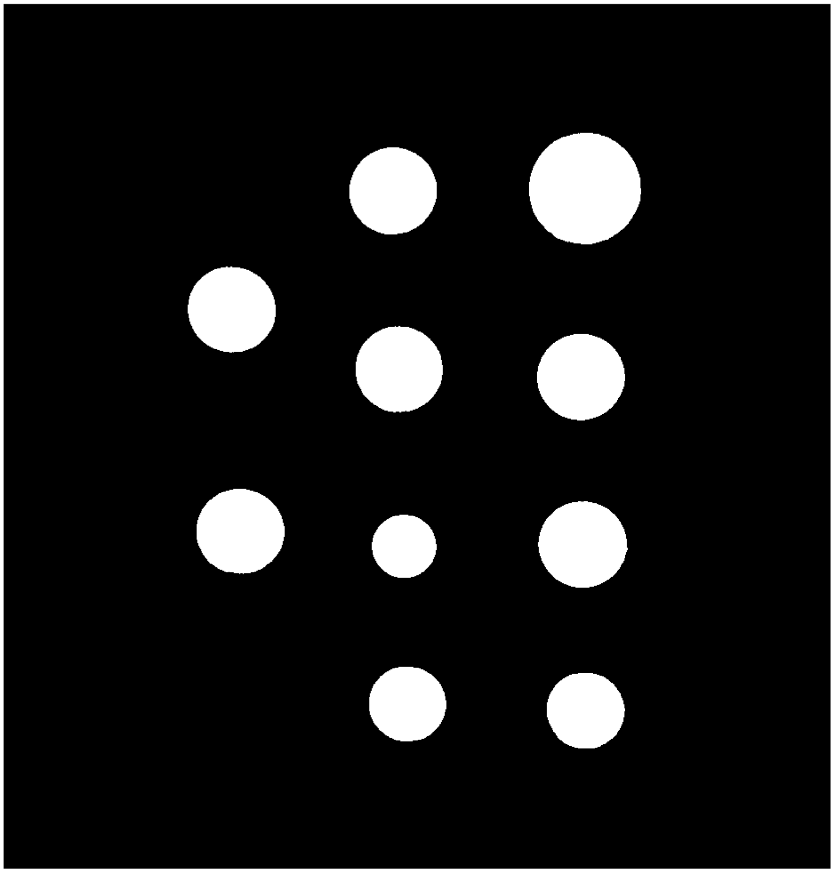
```
[coinMask,maskedCoinImage] = segmentCoin(coinImage);
imshow(coinMask)
```



```
imshow(maskedCoinImage)
```



```
bw = maskedCoinImage > 100;
imshow(bw)
```



```
coinSizes = regionprops("table",bw,"Area","Perimeter");
nDimes = coinSizes.Area < 4100;
nDimes = sum(nDimes);
nNickels = (coinSizes.Area > 4100 & coinSizes.Area <6000);
nNickels = sum(nNickels);
nQuarters = (coinSizes.Area > 6000 & coinSizes.Area < 7800);
nQuarters = sum(nQuarters);
nFiftyCents = coinSizes.Area > 7800;
nFiftyCents = sum(nFiftyCents);
USD = (nDimes * 0.10) + (nNickels * 0.05) + ...
    (nQuarters * 0.25) + (nFiftyCents * 0.50)
```

```
USD = 2.2000
```

```
function [coinMask,maskedCoinImage] = segmentCoin(X)
%segmentImage Segment image using auto-generated code from Image Segmenter app
%  [BW,MASKEDIMAGE] = segmentImage(X) segments image X using auto-generated
%  code from the Image Segmenter app. The final segmentation is returned in
%  BW, and a masked image is returned in MASKEDIMAGE.
% Auto-generated by imageSegmenter app on 31-Dec-2022
%----------------------------------------------------


% Threshold image - manual threshold
coinMask = im2gray(X) > 188;

% Close mask with default
radius = 10;
decomposition = 0;
se = strel('disk', radius, decomposition);
coinMask = imclose(coinMask, se);

% Create masked image.
maskedCoinImage = X;
maskedCoinImage(~coinMask) = 0;
end
```