Design:

a. **General structures:**
   1. Functions: read_digit function, which read the txt files and return the label and vector
      There are local variables label and vector.
   2. Functions: load_data function, which put the data in the digit-training txt into the g_dataSet dictionary and g_dataSet_modify dictionary
      There are global variables g_dataSet and g_dataSet_modify
   3. Functions: show_info function, which output the data you have in training file
   4. Functions: show_info_test function, which output the data you have in testing file
   5. Functions: predict function with input p_vec, find the top 9 nearest digit with p_vec by calculating the distance between each sample and the target vector and therefore predict the number with least 10 distance. And the number is the majority of the 10 least distance.
      Tricks: 1. If there are more than one number with the same number in least 10 distance we pick the number with least distance in the two numbers
      Tricks: 2. If for each d (the data) there are more than 2 vectors whose distance with target is more than 15. We suppose that the target must not be d and therefore escape the checking of later data of d
   6. Functions: Data_mean function, which calculate the mean digit from randomly picked three vectors in dataset and output it as a sample to compare with the target vector. We originally have 50 samples for each number.
      Tricks: 1. We compare the sample mean vector with the mean vectors near it. If both neighbors are far different from it which means both of them have distances more than 8. We suppose that the sample is not ideal and therefore remove it from the list.
   7. Functions: test function, which put the data you get in digit-testing txt into the g_testDataSet
      There are global variables g_testDataSet
   8. Functions: Predict_last function which open the digit-predict.txt and put the information into predict function and print the prediction out

b. Describe the model you implemented:
   1. Your choice of k value:

(1) You first use mean to calculate the average value of digits in three vectors randomly picked in dataset

(2) We compare the sample mean vector with the mean vectors near it. If both neighbors are far different from it which means both of them have distances more than 8. We suppose that the sample is not ideal and therefore remove it from the list.

   2. How the closest neighbors are determined:

(1) With input p_vec, find the top 9 nearest digit with p_vec by calculating the distance between each sample and the target vector and therefore predict the number with least 10 distance. And the number is the majority of the 10 least distance.

(2) If for each d (the data) there are more than 2 vectors whose distance with target is more than 15. We suppose that the target must not be d and therefore escape the checking of later data of d

   3. The rules used in making the prediction:

(1) If there are more than one number with the same number in least 10 distance we pick the number with least distance in the two numbers

c.

(1) We compare the sample mean vector with the mean vectors near it. If both neighbors are far different from it which means both of them have distances more than 8. We suppose that the sample is not ideal and therefore remove it from the list.

(2) If for each d (the data) there are more than 2 vectors whose distance with target is more than 15. We suppose that the target must not be d and therefore escape the checking of later data of d

(3) You first use mean to calculate the average value of digits in three vectors randomly picked in dataset