	Marwadi University **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: Programming With Python (01CT1309)	**Aim:** Practical based on File Handling using Python	
Experiment No: 13	**Date:**	**Enrollment No: 92400133037**

Aim: Practical based on File Handling using Python/

IDE:

File handling in Python is a powerful and versatile tool that can be used to perform a wide range of operations. However, it is important to carefully consider the advantages and disadvantages of file handling when writing Python programs, to ensure that the code is secure, reliable, and performs well.

Python provides various functions to perform different file operations, a process known as File Handling.

- ***open()*** *: Opens a file and returns a file object.*
- ***read()*** *: Reads data from a file.*
- ***write()*** *: Writes data to a file.*
- ***close()*** *: Closes the file, releasing its resources.*

Opening Files in Python

In Python, we need to open a file first to perform any operations on it—we use the open() function

Suppose we have a file named ict.txt

To open this file, we can use the open() function.

```
file1 = open("C:\\Users\\Mitesh\\OneDrive\\Desktop \\ict.txt")
```

or

```
file1 = open(r"C:\Users\Mitesh\OneDrive\Desktop \ict.txt")
```

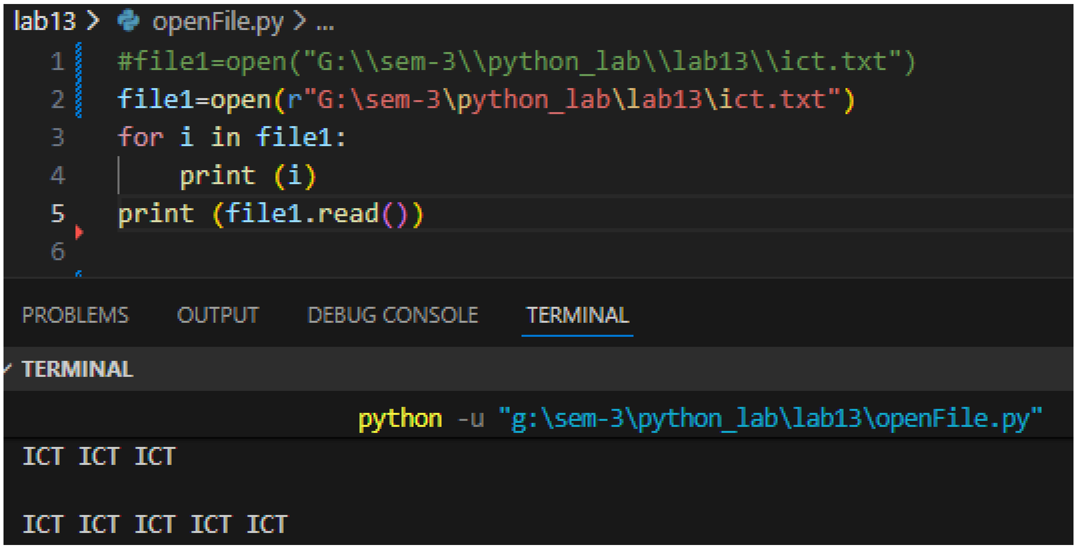
Working in Read mode

The open command will open the Python file in the read mode and the for loop will print each line present in the file.

```
f1 = open(r"C:\Users\Mitesh\OneDrive\Desktop \ict.txt")
# This will print every line one by one in the file
```

for each **in** f1:
   print (each)

Output:



lab13 > openFile.py > ...

```
#file1=open("G:\\sem-3\\python_lab\\lab13\\ict.txt")
file1=open(r"G:\sem-3\python_lab\lab13\ict.txt")
for i in file1:
    print (i)
print (file1.read())
```

```
                 python -u "g:\sem-3\python_lab\lab13\openFile.py"
ICT ICT ICT

ICT ICT ICT ICT ICT
```

In this example, we will extract a string that contains all characters in the Python file then we can use f1.read().

