1. Write a Python program to reverse a number and also find the sum of digits of the number. Prompt the user for input.

```
Number = int(input("Enter any Number: "))
Reverse = 0
Sum=0
while(Number > 0):
   Reminder = Number %10
   Reverse = (Reverse *10) + Reminder
   Sum+=Reminder
   Number = Number //10

print("\n Reverse of entered number is =",Reverse)
print("\n Sum of digits is =",Sum)
```

11. A) Write a Python code to check whether a given year is a leap year or not [An year is a leap year if it’s divisible by 4 but not divisible by 100 except for those divisible by 400]**.**

```
year = int(input("Enter a year: "))

# divided by 100 means century year (ending with 00)
# century year divided by 400 is leap year
if (year % 400 == 0) and (year % 100 == 0):
   print(year, " is a leap year")

# not divided by 100 means not a century year
# year divided by 4 is a leap year
elif (year % 4 ==0) and (year % 100 != 0):
   print("year, " is a leap year"))

# if not divided by both 400 (century year) and 4 (not century year)
# year is not leap year
else:
   print(year, " is not a leap year")
```

11. B) Write a Python program to print the value of ***2^{2n}+n+5*** for ***n*** provided by the user.

```
n=int(input("Enter a number"))
val=2**(2*n)+n+5
print("Result of 2**(2*n)+n+5 is ", val)
```

12. A) Write a Python program to find the value for *sin(x)* up to *n* terms using the series where ***x*** is in degrees

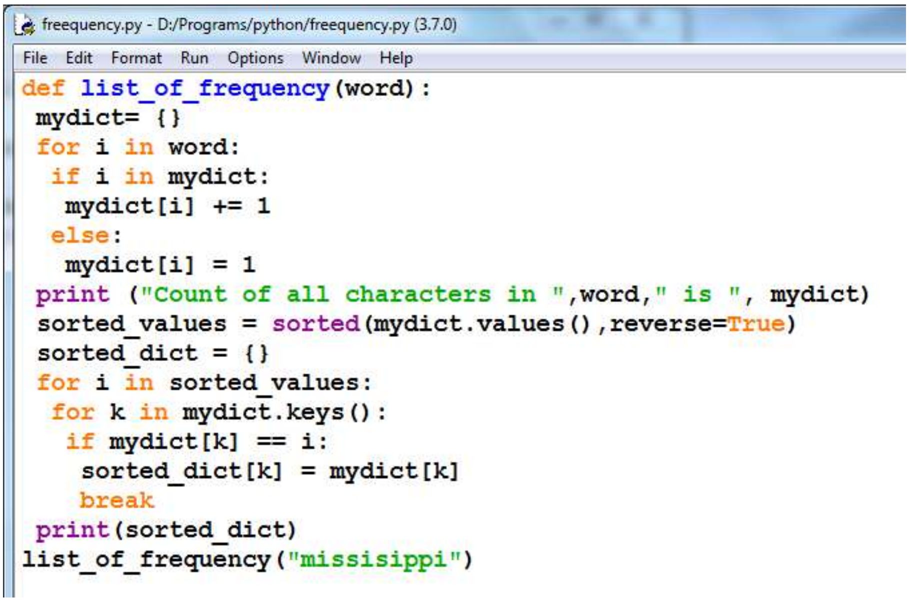
$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$

```
# importing math module
import math
# function which returns sum of sine series


def sumsine(degrees, terms):
 # taking a variable which stores sum of sine series
  sumSeries = 0
  for i in range(terms):
   # getting sign
    signofNum = (-1)**i
    # pie value
    pievalue = 22/7
    # degree value of given number
    degval = degrees*(pievalue/180)
    sumSeries = sumSeries + ((degval**(2.0*i+1)) /math.factorial(2*i+1))*signofNum
   # returning the sum of sine series
  return sumSeries


degrees = int(input("enter the number of degrees = "))
terms = int(input("enter number of terms = "))
print("The sum of sine series of ", degrees, "degrees", "of",  terms, "terms =", round(sumsine(degrees, terms), 2))
```

12. B) Write a Python code to determine whether the given string is a Palindrome or not using slicing. Do not use any string function.

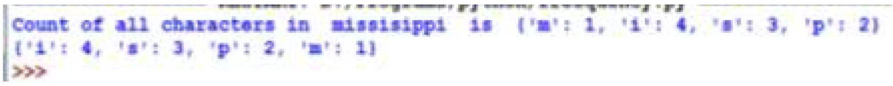
```
# function to check string is palindrome or not

def isPalindrome(str):

        # Run loop from 0 to len/2
        for i in range(0, int(len(str)/2)):
                if str[i] != str[len(str)-i-1]:
                        return False
        return True

s =input("Enter a string")
ans = isPalindrome(s)

if (ans):
        print("Yes. String is palindrome")
else:
        print("No. String is not palindrome")
```

13. A) Write a Python code to create a function called *list_of_frequency* that takes a string and prints the letters in non-increasing order of the frequency of their occurrences. Use dictionaries.


freequency.py - D:/Programs/python/freequency.py (3.7.0)
File Edit Format Run Options Window Help

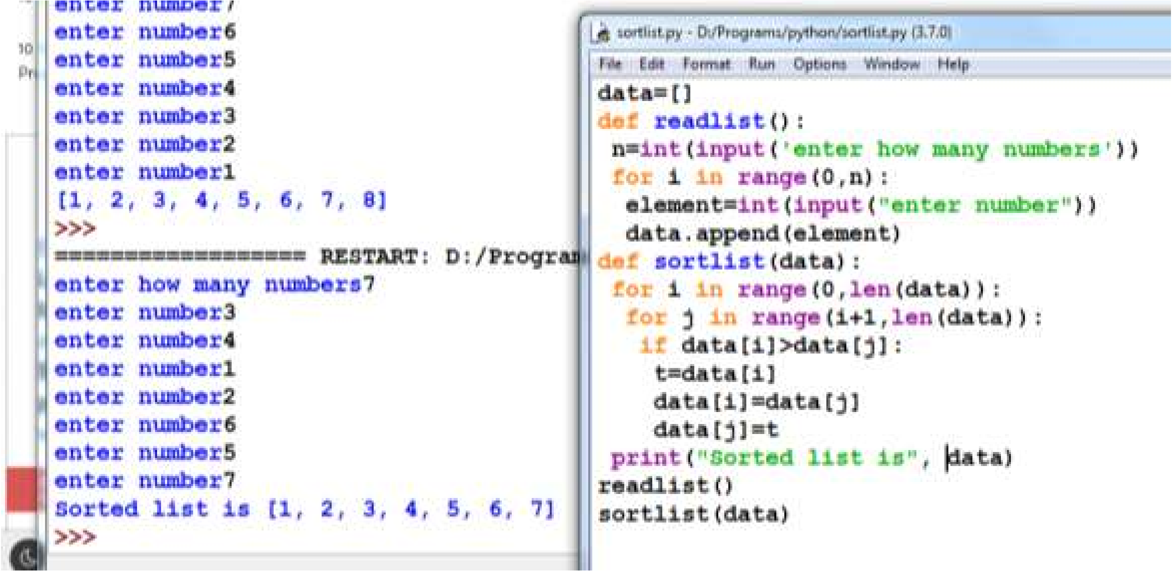
```
def list_of_frequency(word):
 mydict= {}
 for i in word:
  if i in mydict:
   mydict[i] += 1
  else:
   mydict[i] = 1
 print ("Count of all characters in ",word," is ", mydict)
 sorted_values = sorted(mydict.values(),reverse=True)
 sorted_dict = {}
 for i in sorted_values:
  for k in mydict.keys():
   if mydict[k] == i:
    sorted_dict[k] = mydict[k]
    break
 print(sorted_dict)
list_of_frequency("missisippi")
```

Output



```
Count of all characters in  missisippi  is  {'m': 1, 'i': 4, 's': 3, 'p': 2}
{'i': 4, 's': 3, 'p': 2, 'm': 1}
>>>
```

