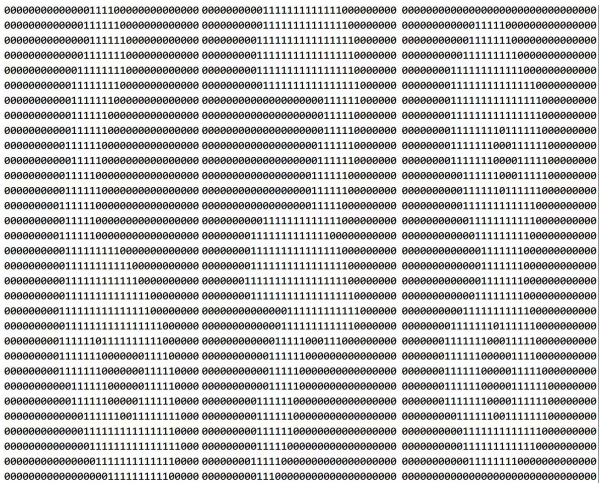
0.1 Application: Handwriting Recognition

We use 32×32 martices to represent the handwritten image, where 1 stands for occupied pixel, and 0 stands for blanket. We have the dataset's dimension deduced to a 1×1024 vector and be compared with the trained data.



Listing 1: **HWR.py**

```
from os import listdir
from K_NN import *

def imageToVec(filename):
     file =  open(filename)
    returnVec = np.zeros([1, 1024], dtype =  int)
    for i in  range(32):
        lineStr =  file.readline()
        for j in  range(32):
            returnVec[0, i * 32 + j] =  int(lineStr[j])
    return returnVec

def handWritingRecognition(filename, dataDir): #The training dataset is stored in ./trainingDigits
    sampleVec = imageToVec(filename)
    trainingFileList = listdir(dataDir)
    listLength =  len(trainingFileList)
    trainingMat = np.zeros([listLength, 1024])
    labelVec = []
    for i in  range(listLength):
        fileName = trainingFileList[i]
        fileStr = fileName.split('.')[0]
        fileClass =  int(fileStr.split('_')[0])
        fileVec = imageToVec('trainingDigits/{}'. format(fileName))
        trainingMat[i,...] = fileVec
```

```
25          labelVec.append(fileClass)
26      return K_NN(sampleVec,  int( input()), trainingMat, labelVec)
```