# Python code to illustrate read() mode
f1 = open(r"C:\Users\Mitesh\OneDrive\Desktop \ict.txt")
print (f1.read())

Example

In this example, Read a file using the with statement in Python.

with open(r"C:\Users\Mitesh\OneDrive\Desktop\ict.txt",'r') as f1:
   data = f1.read()
print(data)

Example 4:

Another way to read a file is to call a certain number of characters like in the following code the interpreter will read the first five characters of stored data and return it as a string:

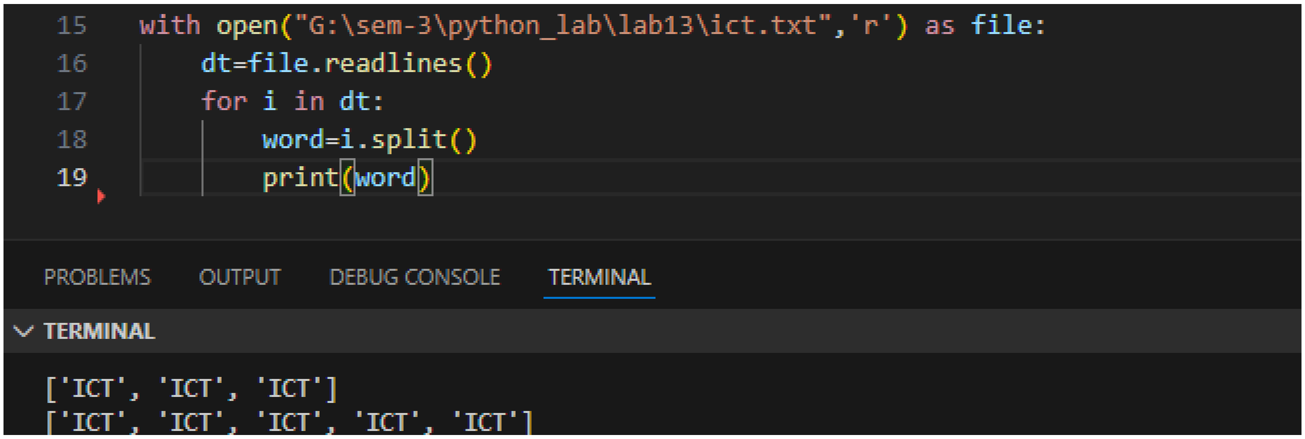
f1 = open(r"C:\Users\Mitesh\OneDrive\Desktop \ict.txt")
print (f1.read(5))

Example

The split() function splits the variable when space is encountered. You can also split using any characters as you wish.

```
with open(r"C:\Users\Mitesh\OneDrive\Desktop\ict.txt",'r') as file:
    data = file.readlines()
    for line in data:
        word = line.split()
        print (word)
```

Output



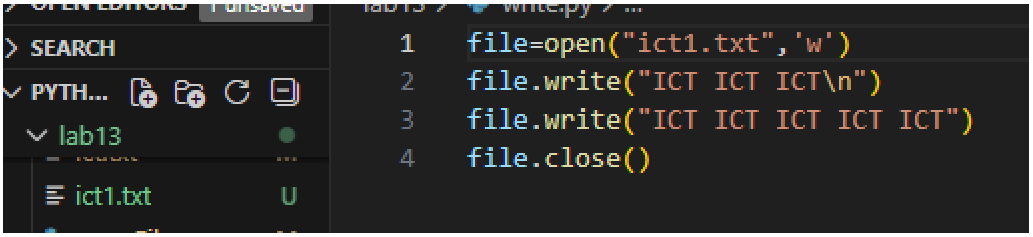
```
with open("G:\sem-3\python_lab\lab13\ict.txt",'r') as file:
    dt=file.readlines()
    for i in dt:
        word=i.split()
        print(word)
```

```
['ICT', 'ICT', 'ICT']
['ICT', 'ICT', 'ICT', 'ICT', 'ICT']
```

Working in Write Mode

The write() function is used to write in a file. The close() command terminates all the resources in use and frees the system of this particular program.

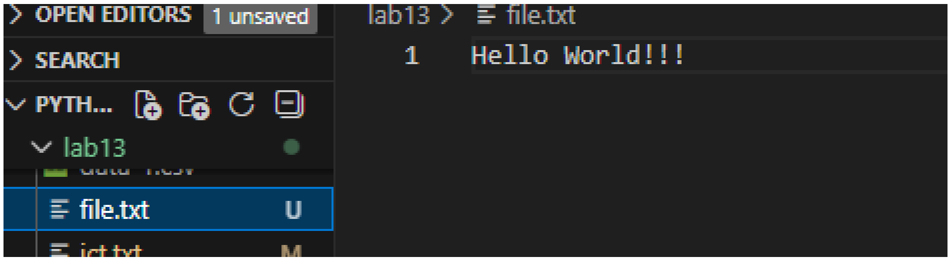
```
file = open("ict1.txt",'w')
file.write("ICT ICT ICT \n")
file.write("ICT ICT ICT ICT ICT")
file.close()
```



```
file=open("ict1.txt",'w')
file.write("ICT ICT ICT\n")
file.write("ICT ICT ICT ICT ICT")
file.close()
```

