In [1]:

```
import matplotlib.pyplot as plt
import os
import numpy as np
from skimage.io import imread
from sklearn.model_selection import train_test_split
from skimage.transform import resize
import seaborn as sns
```

In [2]:

```
target = []
images = []
flat_data = []
PATH = 'C:\\Users\\YakupAkdin\\AAAaVeriMadenciligi\\data'
#'
cate = ['arp','kanun','elektroGitar','baglama','gitar','keman','kemence','mandolin','ud','yayliTambur']
import io
#Görüntü Formatı ve diziye atama
for category in cate:
    class_num = cate.index(category)
    path = os.path.join(PATH,category)
    for img in os.listdir(path):
        print(img)

        img_array = imread(os.path.join(path,img),'*.jpg')
        img_array= resize(img_array, (64,64,3))
        images.append(img_array)
        img_array = img_array.flatten()
        flat_data.append(img_array)
        target.append(class_num)

flat_data = np.array(flat_data)
target = np.array(target)
images = np.array(images)
```

```
yayliTambur(81).jpg
yayliTambur(82).jpg
yayliTambur(83).jpg
yayliTambur(84).jpg
yayliTambur(85).jpg
yayliTambur(86).jpg
yayliTambur(87).jpg
yayliTambur(88).jpg
yayliTambur(89).jpg
yayliTambur(9).jpg
yayliTambur(90).jpg
yayliTambur(91).jpg
yayliTambur(92).jpg
yayliTambur(93).jpg
yayliTambur(94).jpg
yayliTambur(95).jpg
yayliTambur(96).jpg
yayliTambur(97).jpg
yayliTambur(98).jpg
yayliTambur(99).jpg
```

In [3]:

```
# Train-Test
x_train, x_test, y_train, y_test = train_test_split(flat_data, target, test_size=0.33, random_state=1)
```

In [4]:

```
from sklearn.tree import DecisionTreeClassifier
x_train.shape
clf = DecisionTreeClassifier(criterion="entropy")
clf.fit(x_train, y_train)
```

Out[4]:

```
DecisionTreeClassifier(criterion='entropy')
```

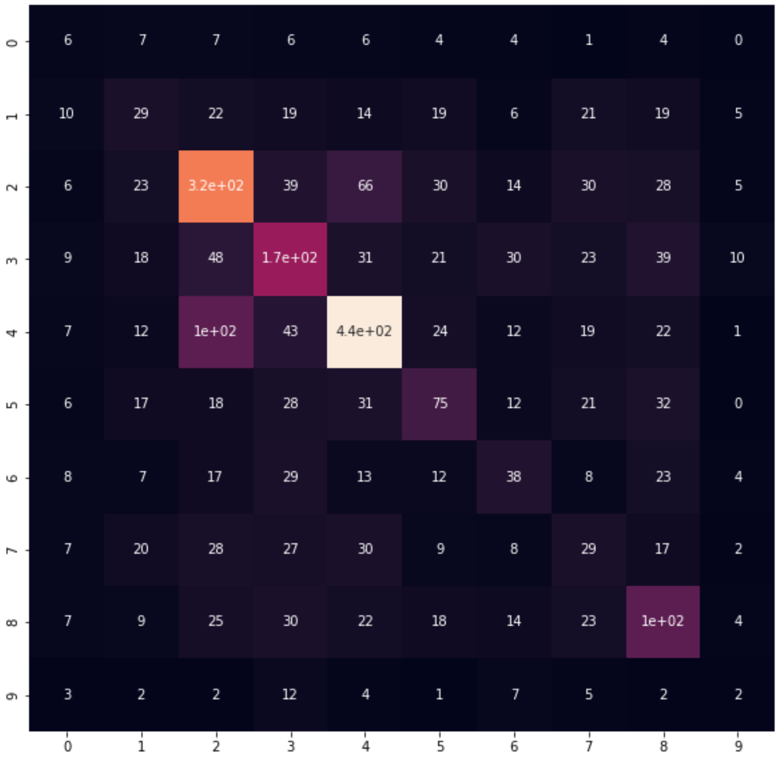
In [5]:

```
from sklearn import metrics
from sklearn.metrics import confusion_matrix, classification_report

predictions = clf.predict(x_test)
```

In [7]:

```
#Grafik
cm = confusion_matrix(y_test, predictions)
fig, ax = plt.subplots(figsize=(10, 10))
sns.heatmap(cm, annot=True, xticklabels=sorted(set(y_test)), yticklabels=sorted(set(y_test)), cbar=False, ax=ax)

print(classification_report(y_test, predictions, target_names=cate))
print("Precision = %.3f" % metrics.precision_score(y_test, predictions, average='weighted'))
```

```
              precision    recall  f1-score   support

         arp       0.09      0.13      0.11        45
       kanun       0.20      0.18      0.19       164
 elektroGitar       0.54      0.57      0.55       557
     baglama       0.42      0.42      0.42       397
       gitar       0.67      0.64      0.65       679
       keman       0.35      0.31      0.33       240
     kemence       0.26      0.24      0.25       159
    mandolin       0.16      0.16      0.16       177
          ud       0.36      0.41      0.38       256
  yayliTambur       0.06      0.05      0.05        40

    accuracy                           0.44      2714
   macro avg       0.31      0.31      0.31      2714
weighted avg       0.44      0.44      0.44      2714

Precision = 0.444
```



In [ ]: