Write a program to generate tokens from given input string.

Intro:

- Lexical analysis is the first phase of a compiler.

- It reads character streams from the source code, checks for legal tokens, and passes the data to the syntax analyser when it demands.

- this analysis generates tokens from given string.

Code:

# string = input().split()

keyWords = open("keywords.txt","r")

operators = open("operators.txt","r")

delimiters = open("delimiter.txt","r")

#keywords,op from files

keyW = [char.strip() for char in keyWords]

op = [char.strip() for char in operators]

dl = [char.strip() for char in delimiters]

#input from file

mainFile = open("file.txt",'r')

fileList = mainFile.read().split()

mainList = []

#differenciate list

for word in fileList:

    if ',' in word:

        count = word.count(",")

        temp = word.split(",")

        # print(temp)

        for char in temp:

            mainList.append(char)

            if count > 0:

                mainList.append(",")

            count -= 1

    elif word.endswith(';'):

        mainList.append(word[:-1])

        mainList.append(";")

    else:

        mainList.append(word)

print(mainList)

opFile = open("opFile.txt",'w')

opFile.close()

opFile = open("opFile.txt",'a')

for token in mainList:

    if token in keyW:

        p = str(token + " is Keyword")

        opFile.write(p+'\n')

    elif token in op:

        p = str(token + " is Operator")

        opFile.write(p+'\n')

    elif token in dl:

        p = str(token + " is Delimiter")

        opFile.write(p+'\n')

    elif token.isdecimal():

        p = str(token + " is Digit")

        opFile.write(p+'\n')

    elif token.isalnum():

        p = str(token + " is identifier")

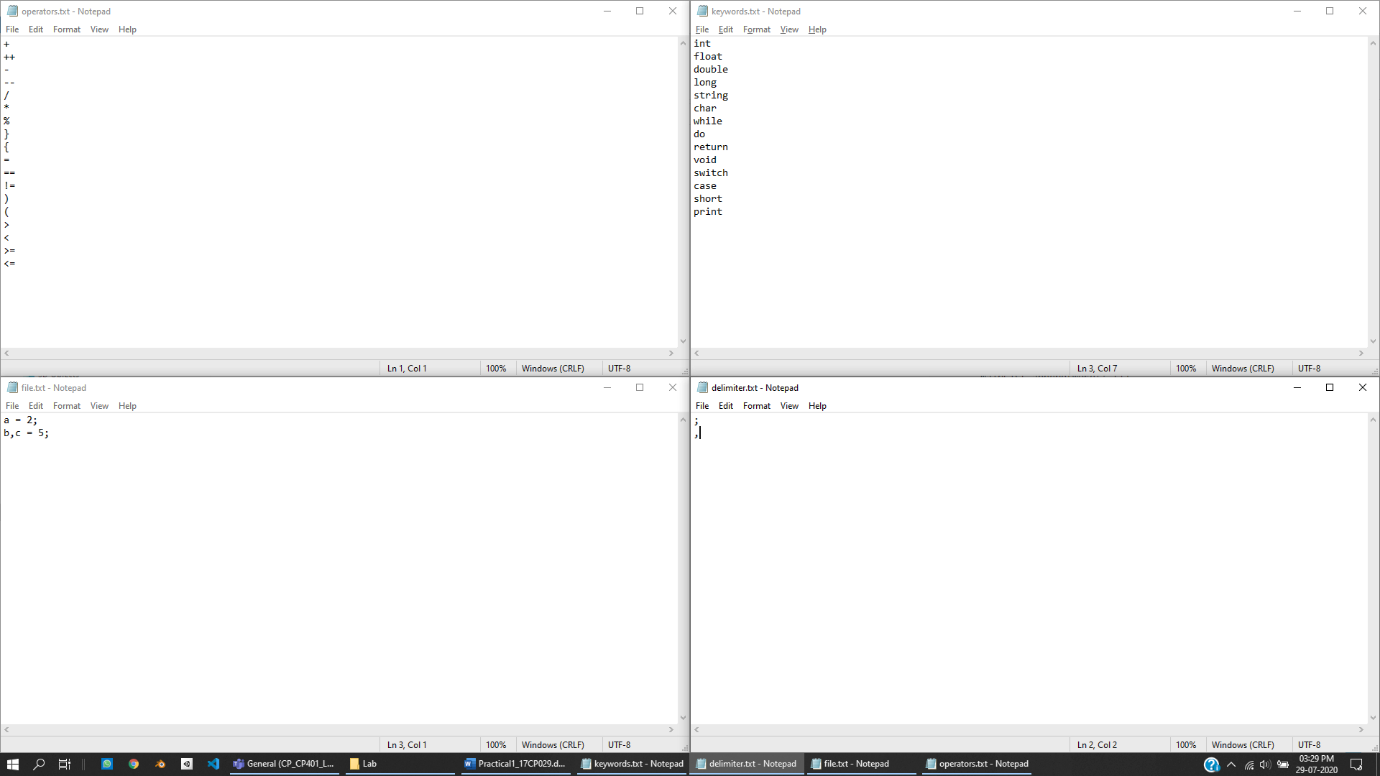
        opFile.write(p+'\n')

