In [12]:
```
import pandas as pd
import seaborn as sns
from sklearn.datasets import load_digits
import matplotlib.pyplot as plt
```

In [13]:
```
ram=load_digits()
```

In [14]:
```
dir(ram)
```

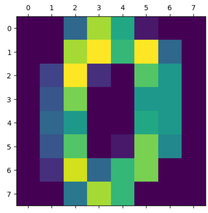
Out[14]:
```
['DESCR', 'data', 'feature_names', 'frame', 'images', 'target', 'target_names']
```

In [15]:
```
ram.data
```

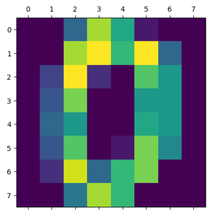
Out[15]:
```
array([[ 0.,  0.,  5.,  ...,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  ..., 10.,  0.,  0.],
       [ 0.,  0.,  0.,  ..., 16.,  9.,  0.],
       ...,
       [ 0.,  0.,  1.,  ...,  6.,  0.,  0.],
       [ 0.,  0.,  2.,  ..., 12.,  0.,  0.],
       [ 0.,  0., 10.,  ..., 12.,  1.,  0.]])
```

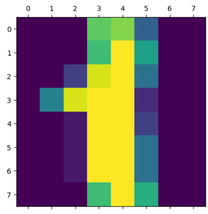
In [16]:
```
plt.matshow(ram.images[0])
```

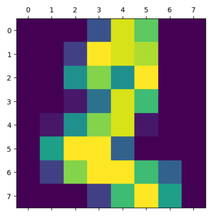
Out[16]:
```
<matplotlib.image.AxesImage at 0x1c774704390>
```

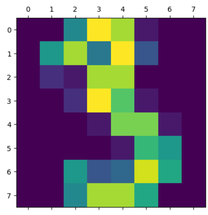


In [17]:
```
for i in range(4):
    plt.matshow(ram.images[i])
```









In [18]:
```
df
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[18], line 1
----> 1 df

NameError: name 'df' is not defined
```

In [19]:
```
df=pd.DataFrame(ram.data) #here we create the dataframe using data
```

In [20]:
```
df
```

Out[20]:

	0	1	2	3	4	5	6	7	8	9	…	54	55	56	57	58	59	60	61	62	63
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	0.0	6.0	13.0	10.0	0.0	0.0	0.0
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	11.0	16.0	10.0	0.0	0.0
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	…	5.0	0.0	0.0	0.0	0.0	3.0	11.0	16.0	9.0	0.0
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	0.0	0.0	8.0	…	9.0	0.0	0.0	0.0	7.0	13.0	13.0	9.0	0.0	0.0
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	0.0	0.0	2.0	16.0	4.0	0.0	0.0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
1792	0.0	0.0	4.0	10.0	13.0	6.0	0.0	0.0	0.0	1.0	…	4.0	0.0	0.0	0.0	2.0	14.0	15.0	9.0	0.0	0.0
1793	0.0	0.0	6.0	16.0	13.0	11.0	1.0	0.0	0.0	0.0	…	1.0	0.0	0.0	0.0	6.0	16.0	14.0	6.0	0.0	0.0
1794	0.0	0.0	1.0	11.0	15.0	1.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	0.0	2.0	9.0	13.0	6.0	0.0	0.0
1795	0.0	0.0	2.0	10.0	7.0	0.0	0.0	0.0	0.0	0.0	…	2.0	0.0	0.0	0.0	5.0	12.0	16.0	12.0	0.0	0.0
1796	0.0	0.0	10.0	14.0	8.0	1.0	0.0	0.0	0.0	2.0	…	8.0	0.0	0.0	1.0	8.0	12.0	14.0	12.0	1.0	0.0

1797 rows × 64 columns

In [21]:
```
df['target']=ram.target
```

In [22]:
```
df
```

Out[22]:

	0	1	2	3	4	5	6	7	8	9	…	55	56	57	58	59	60	61	62	63	target
0	0.0	0.0	5.0	13.0	9.0	1.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	6.0	13.0	10.0	0.0	0.0	0.0	0
1	0.0	0.0	0.0	12.0	13.0	5.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	0.0	11.0	16.0	10.0	0.0	0.0	1
2	0.0	0.0	0.0	4.0	15.0	12.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	0.0	3.0	11.0	16.0	9.0	0.0	2
3	0.0	0.0	7.0	15.0	13.0	1.0	0.0	0.0	0.0	8.0	…	0.0	0.0	0.0	7.0	13.0	13.0	9.0	0.0	0.0	3
4	0.0	0.0	0.0	1.0	11.0	0.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	0.0	2.0	16.0	4.0	0.0	0.0	4
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
1792	0.0	0.0	4.0	10.0	13.0	6.0	0.0	0.0	0.0	1.0	…	0.0	0.0	0.0	2.0	14.0	15.0	9.0	0.0	0.0	9
1793	0.0	0.0	6.0	16.0	13.0	11.0	1.0	0.0	0.0	0.0	…	0.0	0.0	0.0	6.0	16.0	14.0	6.0	0.0	0.0	0
1794	0.0	0.0	1.0	11.0	15.0	1.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	2.0	9.0	13.0	6.0	0.0	0.0	8
1795	0.0	0.0	2.0	10.0	7.0	0.0	0.0	0.0	0.0	0.0	…	0.0	0.0	0.0	5.0	12.0	16.0	12.0	0.0	0.0	9
1796	0.0	0.0	10.0	14.0	8.0	1.0	0.0	0.0	0.0	2.0	…	0.0	0.0	1.0	8.0	12.0	14.0	12.0	1.0	0.0	8

1797 rows × 65 columns

In [23]:
```
from sklearn.model_selection import train_test_split
```

In [24]:
```
x=df.drop('target',axis=1)
```

In [25]:
```
y=df.target  #
```

In [26]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2) # here we set the training and test dataset
```

In [27]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [28]:
```
model=RandomForestClassifier()
```

In [30]:
```
model.fit(x_train,y_train)
```

Out[30]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [32]:
```
model.score(x_test,y_test) @ the accuracy is very high
```

Out[32]:
```
0.9888888888888889
```

In [34]:
```
ypredicted=model.predict(x_test)
```

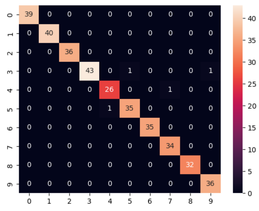
In [37]:
```
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,ypredicted)
```

In [38]:
```
cm
```

Out[38]:
```
array([[39,  0,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0, 40,  0,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0, 36,  0,  0,  0,  0,  0,  0,  0],
       [ 0,  0,  0, 43,  0,  1,  0,  0,  0,  1],
       [ 0,  0,  0,  0, 26,  0,  0,  1,  0,  0],
       [ 0,  0,  0,  0,  1, 35,  0,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0, 35,  0,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0, 34,  0,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0, 32,  0],
       [ 0,  0,  0,  0,  0,  0,  0,  0,  0, 36]], dtype=int64)
```

In [40]:
```
sns.heatmap(cm,annot=True)
```

Out[40]:
```
<AxesSubplot: >
```



In [41]:
```
# here the 39 times the value was 0 and predicted 0
'''1 times the value was 5 then predicted as 4'''
```

Out[41]: '1 times the value was 5 then predicted as 4'

In [ ]: