Run examples:

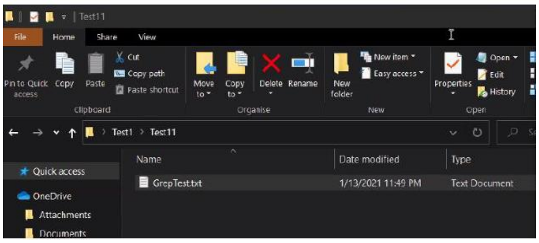
1-Grep command example:

In the following figure, we can see in script.txt, all lines will be executed successfully, but the first line only can find a target to retrieve since “GrepTest” already exists in Test1/Test11 this means that the Grep command searched in the directory and its sub-directories, while the others fail to find a targeted file “GrepFail1” and “GrepFail2” (but still execute).

Notice that the Terminal is where the commands were performed, and a result will be pushed into a python dictionary.

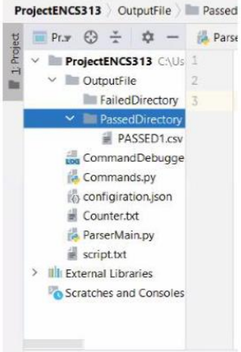
Script.txt:

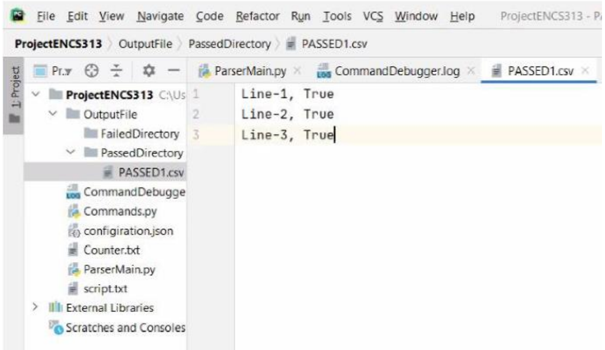
```
Grep <GrepTest> <C:\Users\Tarek\Desktop\Test1>
Grep <GrepFail1> <C:\Users\Tarek\Desktop\Test1>
Grep <GrepFail2> <C:\Users\Tarek\Desktop\Test1\Test11>
```



Name	Date modified	Type
GrepTest.txt	1/13/2021 11:49 PM	Text Document

When execution is done, the code pastes data in the format of logger statements from the above-mentioned dictionary to a new log file. The run created a new “csv” file (output) as demanded in the json file, as the following figures shows:

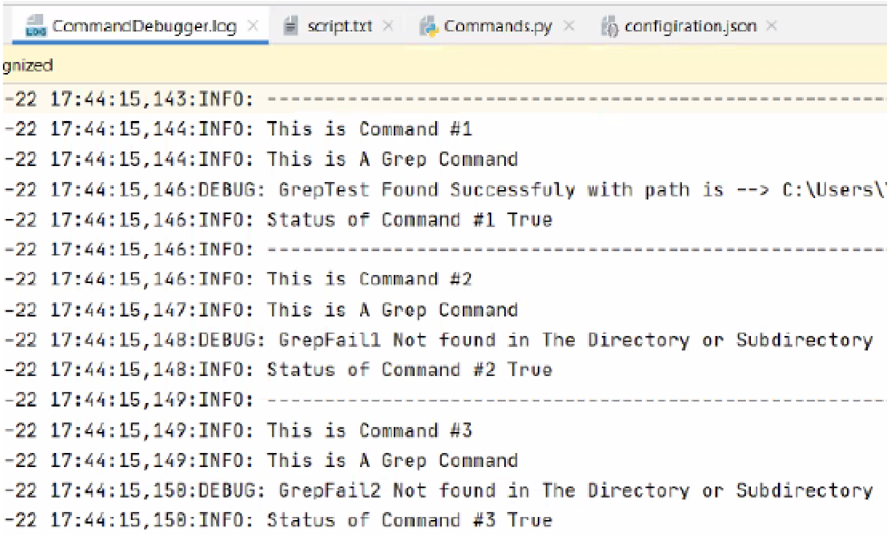




```
Line-1, True
Line-2, True
Line-3, True
```

As we can see that the dictionary was pushed, and the results from each file is shown next to it. note that the location of PASSED1.csv is in subdirectory as demanded in Jason file (Same_dir FALSE).

The following figure shows the debugging for the file. Using the help of logging library.



```
-22 17:44:15,143:INFO: ----------------------------------------------------
-22 17:44:15,144:INFO: This is Command #1
-22 17:44:15,144:INFO: This is A Grep Command
-22 17:44:15,146:DEBUG: GrepTest Found Successfuly with path is --> C:\Users\
-22 17:44:15,146:INFO: Status of Command #1 True
-22 17:44:15,146:INFO: ----------------------------------------------------
-22 17:44:15,146:INFO: This is Command #2
-22 17:44:15,147:INFO: This is A Grep Command
-22 17:44:15,148:DEBUG: GrepFail1 Not found in The Directory or Subdirectory
-22 17:44:15,148:INFO: Status of Command #2 True
-22 17:44:15,149:INFO: ----------------------------------------------------
-22 17:44:15,149:INFO: This is Command #3
-22 17:44:15,149:INFO: This is A Grep Command
-22 17:44:15,150:DEBUG: GrepFail2 Not found in The Directory or Subdirectory
-22 17:44:15,150:INFO: Status of Command #3 True
```

2- Grep, Categorize and Mov commands example:

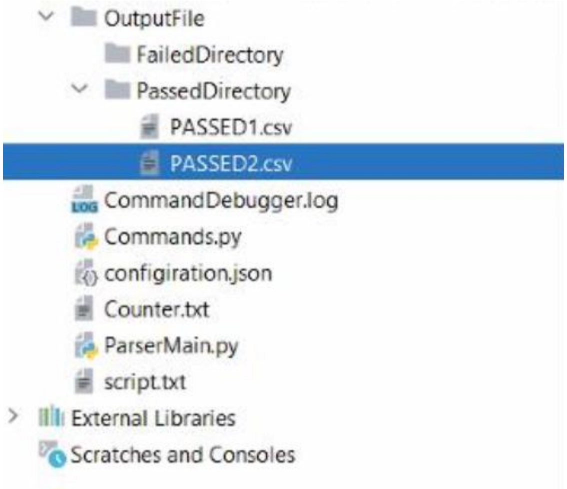
The new script file:

Grep <GrepTest> <C:\Users\Tarek\Desktop\Test1>

Categorize <C:\Users\Tarek\Desktop\Test1>

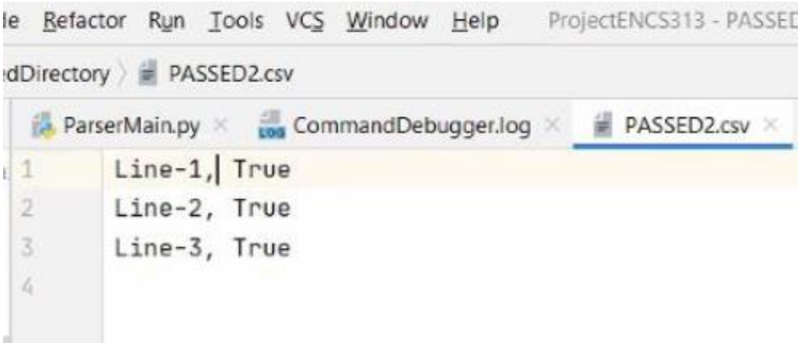
Mv_last <C:\Users\Tarek\Desktop\Test2> <C:\Users\Tarek\Desktop\Test2\MovHere>

After running the program, the following figure shows the new generated “csv” files as demanded in the Jason file.



Notice that the dictionary was pushed, and the results from each file is shown next to it. note that the location of PASSED1.csv is in subdirectory as demanded in Jason file (Same_dir FALSE).

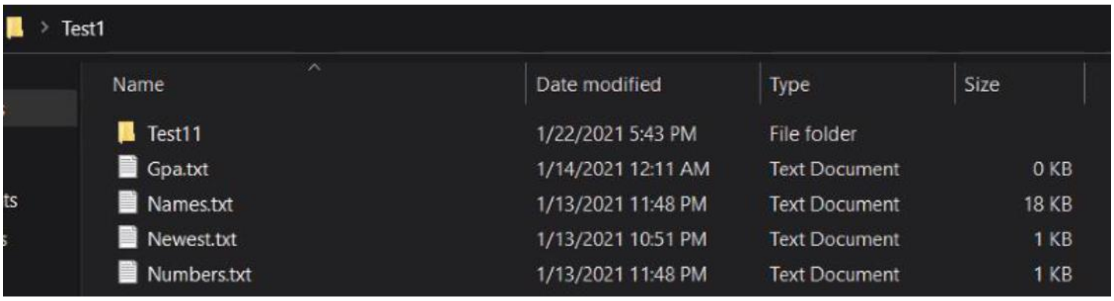
When execution is done, the code pastes data in the format of logger statements from the above-mentioned dictionary to a new log file. The run created a new “csv” file (output) as demanded in the json file, as the following figures shows:



```
Line-1, True
Line-2, True
Line-3, True
```

As we can see that the dictionary was pushed, and the results from each file is shown next to it. note that the location of PASSED2.csv is in subdirectory as demanded in Jason file (Same_dir FALSE).

