

1.

```
plt.imshow(X[i])  
plt.title(f"Image {i} label={label_names[Y[i]]} (num {Y[i]})")  
plt.pause(1)
```

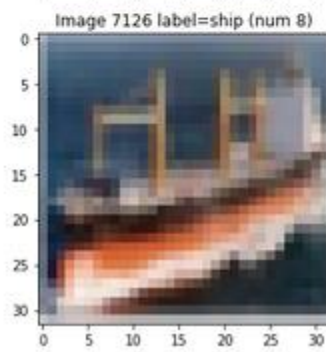
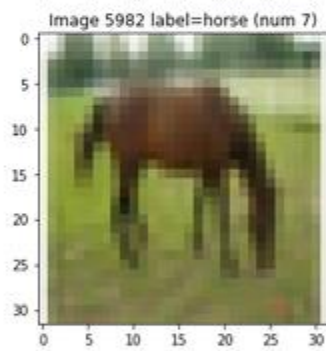
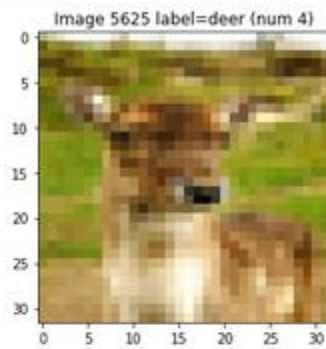


Image 8570 label=automobile (num 1)

2.

```
[10] def class_acc(pred, gt):  
    y_actual = list(gt)  
    predicted = list(pred)  
    score = 0  
  
    for i, j in zip(predicted, y_actual):  
        if i == j:  
            score += 1  
  
    return round((score / len(y_actual))*100, 2), "%", "The correct classified images are:", score  
  
[11] class_acc(Y,Y) # for perfect accuracy  
(100.0, '%', 'The correct classified images are:', 10000)  
  
[12] class_acc(predLab,Y)  
(10.06, '%', 'The correct classified images are:', 1006)
```

3.

### ▼ 3. CIFAR-10 – Random classifier

```
[53] x = unpickle('/content/drive/MyDrive/Excercise_3/test_batch')  
def cifar10_classifier_random(x):  
    lis = []  
    x = x['labels']  
    for i in range(len(x)):  
  
        ranVal = random.choice(x)  
        lis.append(ranVal)  
    return lis  
  
lis = cifar10_classifier_random(x)  
  
class_acc(lis,Y)  
(10.41, '%', 'The correct classified images are:', 1041)
```

4.

```
def cifar10_classifier_1nn(x,xtran,trlabels):
    predicted = []
    IndTrlabel = []
    label = []
    for ind,i in enumerate(x):
        minimum = []
        for index,j in enumerate(xtran):
            mini = np.linalg.norm(i-j)
            minimum.append(mini)
            IndTrlabel.append(minimum.index(min(minimum)))

    for ind,val in enumerate(IndTrlabel):
        predicted.append(trlabels[val])

    return predicted

[22] predicted = cifar10_classifier_1nn(x,xtran,trlables) # these are the predicted label using 1nn classifier.

[41] x = unpickle('/content/drive/MyDrive/Excercise_3/test_batch') # reading again just for accuracy purpose.
xlable = np.array(x['labels'],dtype= np.uint16)

[51] class_acc(predicted,xlable)

(12.02, '%', 'The correct classified images are:', 1202)
```