Using with() function

```
with open("file.txt", "w") as f:
   f.write("Hello World!!!")
   f.close()
```



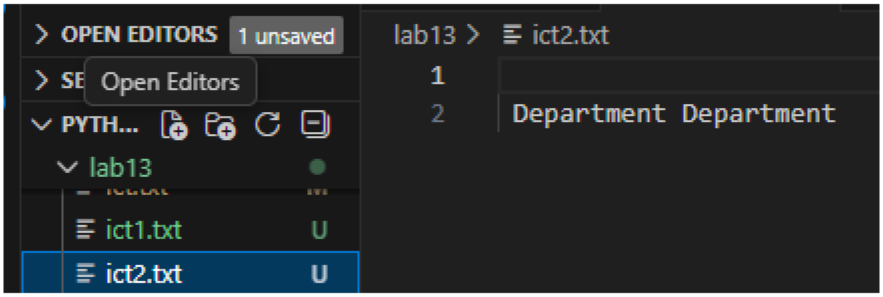
```
Hello World!!!
```

Working of Append Mode

Appending text to an existing file.

```
file = open("ict1.txt",'a')
file.write("\n Department Department")
file.close()
```

Output



```

Department Department
```

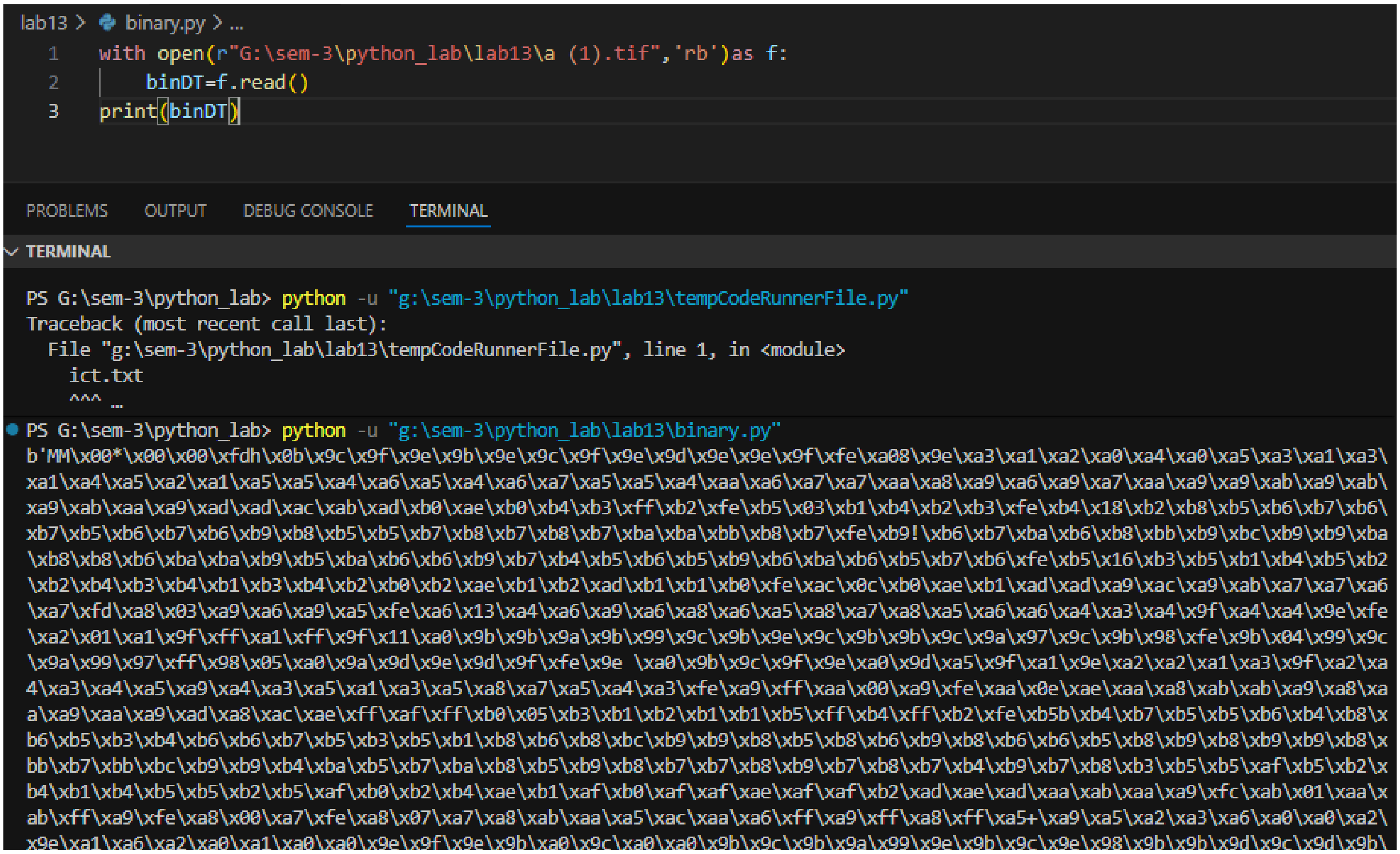
Reading and Writing Binary Files

Reading and writing binary files, such as images.

Rading files

with open(r'C:\Users\Mitesh\OneDrive\Desktop\a.tif', 'rb') as file:
    binary_data = file.read()

Output



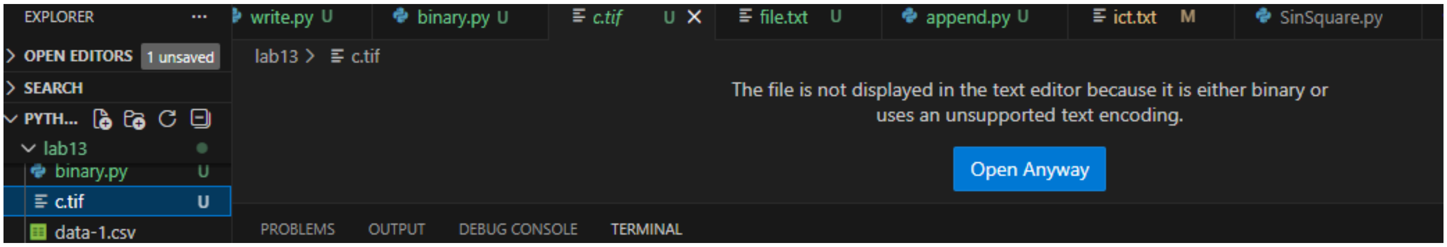
```
with open(r"G:\sem-3\python_lab\lab13\a (1).tif",'rb')as f:
    binDT=f.read()
print(binDT)
```

```
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab13\tempCodeRunnerFile.py"
Traceback (most recent call last):
  File "g:\sem-3\python_lab\lab13\tempCodeRunnerFile.py", line 1, in <module>
    ict.txt
    ^^^ …
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab13\binary.py"
