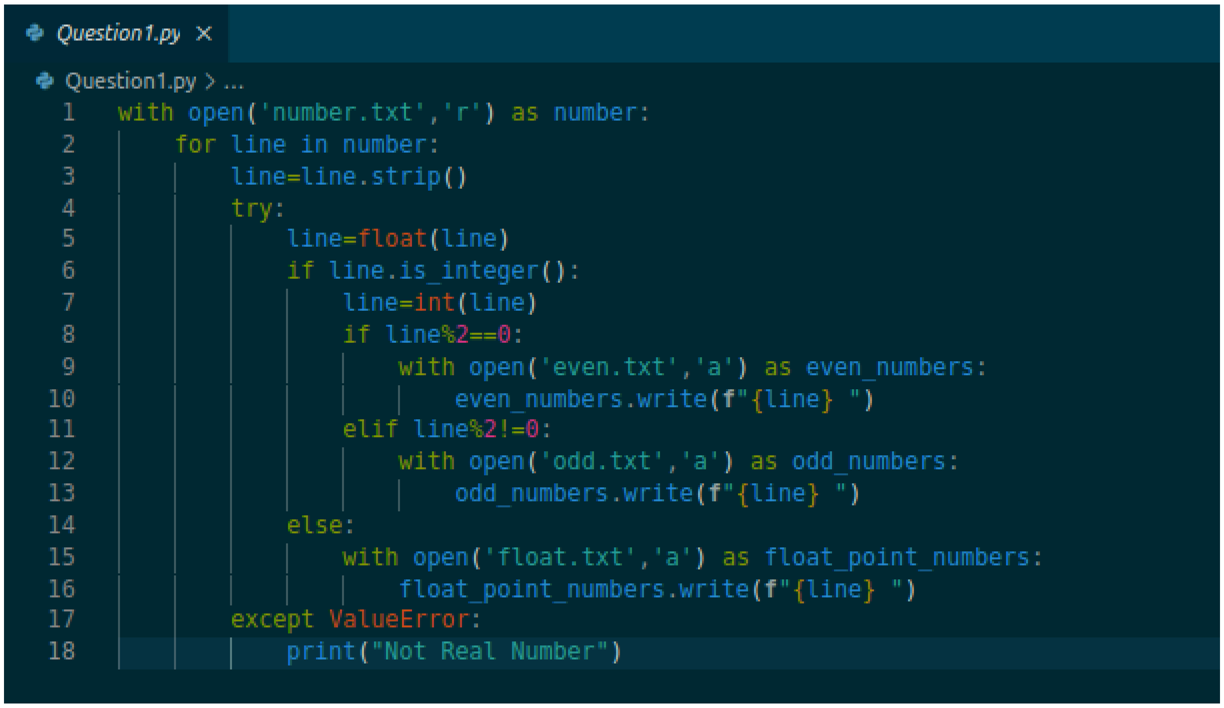
Input-Output and File Handling

1) 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 handlong

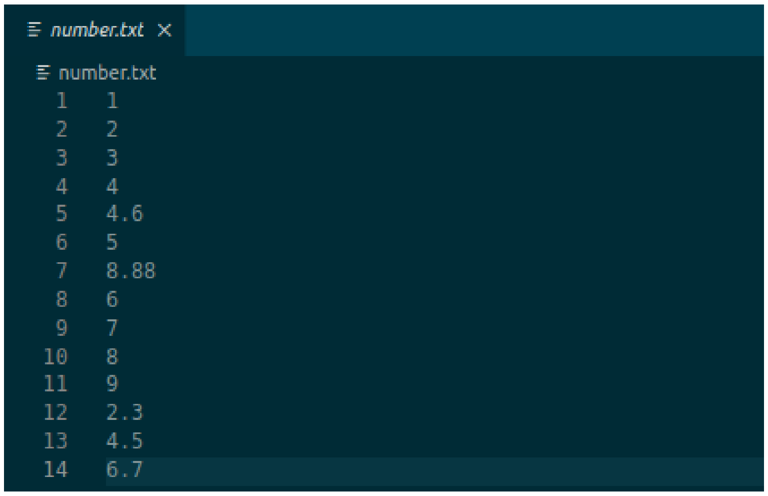


Question1.py

Question1.py > ...

```python
with open('number.txt','r') as number:
    for line in number:
        line=line.strip()
        try:
            line=float(line)
            if line.is_integer():
                line=int(line)
                if line%2==0:
                    with open('even.txt','a') as even_numbers:
                        even_numbers.write(f"{line} ")
                elif line%2!=0:
                    with open('odd.txt','a') as odd_numbers:
                        odd_numbers.write(f"{line} ")
            else:
                with open('float.txt','a') as float_point_numbers:
                    float_point_numbers.write(f"{line} ")
        except ValueError:
            print("Not Real Number")
```

Input File : number.txt



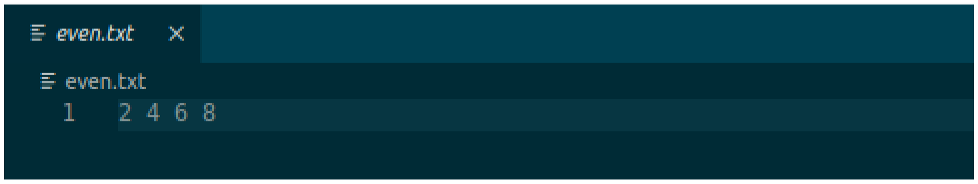
number.txt

number.txt

```
1
2
3
4
4.6
5
8.88
6
7
8
9
2.3
4.5
6.7
```

OUTPUT :