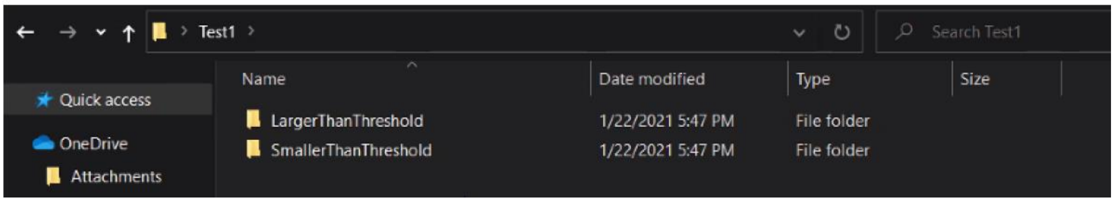
Before (Categorize):



Name	Date modified	Type	Size
Test11	1/22/2021 5:43 PM	File folder	
Gpa.txt	1/14/2021 12:11 AM	Text Document	0 KB
Names.txt	1/13/2021 11:48 PM	Text Document	18 KB
Newest.txt	1/13/2021 10:51 PM	Text Document	1 KB
Numbers.txt	1/13/2021 11:48 PM	Text Document	1 KB

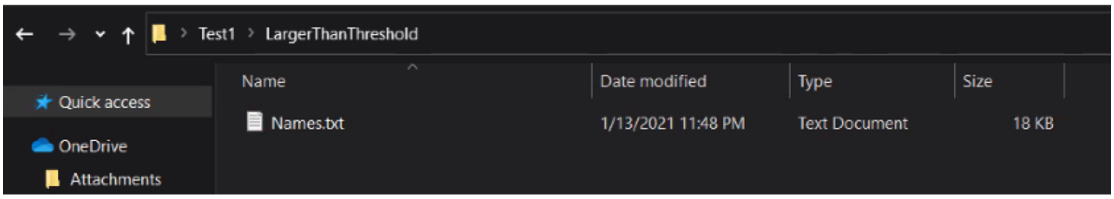
After (Categorize):

The following figure shows the generated directories at the file explorer, two directories were made



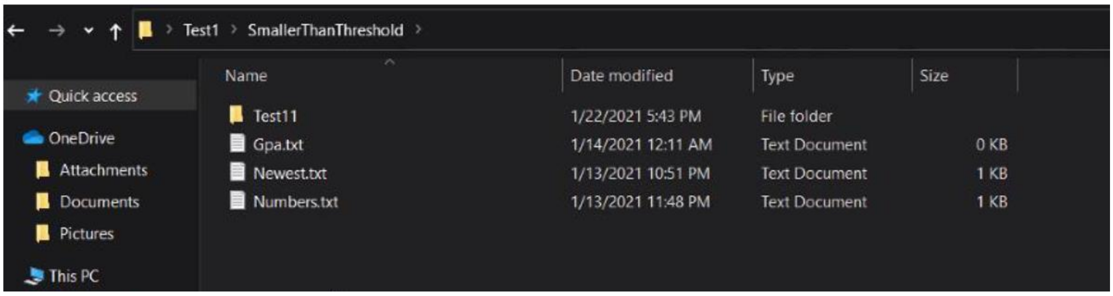
Name	Date modified	Type	Size
LargerThanThreshold	1/22/2021 5:47 PM	File folder	
SmallerThanThreshold	1/22/2021 5:47 PM	File folder	

The first directory “LargerThanThreshold” contains the file which has a size that is larger than the given Threshold value in the Jason file “10KB” as was mentioned in the project.