b'MM\x00*\x00\x00\xfdh\x0b\x9c\x9f\x9e\x9b\x9e\x9c\x9f\x9e\x9d\x9e\x9e\x9f\xfe\xa08\x9e\xa3\xa1\xa2\xa0\xa4\xa0\xa5\xa3\xa1\xa3\
xa1\xa4\xa5\xa2\xa1\xa5\xa5\xa4\xa6\xa5\xa4\xa6\xa7\xa5\xa5\xa4\xaa\xa6\xa7\xa7\xaa\xa8\xa9\xa6\xa9\xa7\xaa\xa9\xa9\xab\xa9\xab\
xa9\xab\xaa\xa9\xad\xad\xac\xab\xad\xb0\xae\xb0\xb4\xb3\xff\xb2\xfe\xb5\x03\xb1\xb4\xb2\xb3\xfe\xb4\x18\xb2\xb8\xb5\xb6\xb7\xb6\
xb7\xb5\xb6\xb7\xb6\xb9\xb8\xb5\xb5\xb7\xb8\xb7\xb8\xb7\xba\xba\xbb\xb8\xb7\xfe\xb9!\xb6\xb7\xba\xb6\xb8\xbb\xb9\xbc\xb9\xb9\xba
\xb8\xb8\xb6\xba\xba\xb9\xb5\xba\xb6\xb6\xb9\xb7\xb4\xb5\xb6\xb5\xb9\xb6\xba\xb6\xb5\xb7\xb6\xfe\xb5\x16\xb3\xb5\xb1\xb4\xb5\xb2
\xb2\xb4\xb3\xb4\xb1\xb3\xb4\xb2\xb0\xb2\xae\xb1\xb2\xad\xb1\xb1\xb0\xfe\xac\x0c\xb0\xae\xb1\xad\xad\xa9\xac\xa9\xab\xa7\xa7\xa6
\xa7\xfd\xa8\x03\xa9\xa6\xa9\xa5\xfe\xa6\x13\xa4\xa6\xa9\xa6\xa8\xa6\xa5\xa8\xa7\xa8\xa5\xa6\xa6\xa4\xa3\xa4\x9f\xa4\xa4\x9e\xfe
\xa2\x01\xa1\x9f\xff\xa1\xff\x9f\x11\xa0\x9b\x9b\x9a\x9b\x99\x9c\x9b\x9e\x9c\x9b\x9b\x9c\x9a\x97\x9c\x9b\x98\xfe\x9b\x04\x99\x9c
