Answer 2. Code

```
 1 import numpy as np
 2
 3 gridLength = 28 * 28
 4 firstHalf = 5000
 5 end = 10000
 6 weight = np.zeros((10, gridLength + 1))
 7 # Initialize starting values
 8 prediction = activation = accuracy = 0
 9 data = np.loadtxt('mnist_data.txt', dtype=int, delimiter=
   ' ')
10 label = np.loadtxt('mnist_labels.txt', dtype=int, delimiter
   =' ')
11
12 # for training:
13 for i in range(0, firstHalf):
14     yT = yD = 0
15     for j in range(0, 10):
16         for k in range(0, gridLength):
17             activation = activation + weight[j][k] * data[i
   ][k]
18         if yT < activation:
19             yT = activation
20             yD = j
21         activation = 0
22     if yD != label[i]:
23         y = label[i]
24         for j in range(0, 784):
25             weight[yD][j] = weight[yD][j] - data[i][j]
26             weight[y][j] = weight[y][j] + data[i][j]
27 # checking the accuracy:
28 for i in range(0, firstHalf):
29     yT = yD = 0
30     for j in range(0, 10):
31         for k in range(0, gridLength):
32             activation = activation + weight[j][k] * data[i
   ][k]
33         if yT < activation:
34             yT = activation
35             yD = j
36         activation = 0
37     if yD == label[i]:
38         accuracy = accuracy + 1
39 print('Accuracy of classifier on training set= ' + str((
   accuracy / 5000) * 100) + ' %')
40
41 # for testing:
```

Accuracy

Answer 2. Code

```
42 weight = np.zeros((10, gridLength + 1))
43 # Initialize starting values
44 prediction = activation = accuracy = 0
45 for i in range(firstHalf + 1, end):
46     yT = yD = 0
47     for j in range(0, 10):
48         for k in range(0, gridLength):
49             activation = activation + weight[j][k] * data[i
   ][k]
50         if yT < activation:
51             yT = activation
52             yD = j
53         activation = 0
54     if yD != label[i]:
55         y = label[i]
56         for j in range(0, 784):
57             weight[yD][j] = weight[yD][j] - data[i][j]
58             weight[y][j] = weight[y][j] + data[i][j]
59 # checking the accuracy:
60 for i in range(firstHalf + 1, end):
61     yT = yD = 0
62     for j in range(0, 10):
63         for k in range(0, gridLength):
64             activation = activation + weight[j][k] * data[i
   ][k]
65         if yT < activation:
66             yT = activation
67             yD = j
68         activation = 0
69     if yD == label[i]:
70         accuracy = accuracy + 1
71 print('Accuracy of classifier on test set= ' + str((
   accuracy / 5000) * 100) + ' %')
72
```

Accuracy of classifier on training set= 83.2 %
Accuracy of classifier on test set= 86.44 %

Used the implementation same to answer 1 with multiple labels and have weights for each values.
Convergence Criterion : It completes the loop with all the data in it then it checks for accuracy

Accuracy