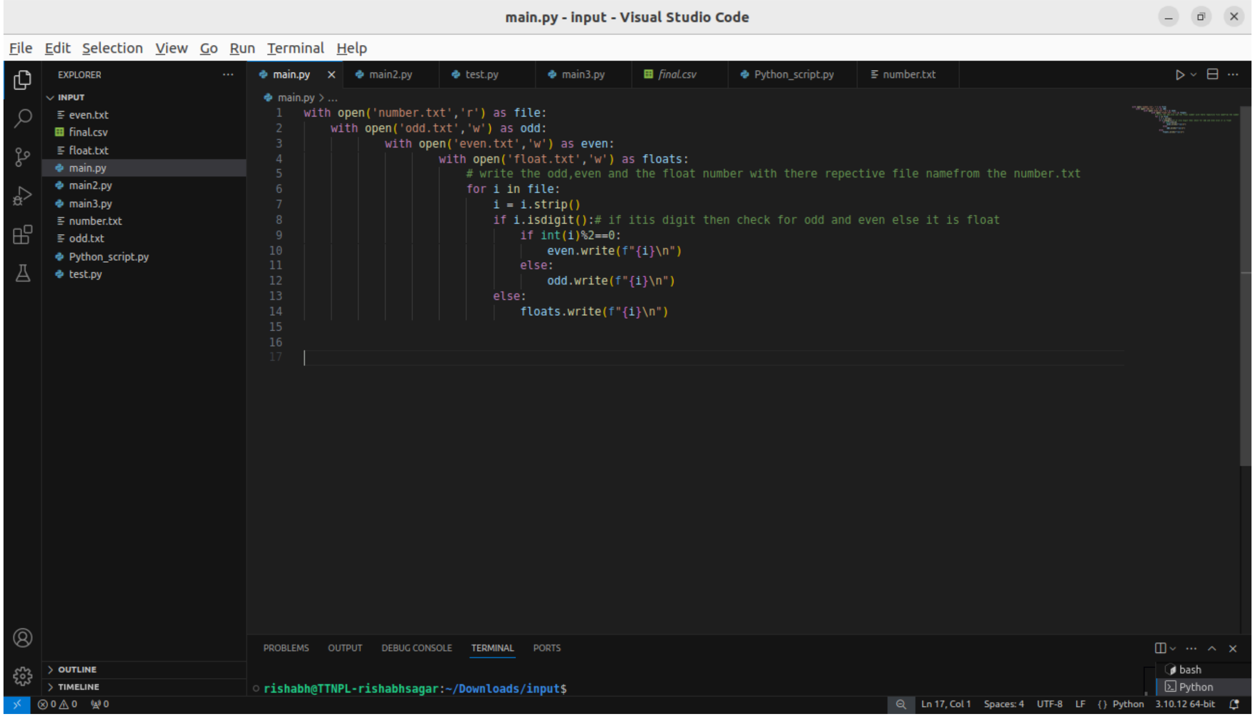
Input-Output and File Handling

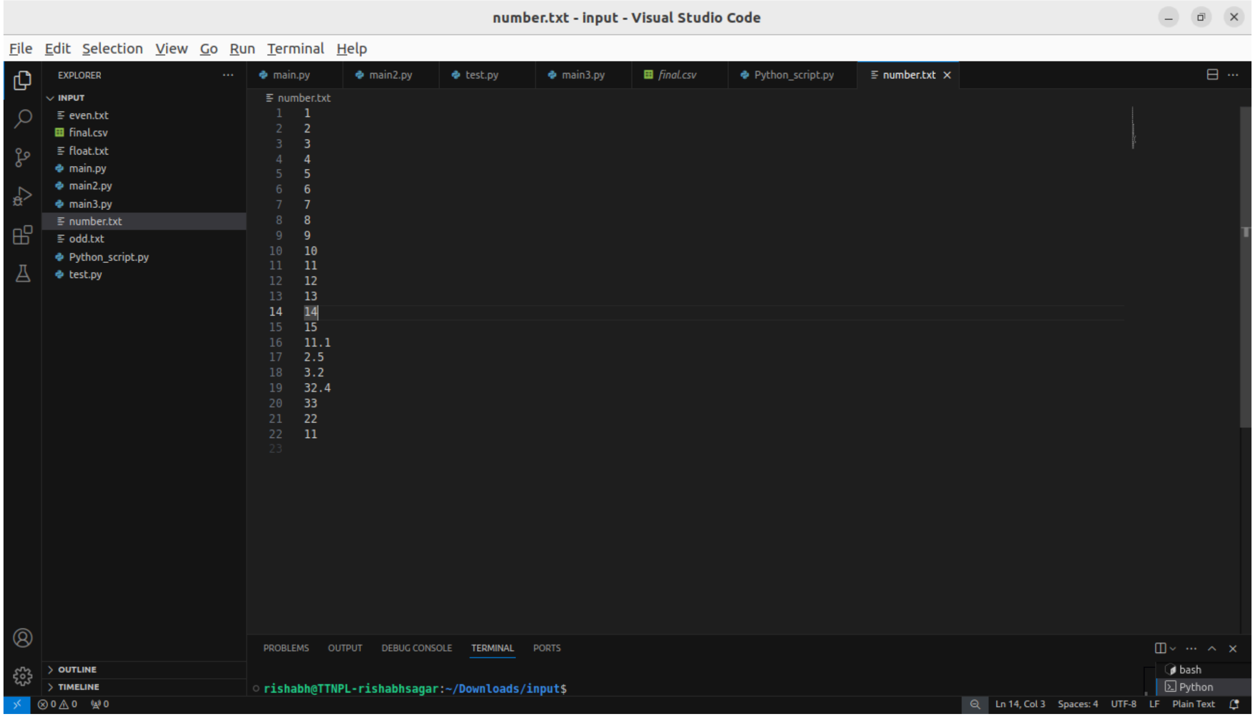
Q1)You have a number.txt, with each line a real number. Write a code to split this file into 3 files as follows: even.txt -- contain all even numbers odd.txt -- all odd number float.txt -- all floating point number Use with() clause for file handling