\x9a\x99\x97\xff\x98\x05\xa0\x9a\x9d\x9e\x9d\x9f\xfe\x9e \xa0\x9b\x9c\x9f\x9e\xa0\x9d\xa5\x9f\xa1\x9e\xa2\xa2\xa1\xa3\x9f\xa2\xa
4\xa3\xa4\xa5\xa9\xa4\xa3\xa5\xa1\xa3\xa5\xa8\xa7\xa5\xa4\xa3\xfe\xa9\xff\xaa\x00\xa9\xfe\xaa\x0e\xae\xaa\xa8\xab\xab\xa9\xa8\xa
a\xa9\xaa\xa9\xad\xa8\xac\xae\xff\xaf\xff\xb0\x05\xb3\xb1\xb2\xb1\xb1\xb5\xff\xb4\xff\xb2\xfe\xb5b\xb4\xb7\xb5\xb5\xb6\xb4\xb8\x
b6\xb5\xb3\xb4\xb6\xb6\xb7\xb5\xb3\xb5\xb1\xb8\xb6\xb8\xbc\xb9\xb9\xb8\xb5\xb8\xb6\xb9\xb8\xb6\xb6\xb5\xb8\xb9\xb8\xb9\xb9\xb8\x
bb\xb7\xbb\xbc\xb9\xb9\xb4\xba\xb5\xb7\xba\xb8\xb5\xb9\xb8\xb7\xb7\xb8\xb9\xb7\xb8\xb7\xb4\xb9\xb7\xb8\xb3\xb5\xb5\xaf\xb5\xb2\x
b4\xb1\xb4\xb5\xb5\xb2\xb5\xaf\xb0\xb2\xb4\xae\xb1\xaf\xb0\xaf\xaf\xae\xaf\xaf\xb2\xad\xae\xad\xaa\xab\xaa\xa9\xfc\xab\x01\xaa\x
ab\xff\xa9\xfe\xa8\x00\xa7\xfe\xa8\x07\xa7\xa8\xab\xaa\xa5\xac\xaa\xa6\xff\xa9\xff\xa8\xff\xa5+\xa9\xa5\xa2\xa3\xa6\xa0\xa0\xa2\
x9e\xa1\xa6\xa2\xa0\xa1\xa0\xa0\x9e\x9f\x9e\x9b\xa0\x9c\xa0\xa0\x9b\x9c\x9b\x9a\x99\x9e\x9b\x9c\x9e\x98\x9b\x9b\x9d\x9c\x9d\x9b\
```

Writing binary files

with open('c.tif', 'wb') as f:
    f.write(binary_data)
    f.close()

Output



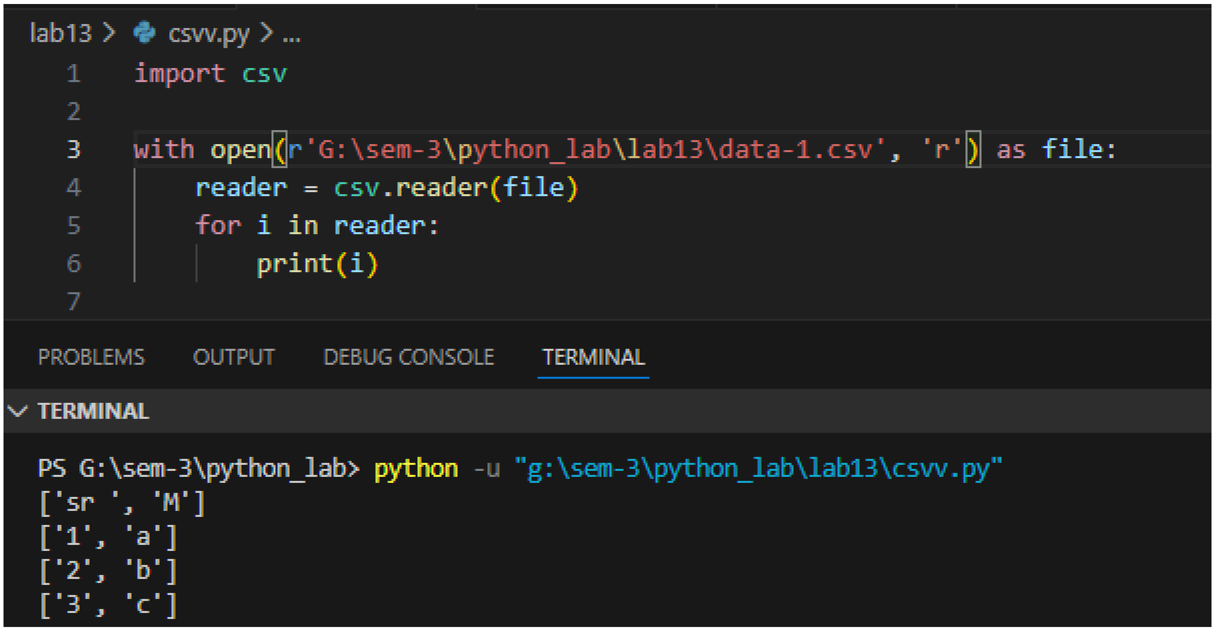
The file is not displayed in the text editor because it is either binary or uses an unsupported text encoding.

Working with CSV Files

import csv

# Reading from a CSV file
with open('data.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)

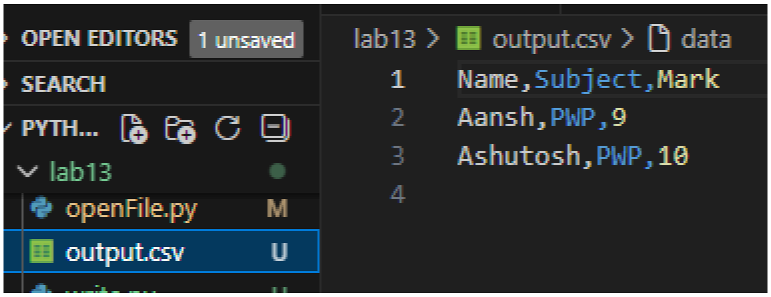
Output



```
import csv

with open(r'G:\sem-3\python_lab\lab13\data-1.csv', 'r') as file:
    reader = csv.reader(file)
    for i in reader:
        print(i)

```

```
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab13\csvv.py"
['sr ', 'M']
['1', 'a']
['2', 'b']
['3', 'c']
```

```
# Writing to a CSV file
import csv

with open('output.csv', 'w', newline='') as file:
    writer = csv.writer(file)
    writer.writerow(['Name', 'Subject', 'Mark'])
    writer.writerow(['Aansh', 'PWP', 9])
    writer.writerow(['Ashutosh', 'PWP', 10])
    file.close()
```



```
Name,Subject,Mark
Aansh,PWP,9
Ashutosh,PWP,10
```

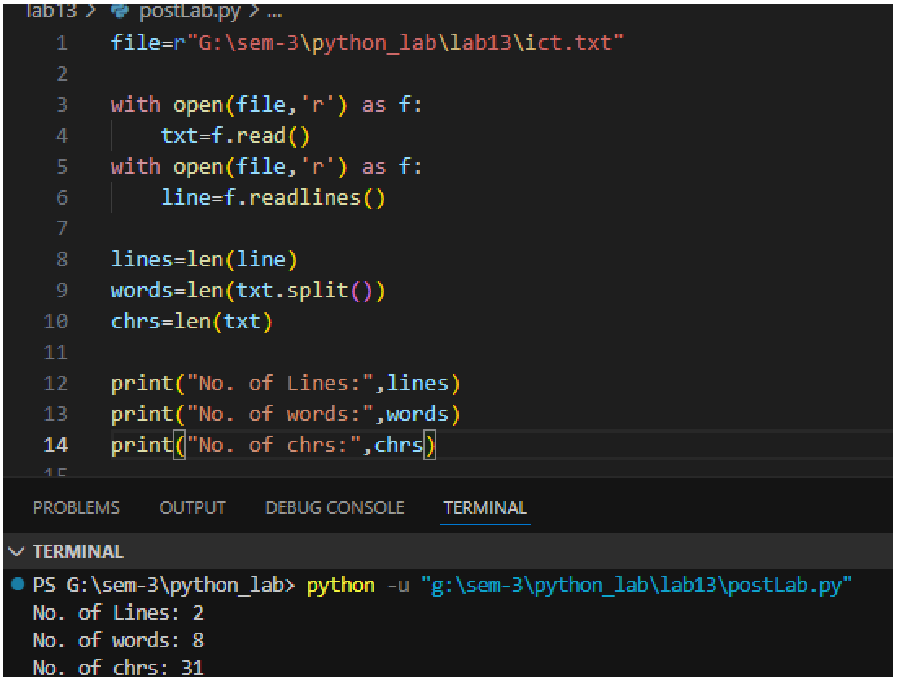
Post Lab Exercise:

