# Q2 Artificial Neral Network (ANN) for Digits (0-9) Classification

In [1]:

```
import matplotlib.pyplot as plt
import numpy as np
import cv2
import os
import tensorflow as tf
```

In [2]:

```
# For one hot encoding our labels as they are multiclass
def one_hot_encode(label):
    encoded = [0 for _ in range(10)]
    encoded[label] = 1
    return encoded
```

## Reading images from relevant directory

In [3]:

```
digits_images = []
labels = []
for i in range(0,10):
    for file in os.listdir('data-digits/{}'.format(i)):
        labels.append(one_hot_encode(i))
        image = cv2.imread('data-digits/{}/{}'.format(i, file), cv2.IMREAD_GRAYSCALE)
        digits_images.append(image)

digits_images = np.where(np.array(digits_images)==255, 0, 1)  # To binary form from grayscale image, if you pass grayscale image network would not learn at all (I tried)
labels = np.array(labels)
print(np.array(digits_images).shape)
print(np.array(labels).shape)
```

```
(500, 100, 100)
(500, 10)
```

In [12]:

```
print(digits_images[0:2], '\n')
print(labels[0:2])
plt.imshow(digits_images[80], cmap='binary')
```

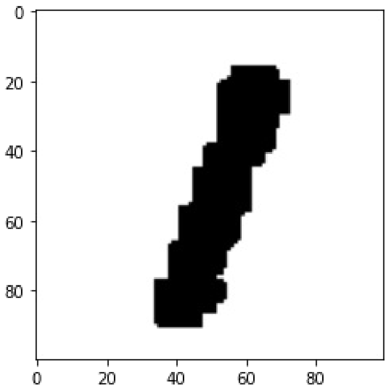
```
[[[0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  ...
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]]

 [[0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  ...
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]
  [0 0 0 ... 0 0 0]]] 

[[1 0 0 0 0 0 0 0 0 0]
 [1 0 0 0 0 0 0 0 0 0]]
```

Out[12]:

```
<matplotlib.image.AxesImage at 0x7f62c82d0ac0>
```




# Justification

- We would be using **Tensorflow** for making ANN, as it was **mentioned in video recording that we can use any machine library for q2**
- Input shape would be (100,100,) as each this is the dimension of each image but we would be flattening this into 1D array
- This would be passed to the layer with neurons 128, I tried 1000 neurons but it was taking too much time in each epoch to update weights
- Then I used layer with 64 neurons, notice that 64 and 128 are combination of 32, this is recommended by some researchers
- Output layer has 10 neurons because we need to classify digits 0-9 which makes this problem **multiclassification** prolem, in multiclassification problem in neural network we tend to use **softmax** as output layer activation function and we cannot use sigmoid as it is used in multilabel problem but in our problem each image cannot be multilabeled as it can be only one of 0-9 digits so thats why softmax.
- Categorical crossentropy is the Error here, as we are also in multiclassification problem

In [5]:

```
model = tf.keras.models.Sequential([
        tf.keras.layers.Flatten(input_shape=(100,100,)),
        tf.keras.layers.Dense(128, activation='relu'),
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dense(10, activation='softmax')])

model.compile(loss='categorical_crossentropy', metrics='accuracy')

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
flatten (Flatten)            (None, 10000)             0         
_________________________________________________________________
dense (Dense)                (None, 128)               1280128   
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256      
_________________________________________________________________
dense_2 (Dense)              (None, 10)                650       
=================================================================
Total params: 1,289,034
Trainable params: 1,289,034
Non-trainable params: 0
_________________________________________________________________
```