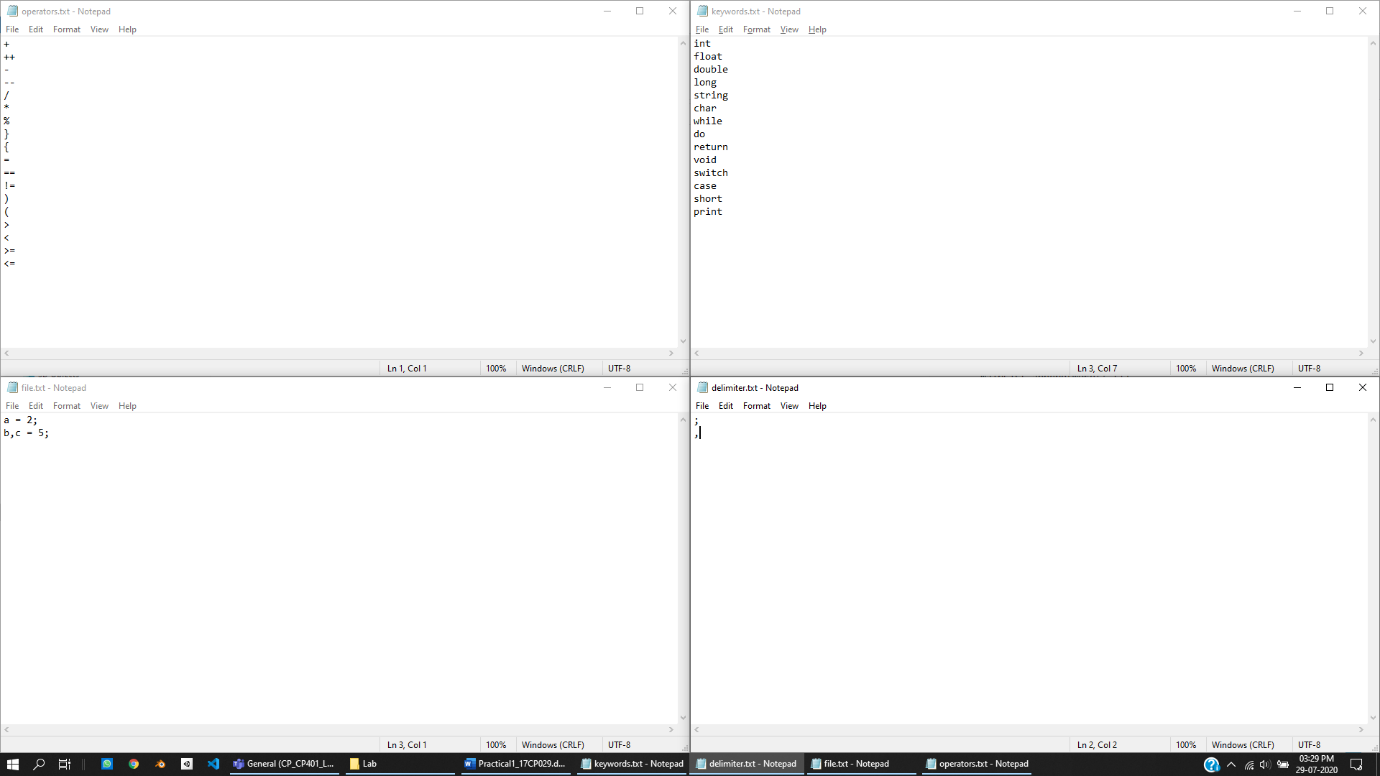
opFile.close()