a. Write a program that reads a text file example.txt and counts the number of lines, words, and characters in the file. Print these counts.

b. Write a Python program to read a text file line by line and store each line in a list. Print the list after reading the entire file.

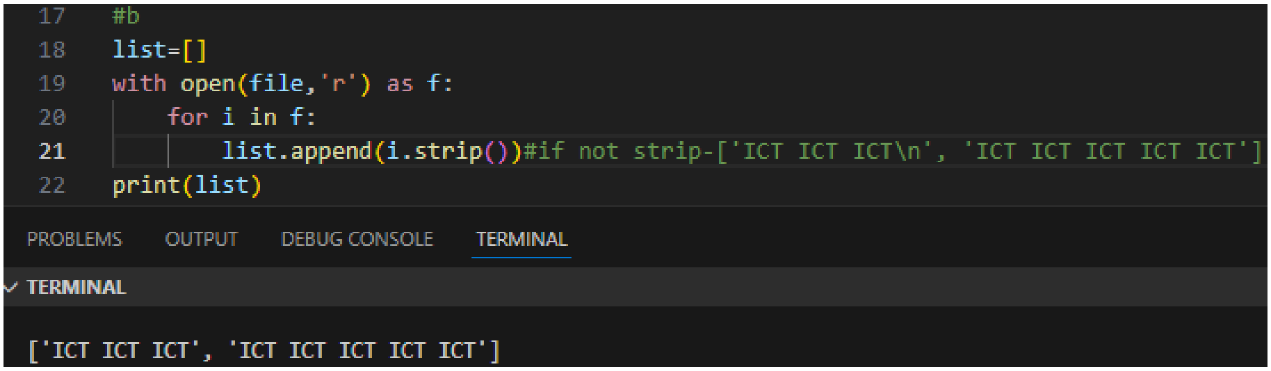
c. Write a Python program to read data from a CSV file data.csv and print each row to the console.

d. Write a Python program that merges the contents of two text files file1.txt and file2.txt into a third file merged.txt. Ensure that the contents of file1.txt come first.



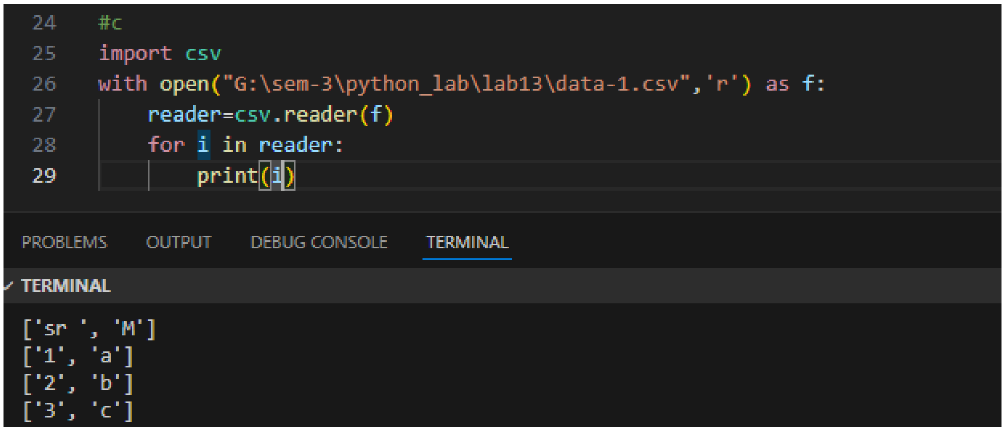
```
file=r"G:\sem-3\python_lab\lab13\ict.txt"

with open(file,'r') as f:
    txt=f.read()
with open(file,'r') as f:
    line=f.readlines()

lines=len(line)
words=len(txt.split())
chrs=len(txt)

print("No. of Lines:",lines)
print("No. of words:",words)
print("No. of chrs:",chrs)
```

```
PS G:\sem-3\python_lab> python -u "g:\sem-3\python_lab\lab13\postLab.py"
No. of Lines: 2
No. of words: 8
No. of chrs: 31
```