In [6]:

```
model.fit(x=digits_images, y=labels, epochs=100)
```

```
Epoch 1/100
16/16 [==============================] - 1s 11ms/step - loss: 3.9197 - accuracy: 0.2214
Epoch 2/100
16/16 [==============================] - 0s 12ms/step - loss: 0.8395 - accuracy: 0.7245
Epoch 3/100
16/16 [==============================] - 0s 12ms/step - loss: 0.6024 - accuracy: 0.7964
Epoch 4/100
16/16 [==============================] - 0s 13ms/step - loss: 0.3817 - accuracy: 0.8889
Epoch 5/100
16/16 [==============================] - 0s 12ms/step - loss: 0.4545 - accuracy: 0.8639
Epoch 6/100
16/16 [==============================] - 0s 12ms/step - loss: 0.1801 - accuracy: 0.9538
Epoch 7/100
16/16 [==============================] - 0s 12ms/step - loss: 0.3983 - accuracy: 0.8948
Epoch 8/100
16/16 [==============================] - 0s 13ms/step - loss: 0.0861 - accuracy: 0.9717
Epoch 9/100
16/16 [==============================] - 0s 19ms/step - loss: 0.1015 - accuracy: 0.9692
Epoch 10/100
16/16 [==============================] - 0s 16ms/step - loss: 0.1106 - accuracy: 0.9629
Epoch 11/100
16/16 [==============================] - 0s 12ms/step - loss: 0.2418 - accuracy: 0.9530
Epoch 12/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0544 - accuracy: 0.9838
Epoch 13/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0628 - accuracy: 0.9823
Epoch 14/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0071 - accuracy: 1.0000
Epoch 15/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0356 - accuracy: 0.9912
Epoch 16/100
16/16 [==============================] - 0s 15ms/step - loss: 0.0270 - accuracy: 0.9884
Epoch 17/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0218 - accuracy: 0.9890
Epoch 18/100
16/16 [==============================] - 0s 11ms/step - loss: 0.0016 - accuracy: 1.0000
Epoch 19/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0504 - accuracy: 0.9960
Epoch 20/100
16/16 [==============================] - 0s 16ms/step - loss: 0.0152 - accuracy: 0.9963
Epoch 21/100
16/16 [==============================] - 0s 14ms/step - loss: 9.8821e-04 - accuracy: 1.0000
Epoch 22/100
16/16 [==============================] - 0s 17ms/step - loss: 4.8841e-04 - accuracy: 1.0000
Epoch 23/100
16/16 [==============================] - 0s 16ms/step - loss: 0.1308 - accuracy: 0.9696
Epoch 24/100
```

```
16/16 [==============================] - 0s 15ms/step - loss: 0.0013 - accuracy: 1.0000
Epoch 25/100
16/16 [==============================] - 0s 12ms/step - loss: 3.7428e-04 - accuracy: 1.0000
Epoch 26/100
16/16 [==============================] - 0s 12ms/step - loss: 2.3928e-04 - accuracy: 1.0000
Epoch 27/100
16/16 [==============================] - 0s 12ms/step - loss: 8.8571e-04 - accuracy: 0.9995
Epoch 28/100
16/16 [==============================] - 0s 11ms/step - loss: 0.1894 - accuracy: 0.9233
Epoch 29/100
16/16 [==============================] - 0s 13ms/step - loss: 4.9054e-04 - accuracy: 1.0000
Epoch 30/100
16/16 [==============================] - 0s 14ms/step - loss: 1.7580e-04 - accuracy: 1.0000
Epoch 31/100
16/16 [==============================] - 0s 12ms/step - loss: 1.0557e-04 - accuracy: 1.0000
Epoch 32/100
16/16 [==============================] - 0s 12ms/step - loss: 7.7407e-05 - accuracy: 1.0000
Epoch 33/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0815 - accuracy: 0.9900
Epoch 34/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0011 - accuracy: 1.0000
Epoch 35/100
16/16 [==============================] - 0s 12ms/step - loss: 9.1215e-05 - accuracy: 1.0000
Epoch 36/100
16/16 [==============================] - 0s 12ms/step - loss: 4.2920e-05 - accuracy: 1.0000
Epoch 37/100
16/16 [==============================] - 0s 12ms/step - loss: 4.5968e-05 - accuracy: 1.0000
Epoch 38/100
16/16 [==============================] - 0s 11ms/step - loss: 1.9726e-05 - accuracy: 1.0000
Epoch 39/100
16/16 [==============================] - 0s 12ms/step - loss: 1.2165e-05 - accuracy: 1.0000
