Steps to solve Assignment:

- Text in red color is Python code
- Load or read data from txt files in to NumPy arrays
- Don't need to reshape data. Reshape use only for visualization of image
- You can use pandas library for loading data. First load in data frames then in NumPy array as we did in assignment 2

- Train_X
  contain all "trainX.txt" data.
  Size should be (500 , 256)

  ```
  df = pd.read_csv('trainX.txt' , delimiter=' ' , header=None )
  Train_X = np.array(df)
  Train_X = Train_X[:Train_X.shape[0],1:].astype(int))
  ```

- Train_X_2
  contain all 2 class data of Train_X.
  use array slice operator
  Size should be (250 , 256)

  ```
  Train_X_2 = Train_X[ : int(Train_X.shape[0] / 2) , :]
  ```

  Similarly you can make
  Train_X_4 (Do it yourself)

- Train_Y it is a vector. You can convert array into vector by using np.array(df).ravel()

  ```
  df = pd.read_csv('trainY.txt' , delimiter=' ' , header=None)
  Train_Y = np.array(df)
  Train_Y = Train_Y[:Train_Y.shape[0],1:].ravel().astype(int)
  ```

  Train_Y_2 (Do it yourself)
  Train_Y_4 (Do it yourself)

- Similarly for Testing data (Do it yourself)
  Test_X
  Test_X_2
  Test_X_4
  Test_Y
  Test_Y_2
  Test_Y_4

- Then compute probabilities by using only Training Data { Train_X_2 and Train_X_4}

  ```
  Train_X_2_One_Prob = (Train_X_2.sum(axis=0)) / (Train_X_2.shape[0])
  ```

  Similarly for

  Train_X_2_Zero_Prob (Do it yourself)

Train_X_4_One_Prob (Do it yourself)
Train_X_4_Zero_Prob (Do it yourself)

- Then make function that calculate accuracies and return it

  Parameter lists of this function

  Class to be tested
  Class actual answers

  Train_X,
  Test_X_2
  Train_X_2_One_Prob ,
  Train_X_2_Zero_Prob ,
  Train_X_4 , Train_X_4_One_Prob
  Train_X_4_Zero_Prob

  You can pass all this to function
  OR
  only pass tested class and its actual answers and globally access all remaining parameters

  now compute probabilities for class 2

  Prob_Of_2 = np.prod((To_Be_Test * Train_X_2_One_Prob) + ((1 - To_Be_Test) * Train_X_2_Zero_Prob) , axis = 1) * (Train_X_2.shape[0] / Train_X.shape[0])

  Similarly for 4 class

  Prob_Of_4 think yourself

  Compare both **Prob_Of_2** and **Prob_Of_4** and calculate accuracy Hint (using np.where() function)