Ans.



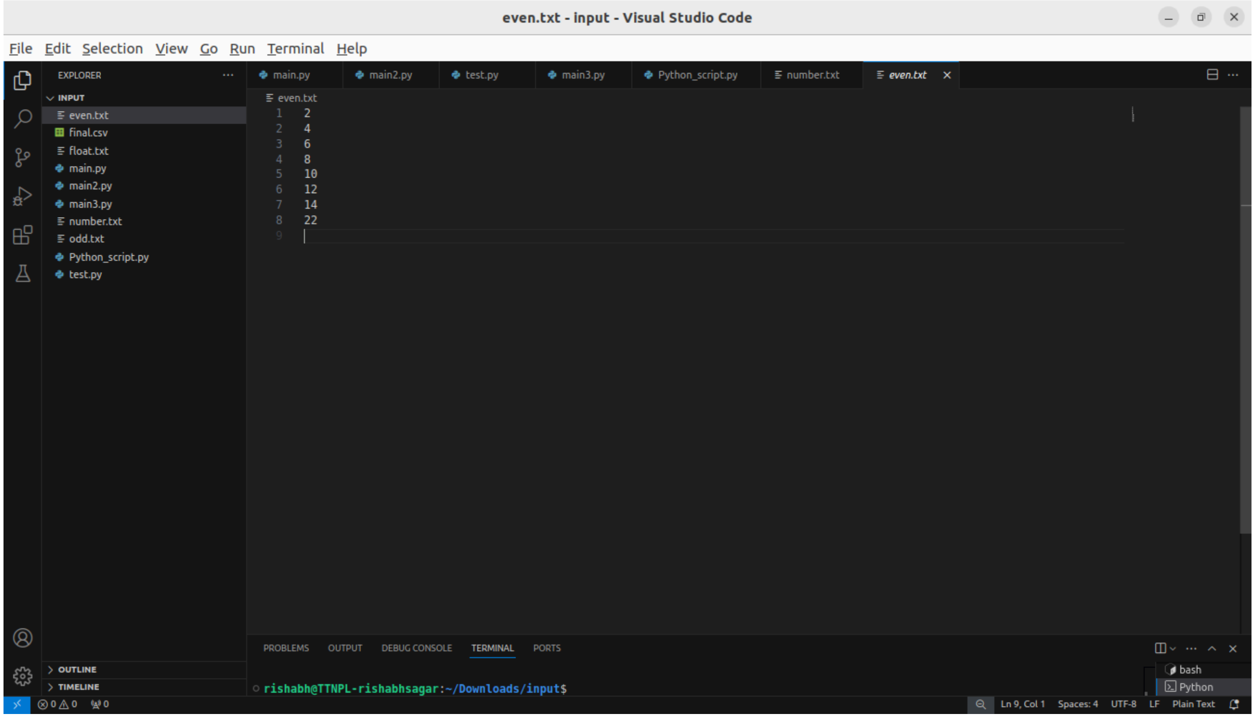
```python
with open('number.txt','r') as file:
    with open('odd.txt','w') as odd:
            with open('even.txt','w') as even:
                    with open('float.txt','w') as floats:
                        # write the odd,even and the float number with there repective file namefrom the number.txt
                        for i in file:
                            i = i.strip()
                            if i.isdigit():# if itis digit then check for odd and even else it is float
                                if int(i)%2==0:
                                    even.write(f"{i}\n")
                                else:
                                    odd.write(f"{i}\n")
                            else:
                                floats.write(f"{i}\n")
```

```
rishabh@TTNPL-rishabhsagar:~/Downloads/input$
```



```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
11.1
2.5
3.2
32.4
33
22
11
```

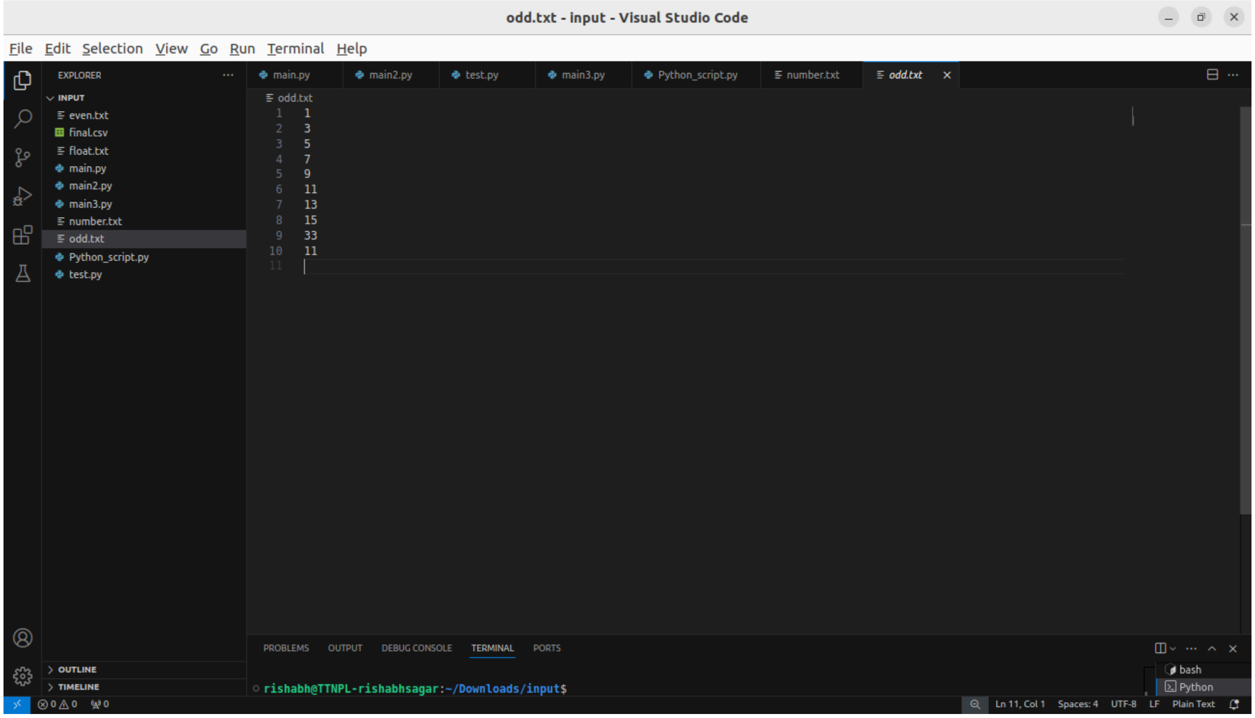
```
rishabh@TTNPL-rishabhsagar:~/Downloads/input$
```



even.txt

```
2
4
6
8
10
12
14
22

```

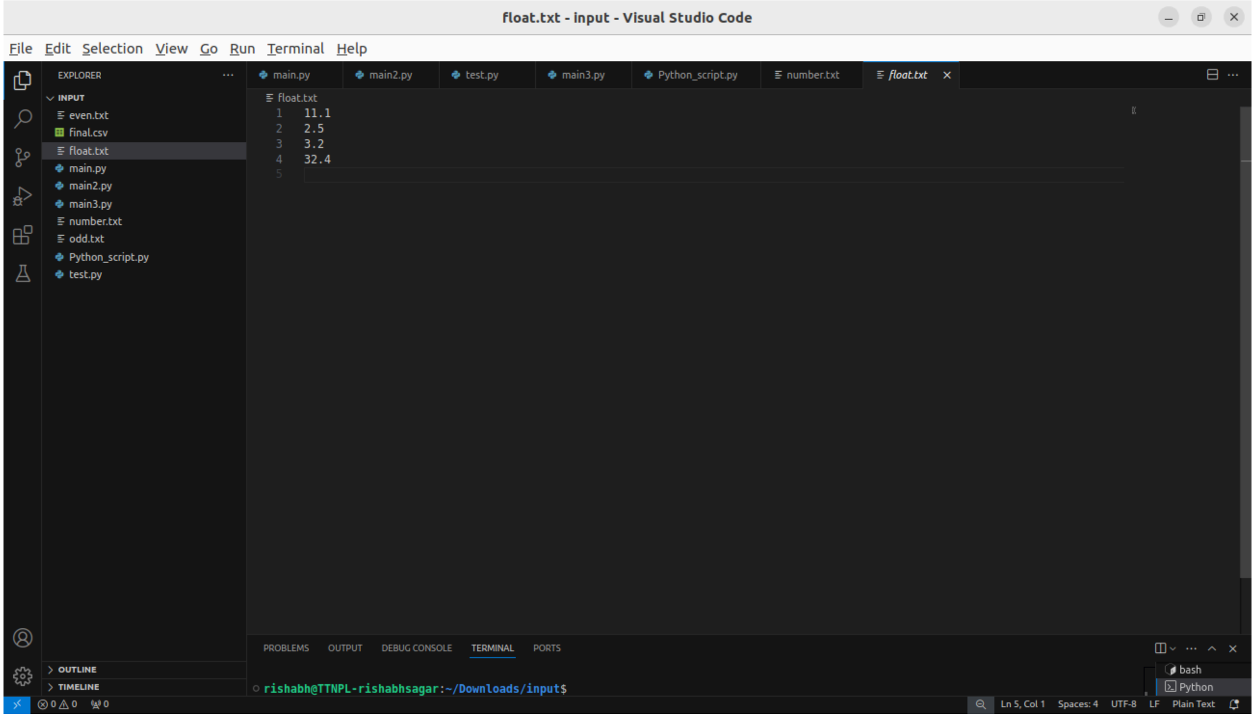
```
rishabh@TTNPL-rishabhsagar:~/Downloads/input$
```