1. **Content of even.txt file**

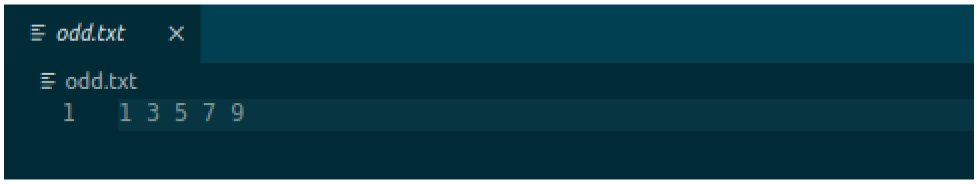


even.txt

even.txt

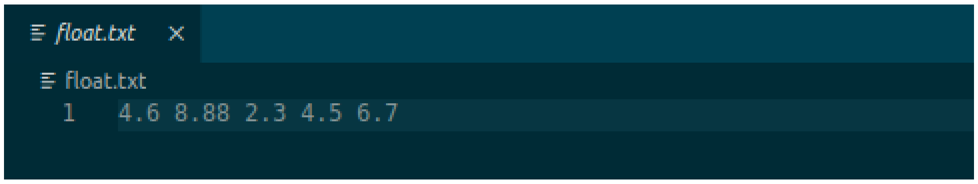
```
2 4 6 8
```

2. **Content of odd.txt file**



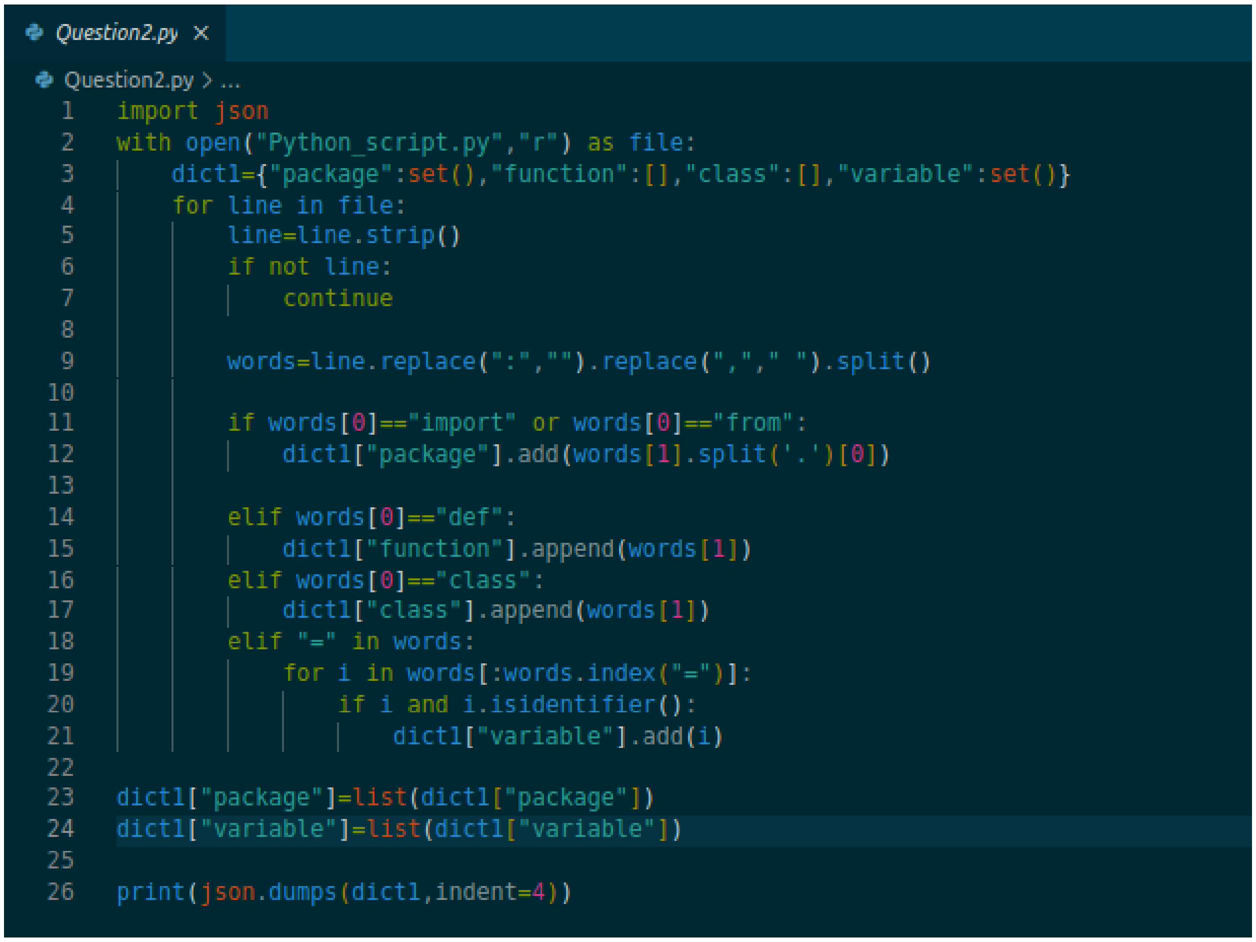
```
1 3 5 7 9
```

3. **Content of float.txt file**



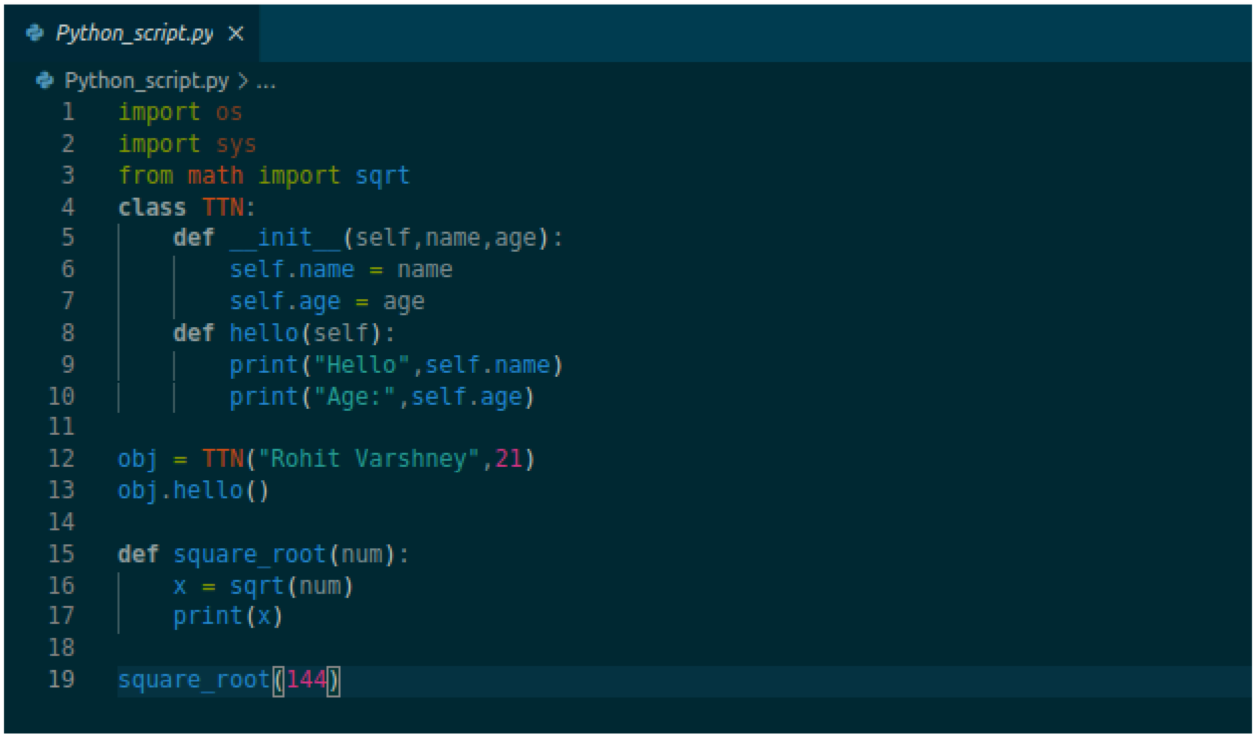
```
4.6 8.88 2.3 4.5 6.7
```

2) Write a code to read a "Python_script.py" as inpit 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"] }



```python
import json
with open("Python_script.py","r") as file:
    dict1={"package":set(),"function":[],"class":[],"variable":set()}
    for line in file:
        line=line.strip()
        if not line:
            continue

        words=line.replace(":","").replace(","," ").split()

        if words[0]=="import" or words[0]=="from":
            dict1["package"].add(words[1].split('.')[0])

        elif words[0]=="def":
            dict1["function"].append(words[1])
        elif words[0]=="class":
            dict1["class"].append(words[1])
        elif "=" in words:
            for i in words[:words.index("=")]:
                if i and i.isidentifier():
                    dict1["variable"].add(i)

dict1["package"]=list(dict1["package"])
dict1["variable"]=list(dict1["variable"])

print(json.dumps(dict1,indent=4))
```

Input File : Python_script.py

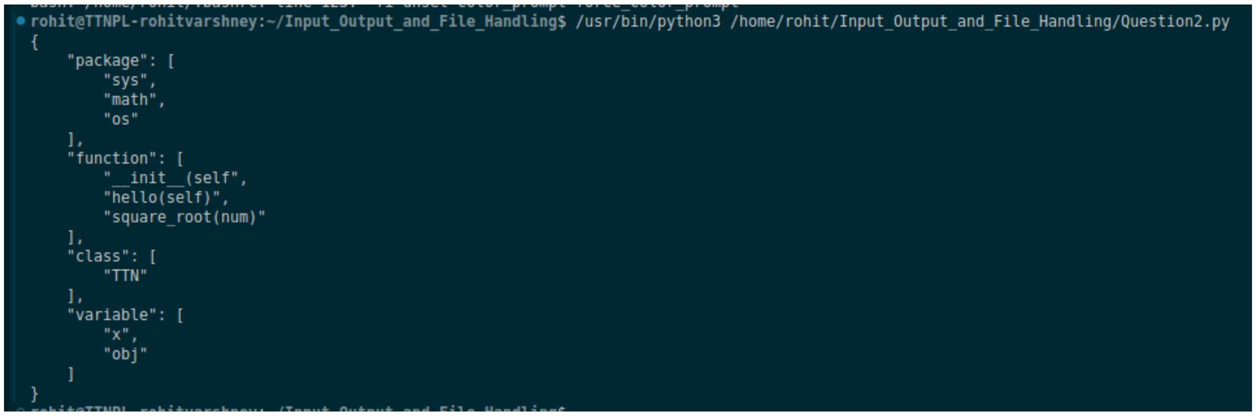


Python_script.py

Python_script.py > ...

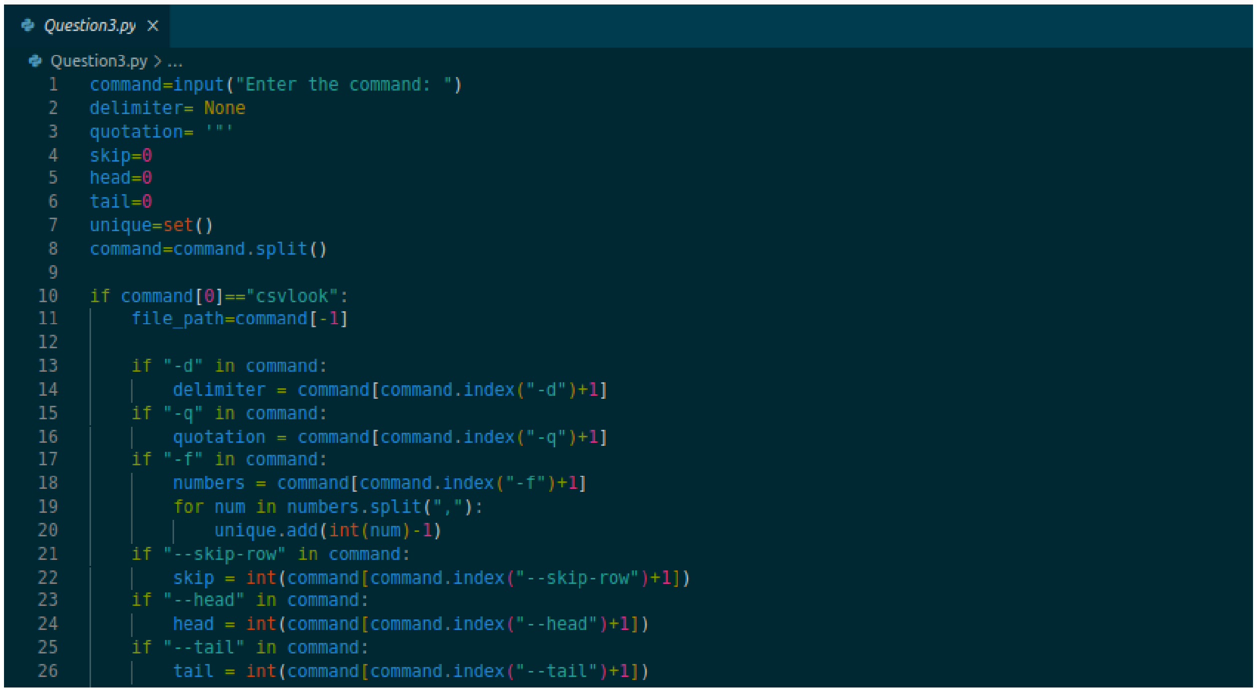
```python
import os
import sys
from math import sqrt
class TTN:
    def __init__(self,name,age):
        self.name = name
        self.age = age
    def hello(self):
        print("Hello",self.name)
        print("Age:",self.age)

obj = TTN("Rohit Varshney",21)
obj.hello()

def square_root(num):
    x = sqrt(num)
    print(x)

square_root(144)
```