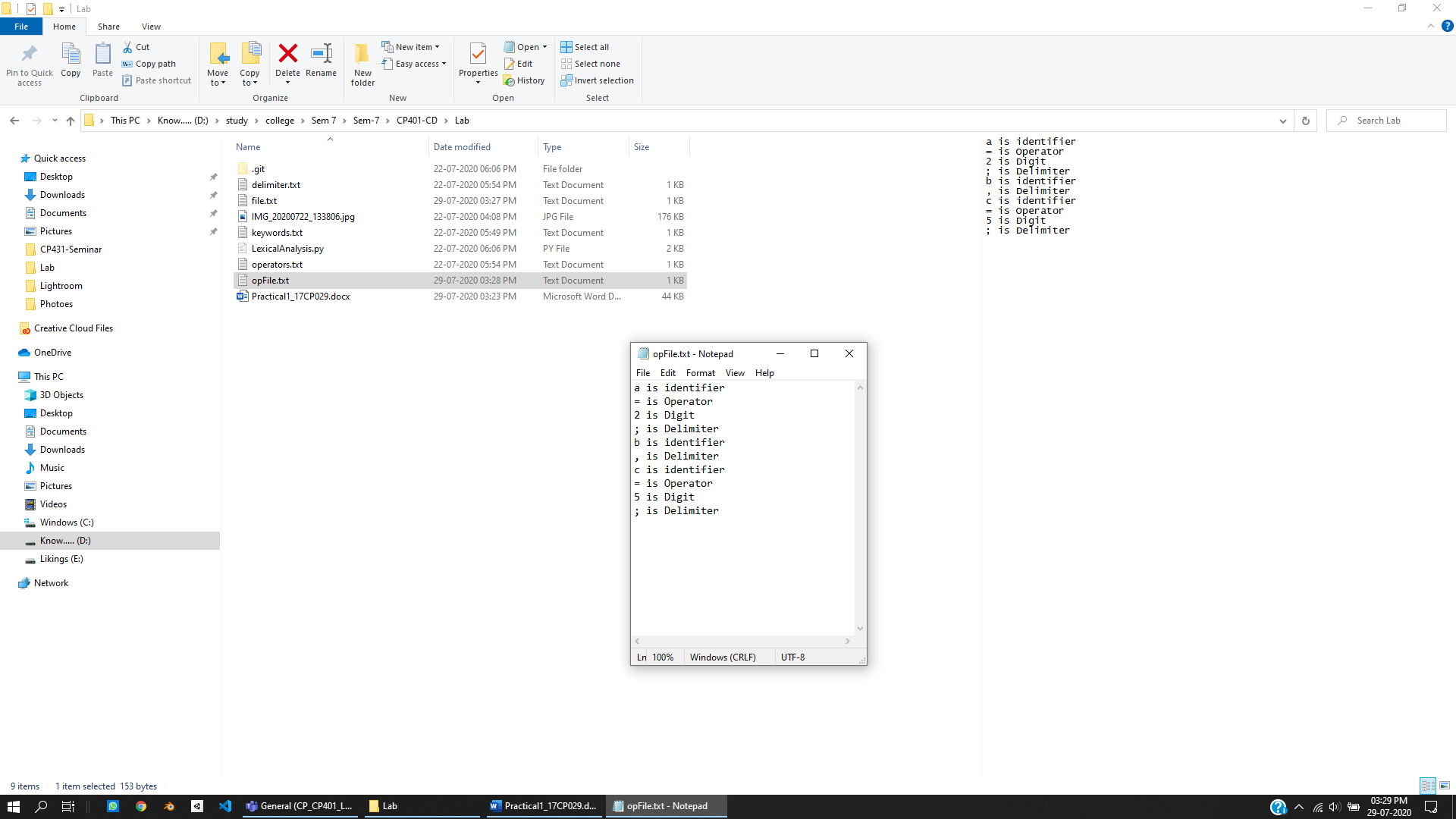
keyWords.close()

operators.close()

mainFile.close()

Files:



Output: