```
# File: Project3.py
# Student: Jennifer Truong
# UT EID: Jat5244
# Course Name: CS303E
#
# Date Created: 4/29/2021
# Date Last Modified: 5/3/2021
# Description of Program: Building a system to allow requesting information about
totals of Covid-19 cases and deaths in Texas counties and statewide.

import os.path

# Open the file
infile = open( "county-covid-data.txt", "r" )
countyNames = list()
confirmedCases = list()
probableCases = list()
deaths = list()
data = list()
covidData = {}
casesAndDeath = []

readText = infile.readlines()[1: ]
for line in readText:
    if line.startswith("#"):
        pass
    else:
        data.append( line.split(",") )

for eachData in data:
    countyNames.append( eachData[ 0 :  : 4] )
    confirmedCases.append( eachData[ 1 :  : 4] )
    probableCases.append( eachData[ 2 :  : 4] )
    deaths.append( eachData[ 3 :  : 4] )

# Turning the lists of lists into strings or integers
countyNames = [ ''.join(names) for names in countyNames ]
confirmedCases = [ ''.join(numbers) for numbers in confirmedCases ]
confirmedCases = [ int(nums) for nums in confirmedCases ]           # Turning into
integers
probableCases = [ ''.join(number) for number in probableCases ]
deaths = [ ''.join(num) for num in deaths ]
deaths = [ int(digit) for digit in deaths ]                         # Turning into
integers

# Associate (confirmedCases, death) with countyName in the dictionary
for name, cases, death in zip( countyNames , confirmedCases, deaths):
    casesAndDeath.append( (name, cases, death) )

for key, case, dead in casesAndDeath:
    covidData.setdefault( key, [] ).append( (case, dead) )

totalDeaths = sum(deaths)
totalConfirmCases = sum(confirmedCases)

covidData["Texas"] = (totalConfirmCases, totalDeaths)

infile.close()
```

```
def main():
    # Checking if the user's name of the file exist
    if not os.path.isfile( "county-covid-data.txt" ):
        print( "File county-covid-data.txt not found")
        return

    # Printing the welcome message
    print()
    print( """Welcome to the Texas Covid Database Dashboard.
This provides Covid data in Texas as of 1/26/21.
Creating dictionary from file: county-covid-data.txt

Enter any of the following commands:
Help  - list available commands;
Quit - exit this dashboard;
Counties - list all Texas counties;
Cases <countyName>/Texas - confirmed Covid cases in specified county or statewide;
Deaths <countyName>/Texas - Covid deaths in specified county or statewide.""")
    print()

    command = str(input("Please enter a command: "))
    # Parse the command into a list of words (assuming there's no punctuation).
    commWords = command.split()

    # Extract the first word in the command (which is always a one-word command):
    comm = commWords[0]

    # Extract the rest of the words and re-assemble them into a single string,
    # separated by spaces.
    args = commWords[1:]
    arg = " ".join(args)

    while command != (command.lower == "quit"):

        if command.lower() == "help":
            print("""Help - list available commands;
Quit - exit this dashboard;
Counties - list all Texas counties;
Cases <countyName>/Texas - confirmed Covid cases in specified county or statewide;
Deaths <countyName>/Texas - Covid deaths in specified county or statewide.""")
            print()

        elif command.lower() == "counties":
            for county in range(0, len(countyNames), 10):
                print(*countyNames[ county : county + 10], sep = ", ")      # Needs
to add the comma at the end of 10
            print()
        elif command.lower() == "cases texas":
            print("Texas total confirmed Covid cases: " + str(totalConfirmCases) )
            print()

        elif command.lower() == "deaths texas":
            print("Texas total confirmed Covid deaths: " + str(totalDeaths) )
            print()

        elif command.lower() == "quit":
            print("Thank you for using the Texas Covid Database Dashboard.
Goodbye!")
            break
```

```
        elif comm.lower() == "cases":
            if arg.title() in covidData:
                # to isolate the value of confirmed cases from the dictionary
                values = str(covidData.get(arg.title()))
                values = values.replace( "[", "")
                values = values.replace( "]", "")
                values = values.replace( "(", "")
                values = values.replace( ")", "")
                caseValue = ""
                for letter in values:
                    if letter == ",":
                        break
                    else:
                        caseValue += letter
                print( arg.title() + " county has " + caseValue + " confirmed Covid
cases." )
                print()
            elif arg.title() not in covidData:
                print("County " + arg.title() + " is not recognized.")
                print()

        elif comm.lower() == "deaths":
            if arg.title() in covidData:
                # to isolate the value of deaths from the dictionary
                value = str(covidData.get(arg.title()))
                value = value.replace( "[", "")
                value = value.replace( "]", "")
                value = value.replace( "(", "")
                value = value.replace( ")", "")
                caseValues = ""
                for x in value[ : : -1]:
                    if x == ",":
                        break
                    else:
                        caseValues += x

                caseValues = caseValues[:: -1]
                print( arg.title() + " county has" + caseValues + " fatalities." )

                print()
            elif arg.title() not in covidData:
                print("County " + arg.title() + " is not recognized.")
                print()

        else:
            print("Command is not recognized.  Try again!")
            print()

        command = str(input("Please enter a command: "))
        # Parse the command into a list of words (assuming there's no punctuation).
        commWords = command.split()

        # Extract the first word in the command (which is always a one-word
command):
        comm = commWords[0]

        # Extract the rest of the words and re-assemble them into a single string,
        # separated by spaces.
```

```
        args = commWords[1:]
        arg = " ".join(args)
main()
```