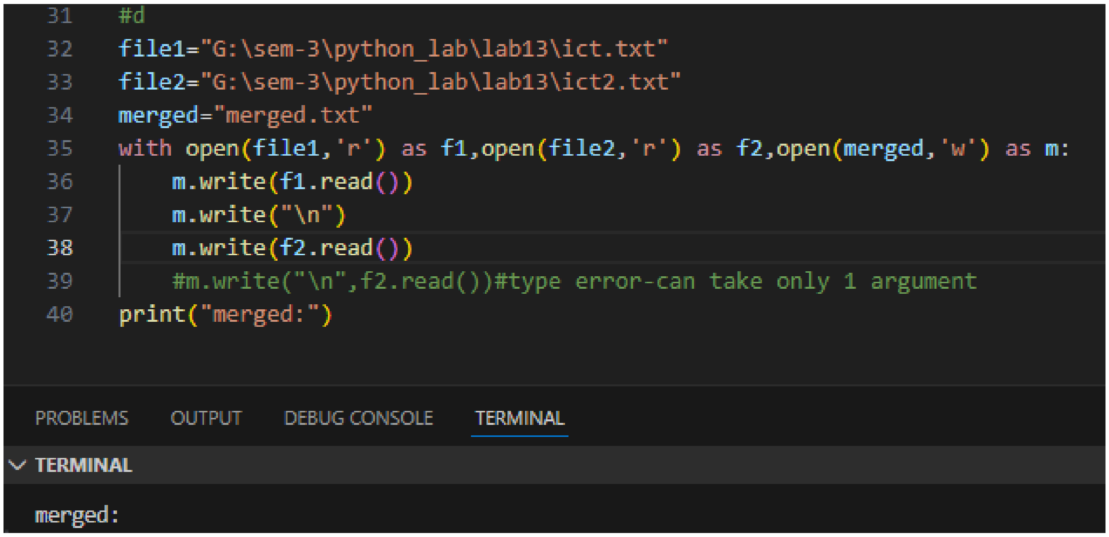
```
#b
list=[]
with open(file,'r') as f:
    for i in f:
        list.append(i.strip())#if not strip-['ICT ICT ICT\n', 'ICT ICT ICT ICT ICT']
print(list)
```

```
['ICT ICT ICT', 'ICT ICT ICT ICT ICT']
```



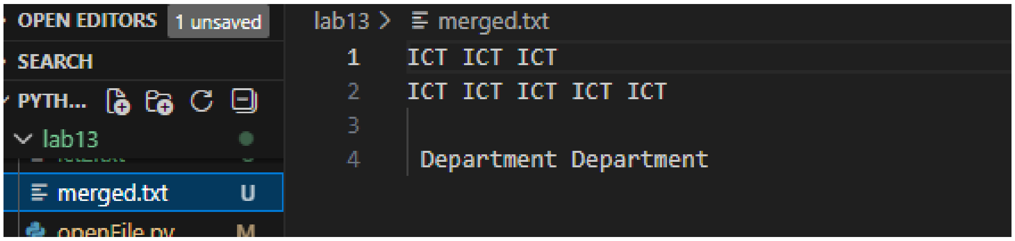
```
#c
import csv
with open("G:\sem-3\python_lab\lab13\data-1.csv",'r') as f:
    reader=csv.reader(f)
    for i in reader:
        print(i)
```

```
['sr ', 'M']
['1', 'a']
['2', 'b']
['3', 'c']
```



```
#d
file1="G:\sem-3\python_lab\lab13\ict.txt"
file2="G:\sem-3\python_lab\lab13\ict2.txt"
merged="merged.txt"
with open(file1,'r') as f1,open(file2,'r') as f2,open(merged,'w') as m:
    m.write(f1.read())
    m.write("\n")
    m.write(f2.read())
    #m.write("\n",f2.read())#type error-can take only 1 argument
print("merged:")
```

```
merged:
```



```
ICT ICT ICT
ICT ICT ICT ICT ICT

 Department Department
```

GITHUB LINK:
https://github.com/Heer972005/Python_Lab