In [11]:
```python
# GENERATOR + CHALLENGE #1
import numpy as np
import cv2

# LOAD CORRECT TAG DICTIONARY
arucoDict = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_APRILTAG_16H5)
SIZE = 500 # pixels

# CREATE ARRAY FOR MARKER
marker = np.zeros((SIZE, SIZE, 1), dtype=np.uint8)

# DRAW AND SAVE MARKER
ID = 23
cv2.aruco.generateImageMarker(arucoDict, ID, SIZE, marker, 1)
cv2.imwrite('DICT_APRILTAG_16H5_id_{}_{}.png'.format(ID, SIZE), marker)
```

Out[11]: True

In [26]:
```python
# DETECTOR + CHALLENGE #2
import cv2
import math

# LOAD IMAGE AND TAG DICTIONARY
tags = cv2.imread('data/two_tags_APRILTAG_16H5.png')
dictionary = cv2.aruco.getPredefinedDictionary(cv2.aruco.DICT_APRILTAG_16H5)
parameters = cv2.aruco.DetectorParameters()
detector = cv2.aruco.ArucoDetector(dictionary, parameters)

# DETECT TAGS IN IMAGE
markerCorners, markerIds, rejectedCandidates = detector.detectMarkers(cv2.cvtColor(
print(markerCorners)

# DRAW DETECTION AND SAVE FILE
cv2.aruco.drawDetectedMarkers(tags, markerCorners, markerIds, borderColor=(255, 0,
cv2.imwrite('detection_two_tags_APRILTAG_16H5.png', tags)

# CALCULATE DISTANCE BETWEEN CLOSEST POINTS (Based on markerCorners)
x1 = 428
y1 = 471
x2 = 584
y2 = 500
scale = (770 - 471)/3 # width of image/3cm
dist = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
distInCM = dist/scale
print("\nDistance in centimeters: " + str(distInCM) + "cm")
```

```
(array([[[129., 471.],
        [428., 471.],
        [428., 770.],
        [129., 770.]]], dtype=float32), array([[[584., 201.],
        [883., 201.],
        [883., 500.],
        [584., 500.]]], dtype=float32))

Distance in centimeters: 1.5920329659127568cm
```