Name	Date modified	Type	Size
Names.txt	1/13/2021 11:48 PM	Text Document	18 KB

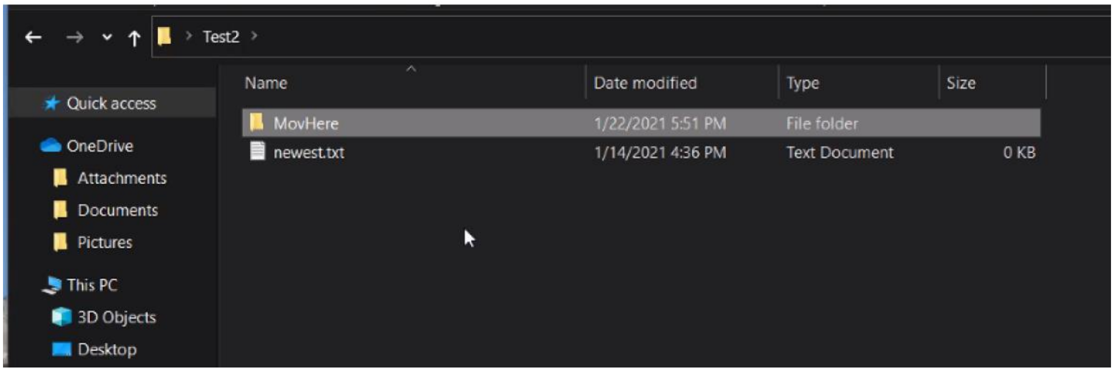
The second directory “LessThanThreshold” contains the file which has a size that is less than the given Threshold value in the Jason file “10KB” as was mentioned in the project.



Name	Date modified	Type	Size
Test11	1/22/2021 5:43 PM	File folder	
Gpa.txt	1/14/2021 12:11 AM	Text Document	0 KB
Newest.txt	1/13/2021 10:51 PM	Text Document	1 KB
Numbers.txt	1/13/2021 11:48 PM	Text Document	1 KB

Before (Mov):

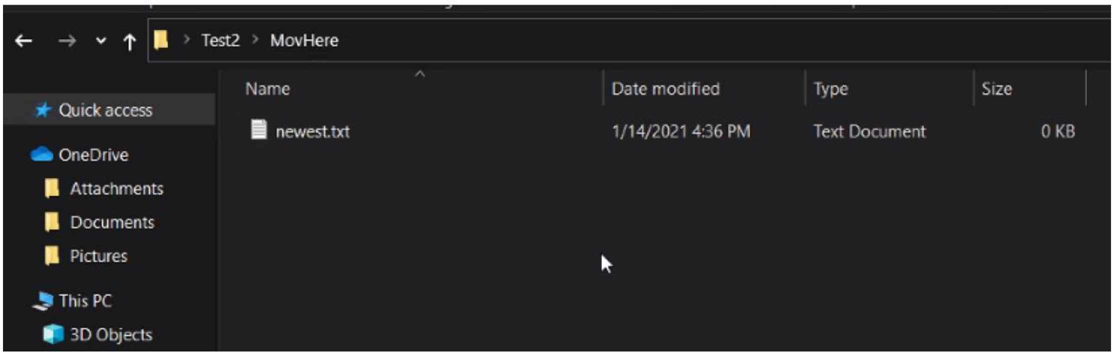
This command moves the newest file from a given path to a given path, As we can see the newest file (newest.txt) will be moved to the given directory (MovHere) as shown in Script.txt



Name	Date modified	Type	Size
MovHere	1/22/2021 5:51 PM	File folder	
newest.txt	1/14/2021 4:36 PM	Text Document	0 KB

After (Mov):