OUTPUT :

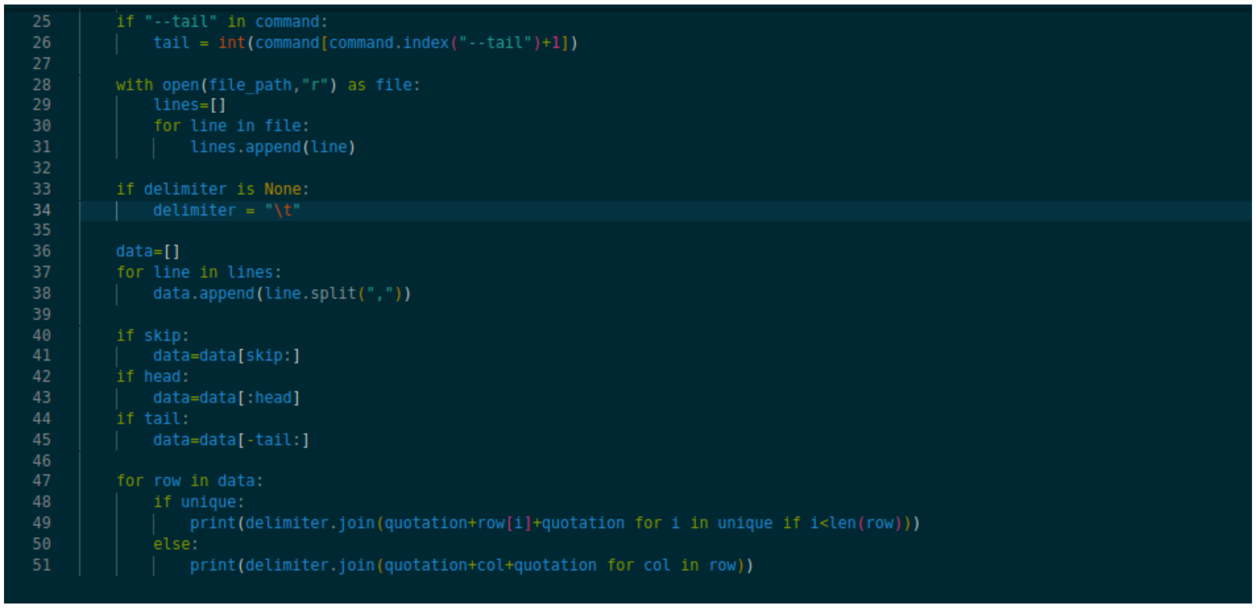


```
rohit@TTNPL-rohitvarshney:~/Input_Output_and_File_Handling$ /usr/bin/python3 /home/rohit/Input_Output_and_File_Handling/Question2.py
{
    "package": [
        "sys",
        "math",
        "os"
    ],
    "function": [
        "__init__(self",
        "hello(self)",
        "square_root(num)"
    ],
    "class": [
        "TTN"
    ],
    "variable": [
        "x",
        "obj"
    ]
}
```

3) 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 seperator * [-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

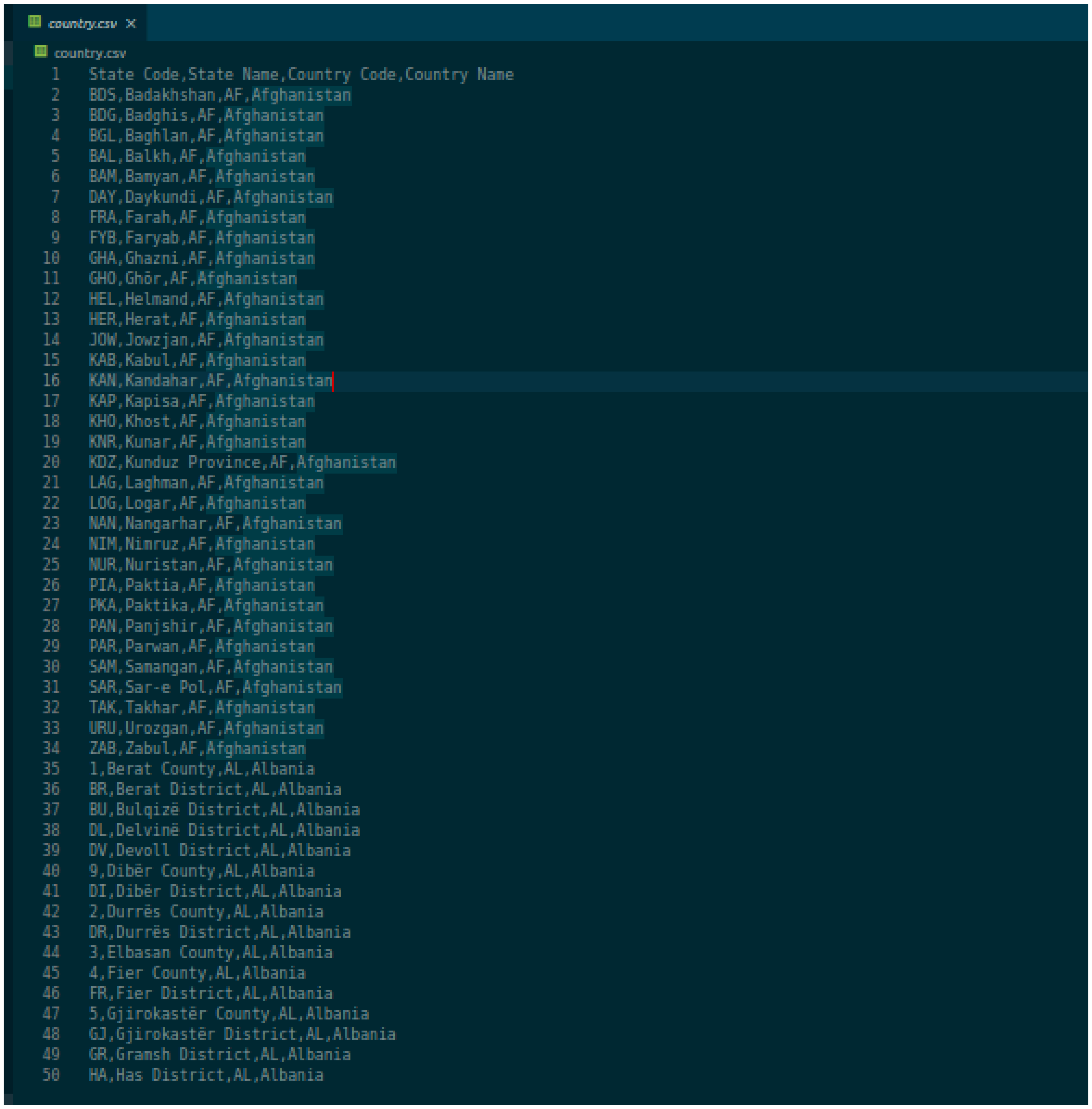


```python
command=input("Enter the command: ")
delimiter= None
quotation= '"'
skip=0
head=0
tail=0
unique=set()
command=command.split()

if command[0]=="csvlook":
    file_path=command[-1]

    if "-d" in command:
        delimiter = command[command.index("-d")+1]
    if "-q" in command:
        quotation = command[command.index("-q")+1]
    if "-f" in command:
        numbers = command[command.index("-f")+1]
        for num in numbers.split(","):
            unique.add(int(num)-1)
    if "--skip-row" in command:
        skip = int(command[command.index("--skip-row")+1])
    if "--head" in command:
        head = int(command[command.index("--head")+1])
    if "--tail" in command:
        tail = int(command[command.index("--tail")+1])
```



```python
    if "--tail" in command:
        tail = int(command[command.index("--tail")+1])

    with open(file_path,"r") as file:
        lines=[]
        for line in file:
            lines.append(line)

    if delimiter is None:
        delimiter = "\t"

    data=[]
    for line in lines:
        data.append(line.split(","))

    if skip:
        data=data[skip:]
    if head:
        data=data[:head]
    if tail:
        data=data[-tail:]

    for row in data:
        if unique:
            print(delimiter.join(quotation+row[i]+quotation for i in unique if i<len(row)))
        else:
            print(delimiter.join(quotation+col+quotation for col in row))
```

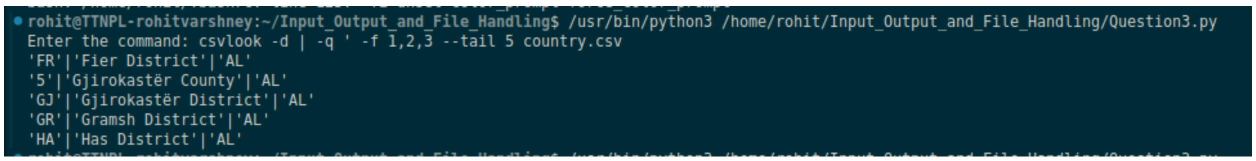
Csv file : country.csv



country.csv

```
State Code,State Name,Country Code,Country Name
BDS,Badakhshan,AF,Afghanistan
BDG,Badghis,AF,Afghanistan
BGL,Baghlan,AF,Afghanistan
BAL,Balkh,AF,Afghanistan
BAM,Bamyan,AF,Afghanistan
DAY,Daykundi,AF,Afghanistan
FRA,Farah,AF,Afghanistan
FYB,Faryab,AF,Afghanistan
GHA,Ghazni,AF,Afghanistan
GHO,Ghōr,AF,Afghanistan
HEL,Helmand,AF,Afghanistan
HER,Herat,AF,Afghanistan
JOW,Jowzjan,AF,Afghanistan
KAB,Kabul,AF,Afghanistan
KAN,Kandahar,AF,Afghanistan
KAP,Kapisa,AF,Afghanistan
KHO,Khost,AF,Afghanistan
KNR,Kunar,AF,Afghanistan
KDZ,Kunduz Province,AF,Afghanistan
LAG,Laghman,AF,Afghanistan
LOG,Logar,AF,Afghanistan
NAN,Nangarhar,AF,Afghanistan
NIM,Nimruz,AF,Afghanistan
NUR,Nuristan,AF,Afghanistan
PIA,Paktia,AF,Afghanistan
PKA,Paktika,AF,Afghanistan
PAN,Panjshir,AF,Afghanistan
PAR,Parwan,AF,Afghanistan
SAM,Samangan,AF,Afghanistan
SAR,Sar-e Pol,AF,Afghanistan
TAK,Takhar,AF,Afghanistan
URU,Urozgan,AF,Afghanistan
ZAB,Zabul,AF,Afghanistan
1,Berat County,AL,Albania
BR,Berat District,AL,Albania
BU,Bulqizë District,AL,Albania
DL,Delvinë District,AL,Albania
DV,Devoll District,AL,Albania
9,Dibër County,AL,Albania
DI,Dibër District,AL,Albania
2,Durrës County,AL,Albania
DR,Durrës District,AL,Albania
3,Elbasan County,AL,Albania
4,Fier County,AL,Albania
FR,Fier District,AL,Albania
5,Gjirokastër County,AL,Albania
GJ,Gjirokastër District,AL,Albania
GR,Gramsh District,AL,Albania
HA,Has District,AL,Albania
```

OUTPUT :



```
rohit@TTNPL-rohitvarshney:~/Input_Output_and_File_Handling$ /usr/bin/python3 /home/rohit/Input_Output_and_File_Handling/Question3.py
Enter the command: csvlook -d | -q ' -f 1,2,3 --tail 5 country.csv
'FR'|'Fier District'|'AL'
'5'|'Gjirokastër County'|'AL'
'GJ'|'Gjirokastër District'|'AL'
'GR'|'Gramsh District'|'AL'
'HA'|'Has District'|'AL'
```