Epoch 40/100
16/16 [==============================] - 0s 13ms/step - loss: 0.2881 - accuracy: 0.9465
Epoch 41/100
16/16 [==============================] - 0s 12ms/step - loss: 6.2954e-04 - accuracy: 1.0000
Epoch 42/100
16/16 [==============================] - 0s 12ms/step - loss: 0.0012 - accuracy: 1.0000
Epoch 43/100
16/16 [==============================] - 0s 12ms/step - loss: 8.1577e-05 - accuracy: 1.0000
Epoch 44/100
16/16 [==============================] - 0s 12ms/step - loss: 2.6788e-05 - accuracy: 1.0000
Epoch 45/100
16/16 [==============================] - 0s 11ms/step - loss: 2.0749e-05 - accuracy: 1.0000
Epoch 46/100
16/16 [==============================] - 0s 13ms/step - loss: 1.5235e-05 - accuracy: 1.0000
Epoch 47/100
16/16 [==============================] - 0s 11ms/step - loss: 1.0740e-05 - accuracy: 1.0000
Epoch 48/100
16/16 [==============================] - 0s 12ms/step - loss: 4.6678e-06 - accuracy: 1.0000
Epoch 49/100
16/16 [==============================] - 0s 12ms/step - loss: 0.3720 - accuracy: 0.9654
Epoch 50/100
16/16 [==============================] - 0s 12ms/step - loss: 4.0452e-05 - accuracy: 1.0000
Epoch 51/100
16/16 [==============================] - 0s 12ms/step - loss: 1.9567e-05 - accuracy: 1.0000
Epoch 52/100
16/16 [==============================] - 0s 12ms/step - loss: 1.1733e-05 - accuracy: 1.0000
Epoch 53/100
16/16 [==============================] - 0s 18ms/step - loss: 1.0607e-05 - accuracy: 1.0000
Epoch 54/100
16/16 [==============================] - 0s 15ms/step - loss: 5.5534e-06 - accuracy: 1.0000
Epoch 55/100
16/16 [==============================] - 0s 17ms/step - loss: 4.8554e-06 - accuracy: 1.0000
Epoch 56/100
16/16 [==============================] - 0s 14ms/step - loss: 2.8922e-06 - accuracy: 1.0000
Epoch 57/100
16/16 [==============================] - 0s 17ms/step - loss: 1.7020e-06 - accuracy: 1.0000
Epoch 58/100
16/16 [==============================] - 0s 18ms/step - loss: 1.3698e-06 - accuracy: 1.0000
Epoch 59/100
16/16 [==============================] - 0s 14ms/step - loss: 0.1011 - accuracy: 0.9893
Epoch 60/100
16/16 [==============================] - 0s 13ms/step - loss: 1.9422e-04 - accuracy: 1.0000
Epoch 61/100
16/16 [==============================] - 0s 15ms/step - loss: 3.6977e-05 - accuracy: 1.0000
Epoch 62/100
16/16 [==============================] - 0s 14ms/step - loss: 2.1479e-05 - accuracy: 1.0000
Epoch 63/100
16/16 [==============================] - 0s 12ms/step - loss: 1.2572e-05 - accuracy: 1.0000
Epoch 64/100
```

```
16/16 [==============================] - 0s 16ms/step - loss: 5.9464e-06 - accuracy: 1.0000
Epoch 65/100
16/16 [==============================] - 0s 14ms/step - loss: 2.8474e-06 - accuracy: 1.0000
Epoch 66/100
16/16 [==============================] - 0s 14ms/step - loss: 2.9407e-06 - accuracy: 1.0000
Epoch 67/100
16/16 [==============================] - 0s 11ms/step - loss: 1.6238e-06 - accuracy: 1.0000
Epoch 68/100
16/16 [==============================] - 0s 12ms/step - loss: 1.0311e-06 - accuracy: 1.0000
Epoch 69/100
16/16 [==============================] - 0s 18ms/step - loss: 6.2829e-07 - accuracy: 1.0000
Epoch 70/100
16/16 [==============================] - 0s 12ms/step - loss: 4.4504e-07 - accuracy: 1.0000
Epoch 71/100
16/16 [==============================] - 0s 12ms/step - loss: 6.4932e-06 - accuracy: 1.0000
Epoch 72/100
16/16 [==============================] - 0s 17ms/step - loss: 0.6276 - accuracy: 0.9422
Epoch 73/100
16/16 [==============================] - 0s 16ms/step - loss: 2.3413e-05 - accuracy: 1.0000
Epoch 74/100
16/16 [==============================] - 0s 14ms/step - loss: 1.0757e-05 - accuracy: 1.0000
Epoch 75/100
16/16 [==============================] - 0s 16ms/step - loss: 7.3890e-06 - accuracy: 1.0000
Epoch 76/100
16/16 [==============================] - 0s 13ms/step - loss: 5.9178e-06 - accuracy: 1.0000