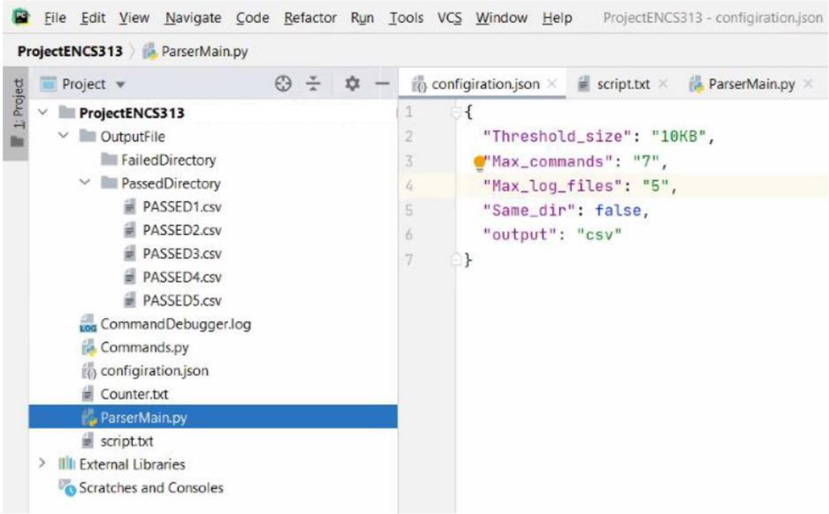
Moved successfully



Name	Date modified	Type	Size
newest.txt	1/14/2021 4:36 PM	Text Document	0 KB

(Max_log_files, Same_dir, Output) effect on each run

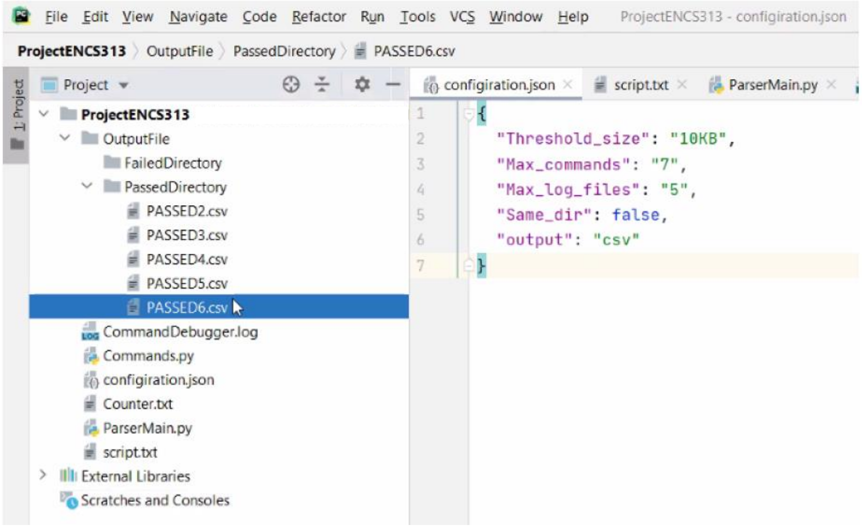
As we can see, max log files are 5, and “Same_dir” is false, and output type is csv, so we made 5 csv files as the figure shows and then if we make any more files, the oldest file must be deleted and replaced by the new one.



```json
{
    "Threshold_size": "10KB",
    "Max_commands": "7",
    "Max_log_files": "5",
    "Same_dir": false,
    "output": "csv"
}
```

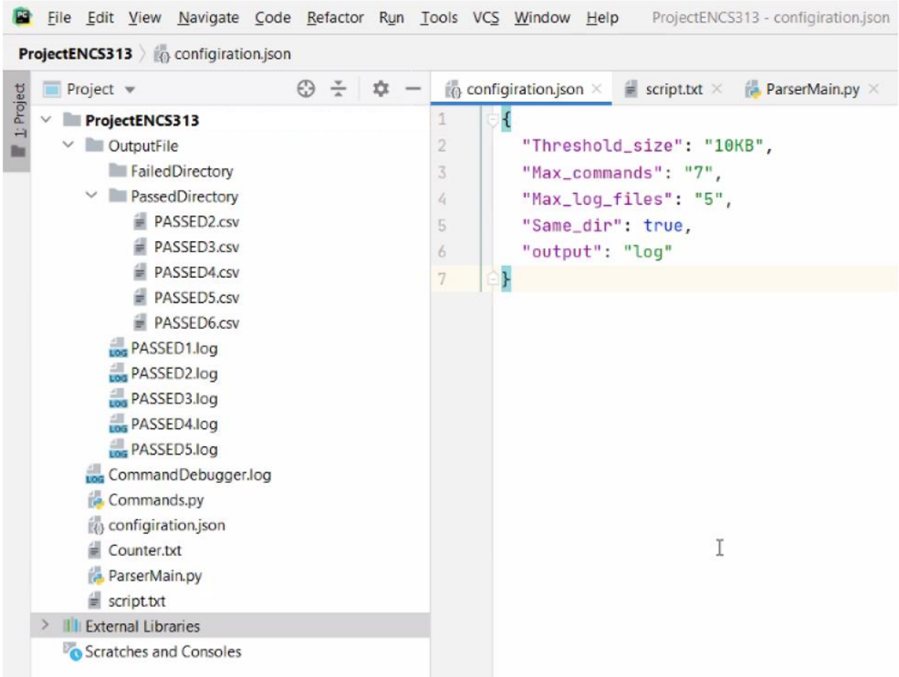
After creating a new file (PASSED6.csv), we can see that the oldest file (PASSED1.sv) was deleted.

Notice that files are put on the Passed directory since Same_dir is false.



```json
{
    "Threshold_size": "10KB",
    "Max_commands": "7",
    "Max_log_files": "5",
    "Same_dir": false,
    "output": "csv"
}
```

Notice that files are not on the Passed directory since Same_dir is true.



```
{
  "Threshold_size": "10KB",
  "Max_commands": "7",
  "Max_log_files": "5",
  "Same_dir": true,
  "output": "log"
}
```