odd.txt - input - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
INPUT
even.txt
final.csv
float.txt
main.py
main2.py
main3.py
number.txt
odd.txt
Python_script.py
test.py
main.py main2.py test.py main3.py Python_script.py number.txt odd.txt
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Python
OUTLINE
TIMELINE
Ln 11, Col 1 Spaces: 4 UTF-8 LF Plain Text

odd.txt

```
1
3
5
7
9
11
13
15
33
11

```

```
rishabh@TTNPL-rishabhsagar:~/Downloads/input$
```


float.txt - input - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
INPUT
even.txt
final.csv
float.txt
main.py
main2.py
main3.py
number.txt
odd.txt
Python_script.py
test.py
main.py main2.py test.py main3.py Python_script.py number.txt float.txt
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
Python
OUTLINE
TIMELINE
Ln 5, Col 1 Spaces: 4 UTF-8 LF Plain Text

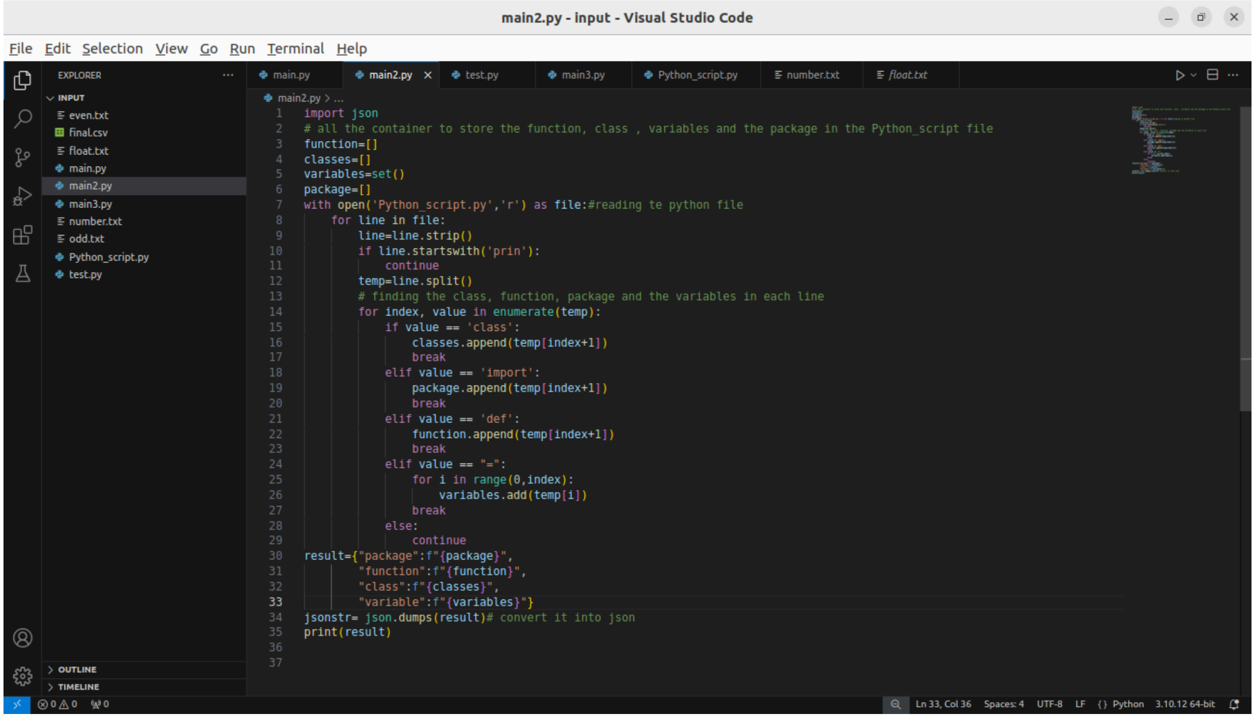
float.txt

```
11.1
2.5
3.2
32.4

```

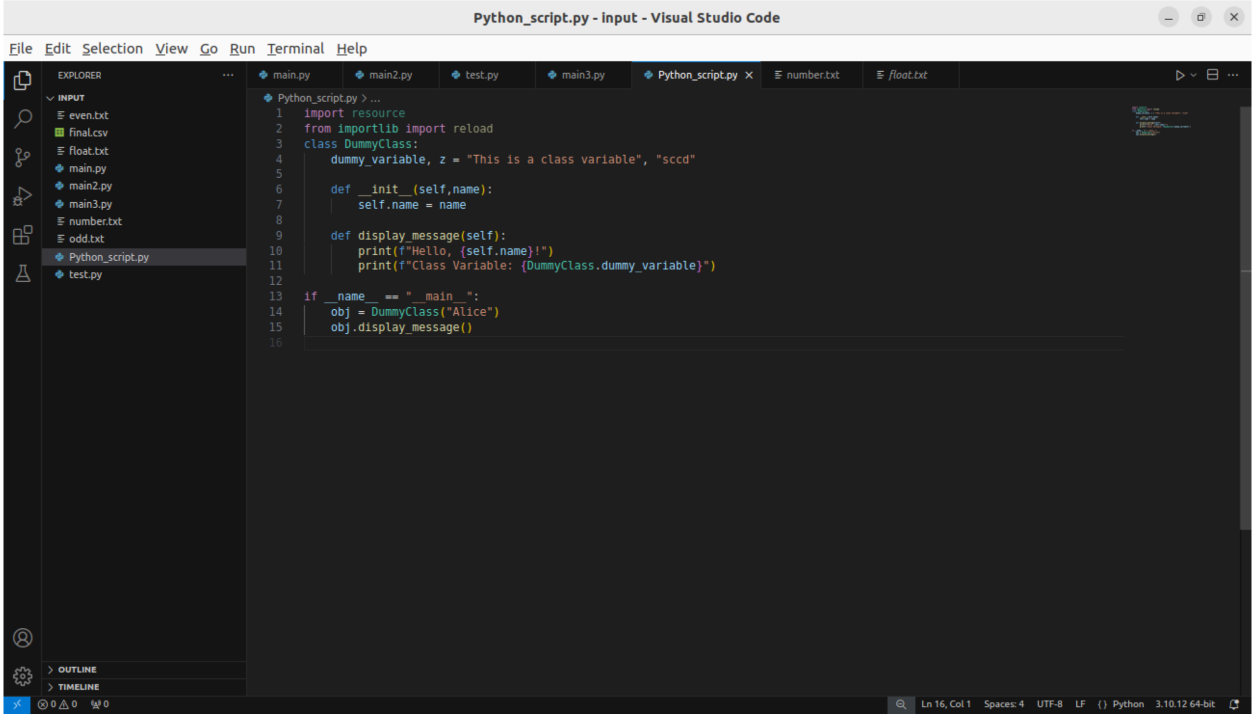
```
rishabh@TTNPL-rishabhsagar:~/Downloads/input$
```

Q2)Write a code to read a "Python_script.py" as input file and extract following information to prepare a JSON * all package name which the input Python script use * all function name which the input Python script define * all class name which the input Python script define * all the variable name which the input Python script define example output: { "package": ["os", "itertools"], "funation": ["function1", "function2"], "class": ["classA", "classB"], "variable": ["num", "i", "j"] }

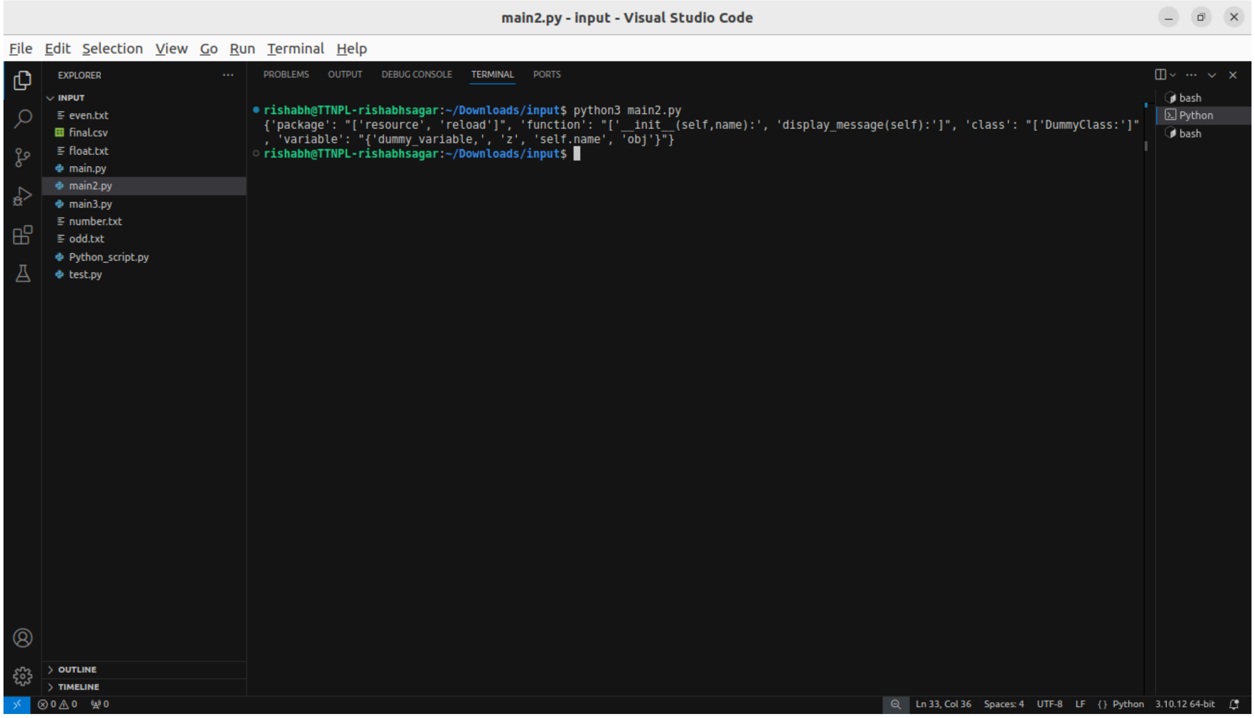
Ans.



```python
import json
# all the container to store the function, class , variables and the package in the Python_script file
function=[]
classes=[]
variables=set()
package=[]
with open('Python_script.py','r') as file:#reading te python file
    for line in file:
        line=line.strip()
        if line.startswith('prin'):
            continue
        temp=line.split()
        # finding the class, function, package and the variables in each line
        for index, value in enumerate(temp):
            if value == 'class':
                classes.append(temp[index+1])
                break
            elif value == 'import':
                package.append(temp[index+1])
                break
            elif value == 'def':
                function.append(temp[index+1])
                break
            elif value == "=":
                for i in range(0,index):
                    variables.add(temp[i])
                break
            else:
                continue
result={"package":f"{package}",
        "function":f"{function}",
        "class":f"{classes}",
        "variable":f"{variables}"}
jsonstr= json.dumps(result)# convert it into json
print(result)
```



```python
import resource
from importlib import reload
class DummyClass:
    dummy_variable, z = "This is a class variable", "sccd"

    def __init__(self,name):
        self.name = name

    def display_message(self):
        print(f"Hello, {self.name}!")
        print(f"Class Variable: {DummyClass.dummy_variable}")

if __name__ == "__main__":
    obj = DummyClass("Alice")
    obj.display_message()

```

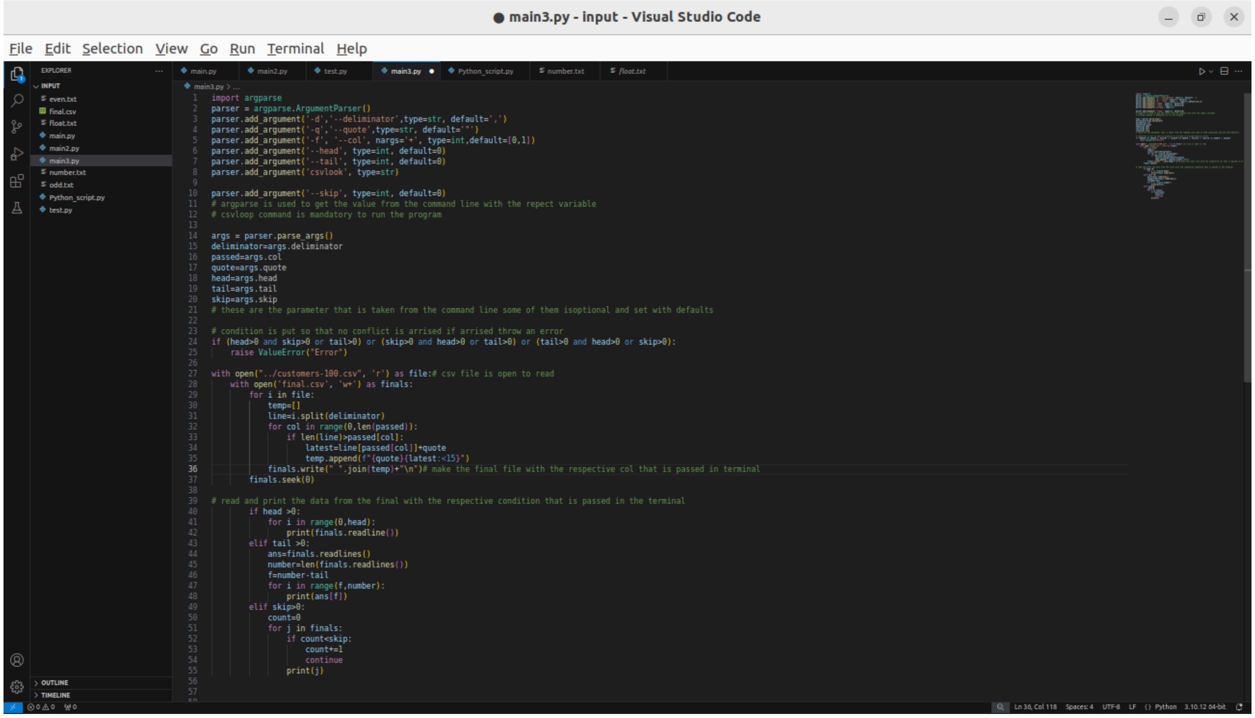


```
rishabh@TTNPL-rishabhsagar:~/Downloads/input$ python3 main2.py
{'package': "['resource', 'reload']", 'function': "['__init__(self,name):', 'display_message(self):']", 'class': "['DummyClass:']"
, 'variable': "{'dummy_variable,', 'z', 'self.name', 'obj'}"}
rishabh@TTNPL-rishabhsagar:~/Downloads/input$ 
```

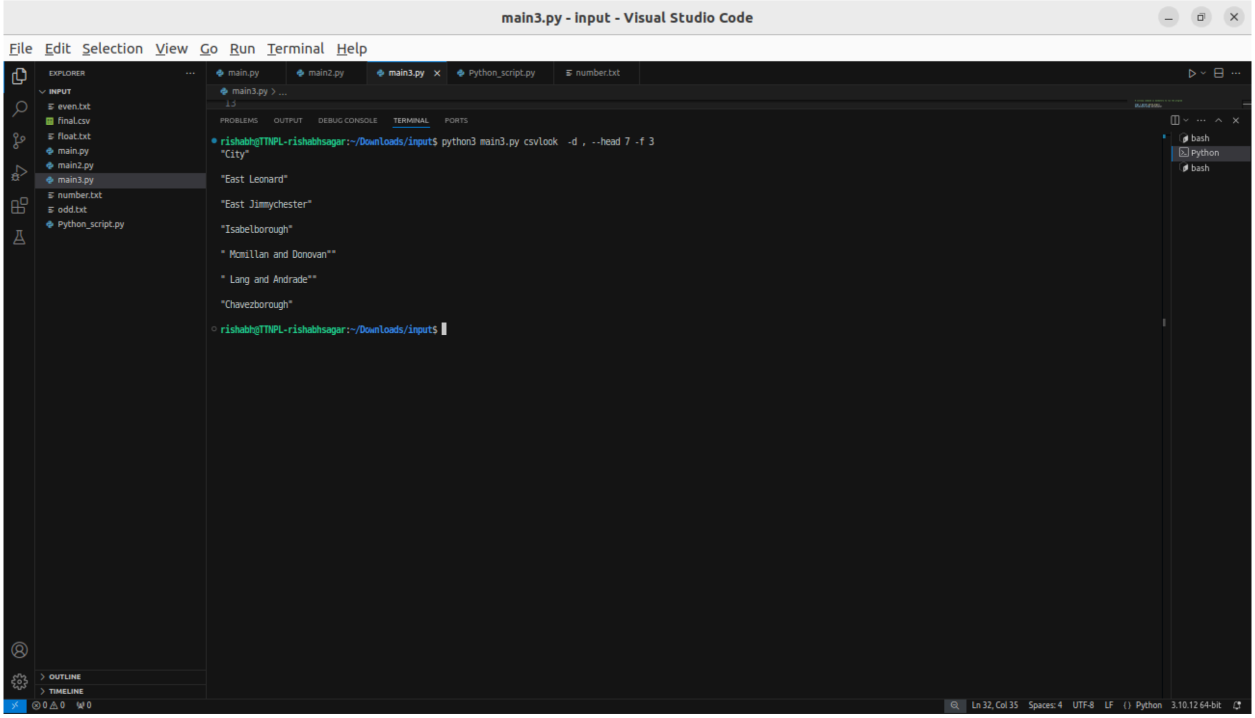
Q3)Without using Python CSV module write a "csvlook` command csvlook should have following features: * [-d DELIMITER] if -`d` option not paased script should be able to guess a separator * [-q QUOTECHAR] used to parsed colum value parenthesised within QUOTECHAR, if the value not passed should assume default value dboult quote `csvlook` should display data nicely on console in uniform width To project the data `csvlook` script should accept comma seprated colum numbers, e.g -f 3,5,7 should print

only column 3, 5 7 --skip-row N to skil first N rows --head N to display only first N rows --tail N to display last N rows

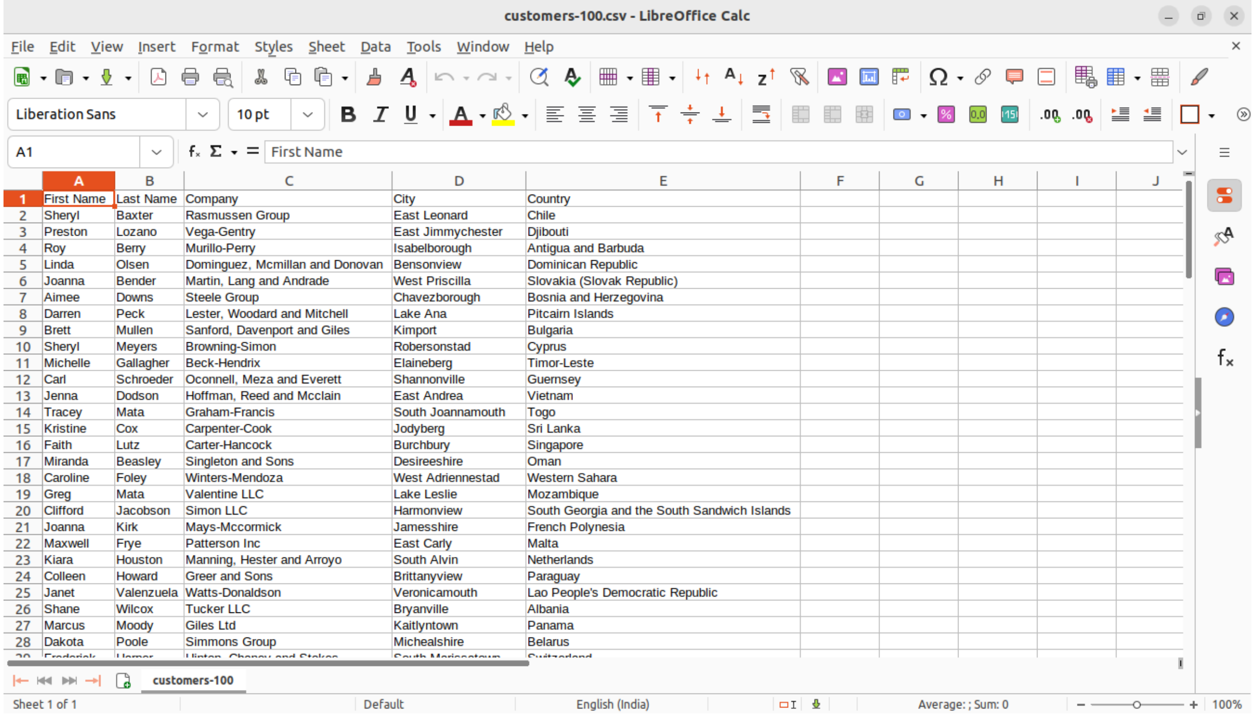
Ans.


● main3.py - input - Visual Studio Code
File Edit Selection View Go Run Terminal Help
main.py main2.py test.py main3.py Python_script.py number.txt float.txt
EXPLORER
INPUT
even.txt
final.csv
float.txt
main.py
main2.py
main3.py
number.txt
odd.txt
Python_script.py
test.py
main3.py > ...
OUTLINE
TIMELINE
Ln 36, Col 118 Spaces: 4 UTF-8 LF {} Python 3.10.12 64-bit

```python
import argparse
parser = argparse.ArgumentParser()
parser.add_argument('-d','--deliminator',type=str, default=',')
parser.add_argument('-q','--quote',type=str, default='"')
parser.add_argument('-f', '--col', nargs='+', type=int,default=[0,1])
parser.add_argument('--head', type=int, default=0)
parser.add_argument('--tail', type=int, default=0)
parser.add_argument('csvlook', type=str)

parser.add_argument('--skip', type=int, default=0)
# argparse is used to get the value from the command line with the repect variable
# csvloop command is mandatory to run the program

args = parser.parse_args()
deliminator=args.deliminator
passed=args.col
quote=args.quote
head=args.head
tail=args.tail
skip=args.skip
# these are the parameter that is taken from the command line some of them isoptional and set with defaults

# condition is put so that no conflict is arrised if arrised throw an error
if (head>0 and skip>0 or tail>0) or (skip>0 and head>0 or tail>0) or (tail>0 and head>0 or skip>0):
    raise ValueError("Error")

with open("../customers-100.csv", 'r') as file:# csv file is open to read
    with open('final.csv', 'w+') as finals:
        for i in file:
            temp=[]
            line=i.split(deliminator)
            for col in range(0,len(passed)):
                if len(line)>passed[col]:
                    latest=line[passed[col]]+quote
                    temp.append(f"{quote}{latest:<15}")
            finals.write(" ".join(temp)+"\n")# make the final file with the respective col that is passed in terminal
        finals.seek(0)

# read and print the data from the final with the respective condition that is passed in the terminal
        if head >0:
            for i in range(0,head):
                print(finals.readline())
        elif tail >0:
            ans=finals.readlines()
            number=len(finals.readlines())
            f=number-tail
            for i in range(f,number):
                print(ans[f])
        elif skip>0:
            count=0
            for j in finals:
                if count<skip:
                    count+=1
                    continue
                print(j)
```



```
rishabh@TTNPL-rishabhsagar:~/Downloads/input$ python3 main3.py csvlook  -d , --head 7 -f 3
"City"

"East Leonard"

"East Jimmychester"

"Isabelborough"

" Mcmillan and Donovan""

" Lang and Andrade""

"Chavezborough"

rishabh@TTNPL-rishabhsagar:~/Downloads/input$
```


customers-100.csv - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
Liberation Sans
10 pt
A1
First Name
customers-100
Sheet 1 of 1
Default
English (India)
Average: ; Sum: 0
100%

	A	B	C	D	E
1	First Name	Last Name	Company	City	Country
2	Sheryl	Baxter	Rasmussen Group	East Leonard	Chile
3	Preston	Lozano	Vega-Gentry	East Jimmychester	Djibouti
4	Roy	Berry	Murillo-Perry	Isabelborough	Antigua and Barbuda
5	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonview	Dominican Republic
6	Joanna	Bender	Martin, Lang and Andrade	West Priscilla	Slovakia (Slovak Republic)
7	Aimee	Downs	Steele Group	Chavezborough	Bosnia and Herzegovina
8	Darren	Peck	Lester, Woodard and Mitchell	Lake Ana	Pitcairn Islands
9	Brett	Mullen	Sanford, Davenport and Giles	Kimport	Bulgaria
10	Sheryl	Meyers	Browning-Simon	Robersonstad	Cyprus
11	Michelle	Gallagher	Beck-Hendrix	Elaineberg	Timor-Leste
12	Carl	Schroeder	Oconnell, Meza and Everett	Shannonville	Guernsey
13	Jenna	Dodson	Hoffman, Reed and Mcclain	East Andrea	Vietnam
14	Tracey	Mata	Graham-Francis	South Joannamouth	Togo
15	Kristine	Cox	Carpenter-Cook	Jodyberg	Sri Lanka
16	Faith	Lutz	Carter-Hancock	Burchbury	Singapore
17	Miranda	Beasley	Singleton and Sons	Desireeshire	Oman
18	Caroline	Foley	Winters-Mendoza	West Adriennestad	Western Sahara
19	Greg	Mata	Valentine LLC	Lake Leslie	Mozambique
20	Clifford	Jacobson	Simon LLC	Harmonview	South Georgia and the South Sandwich Islands
21	Joanna	Kirk	Mays-Mccormick	Jamesshire	French Polynesia
22	Maxwell	Frye	Patterson Inc	East Carly	Malta
23	Kiara	Houston	Manning, Hester and Arroyo	South Alvin	Netherlands
24	Colleen	Howard	Greer and Sons	Brittanyview	Paraguay
25	Janet	Valenzuela	Watts-Donaldson	Veronicamouth	Lao People's Democratic Republic
26	Shane	Wilcox	Tucker LLC	Bryanville	Albania
27	Marcus	Moody	Giles Ltd	Kaitlyntown	Panama
28	Dakota	Poole	Simmons Group	Michealshire	Belarus