Epoch 77/100
16/16 [==============================] - 0s 13ms/step - loss: 2.6195e-06 - accuracy: 1.0000
Epoch 78/100
16/16 [==============================] - 0s 11ms/step - loss: 2.3008e-06 - accuracy: 1.0000
Epoch 79/100
16/16 [==============================] - 0s 13ms/step - loss: 1.1836e-06 - accuracy: 1.0000
Epoch 80/100
16/16 [==============================] - 0s 13ms/step - loss: 6.7618e-07 - accuracy: 1.0000
Epoch 81/100
16/16 [==============================] - 0s 17ms/step - loss: 4.2560e-07 - accuracy: 1.0000
Epoch 82/100
16/16 [==============================] - 0s 16ms/step - loss: 3.3369e-07 - accuracy: 1.0000
Epoch 83/100
16/16 [==============================] - 0s 20ms/step - loss: 1.6350e-07 - accuracy: 1.0000
Epoch 84/100
16/16 [==============================] - 0s 19ms/step - loss: 1.1096e-07 - accuracy: 1.0000
Epoch 85/100
16/16 [==============================] - 0s 17ms/step - loss: 6.4643e-08 - accuracy: 1.0000
Epoch 86/100
16/16 [==============================] - 0s 13ms/step - loss: 4.5711e-08 - accuracy: 1.0000
Epoch 87/100
16/16 [==============================] - 0s 12ms/step - loss: 3.0814e-08 - accuracy: 1.0000
Epoch 88/100
16/16 [==============================] - 0s 19ms/step - loss: 2.6631e-08 - accuracy: 1.0000
Epoch 89/100
16/16 [==============================] - 0s 20ms/step - loss: 1.5367e-08 - accuracy: 1.0000
Epoch 90/100
16/16 [==============================] - 0s 17ms/step - loss: 9.4192e-09 - accuracy: 1.0000
Epoch 91/100
16/16 [==============================] - 0s 17ms/step - loss: 5.5820e-09 - accuracy: 1.0000
Epoch 92/100
16/16 [==============================] - 0s 14ms/step - loss: 5.6165e-09 - accuracy: 1.0000
Epoch 93/100
16/16 [==============================] - 0s 12ms/step - loss: 3.1735e-09 - accuracy: 1.0000
Epoch 94/100
16/16 [==============================] - 0s 14ms/step - loss: 3.9133e-09 - accuracy: 1.0000
Epoch 95/100
16/16 [==============================] - 0s 13ms/step - loss: 3.9634e-09 - accuracy: 1.0000
Epoch 96/100
16/16 [==============================] - 0s 11ms/step - loss: 4.8179e-09 - accuracy: 1.0000
Epoch 97/100
16/16 [==============================] - 0s 12ms/step - loss: 2.9737e-09 - accuracy: 1.0000
Epoch 98/100
16/16 [==============================] - 0s 19ms/step - loss: 3.0331e-09 - accuracy: 1.0000
Epoch 99/100
16/16 [==============================] - 0s 12ms/step - loss: 2.7370e-09 - accuracy: 1.0000
Epoch 100/100
16/16 [==============================] - 0s 12ms/step - loss: 2.1734e-09 - accuracy: 1.0000
```

Out[6]:

```
<tensorflow.python.keras.callbacks.History at 0x7f62d4b6e0a0>
```

In [8]:

```python
for ind, image in enumerate(digits_images):
    print('actual:', np.argmax(labels[ind]), 'predicted:', np.argmax((model.predict(np.array([digits_images[
ind]])))))
```

```
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 0 predicted: 0
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
```

```
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 1 predicted: 1
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 2 predicted: 2
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
```

```
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 3 predicted: 3
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
```

```
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 4 predicted: 4
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 5 predicted: 5
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
```

```
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 6 predicted: 6
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
```

```
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 7 predicted: 7
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 8 predicted: 8
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
```

```
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
actual: 9 predicted: 9
```

In [ ]:

```

```