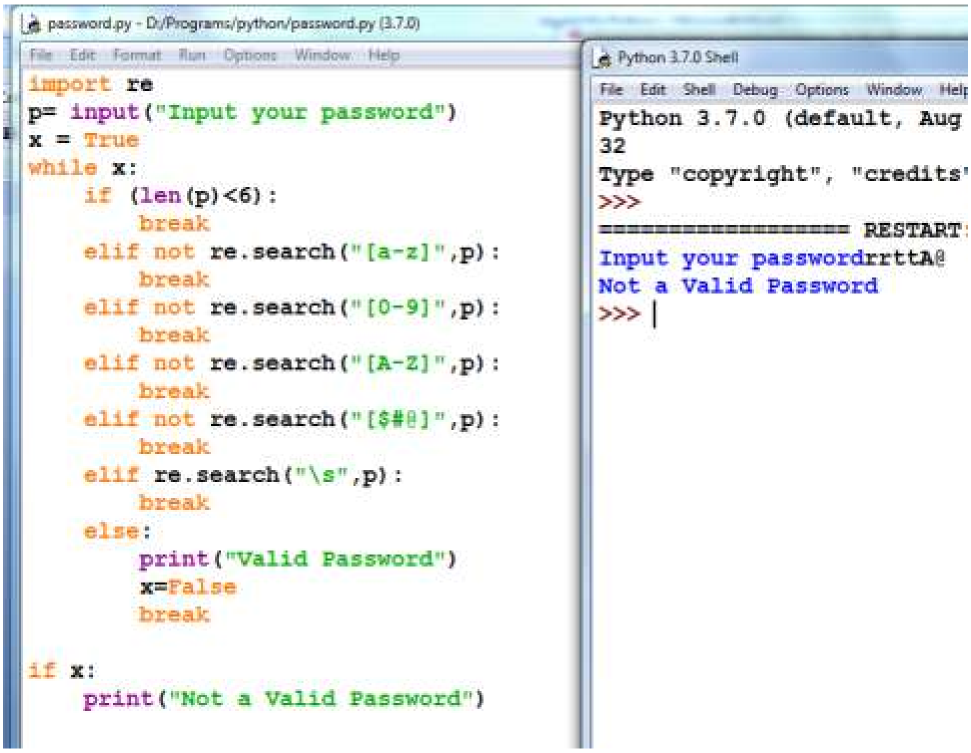
13 B) Write a Python program to read a list of numbers and sort the list in a non-decreasing order without using any built in functions. Separate function should be written to sort the list wherein the name of the list is passed as the parameter


sortlist.py - D:/Programs/python/sortlist.py (3.7.0)
File Edit Format Run Options Window Help

```
enter number7
enter number6
enter number5
enter number4
enter number3
enter number2
enter number1
[1, 2, 3, 4, 5, 6, 7, 8]
>>>
=================== RESTART: D:/Progra
enter how many numbers7
enter number3
enter number4
enter number1
enter number2
enter number6
enter number5
enter number7
Sorted list is [1, 2, 3, 4, 5, 6, 7]
>>>
```

```
data=[]
def readlist():
 n=int(input('enter how many numbers'))
 for i in range(0,n):
  element=int(input("enter number"))
  data.append(element)
def sortlist(data):
 for i in range(0,len(data)):
  for j in range(i+1,len(data)):
   if data[i]>data[j]:
    t=data[i]
    data[i]=data[j]
    data[j]=t
 print("Sorted list is", data)
readlist()
sortlist(data)
```

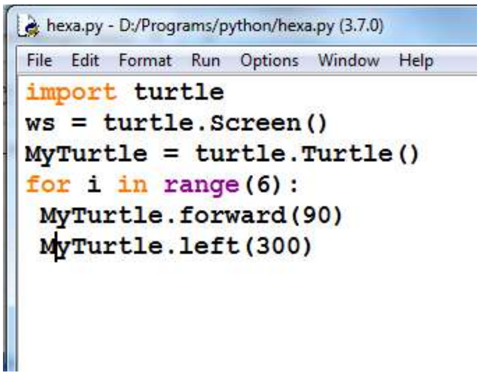
14 B)Write a Python program to check the validity of a password given by the user.


password.py - D:/Programs/python/password.py (3.7.0)
File Edit Format Run Options Window Help
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help

```
import re
p= input("Input your password")
x = True
while x:
    if (len(p)<6):
        break
    elif not re.search("[a-z]",p):
        break
    elif not re.search("[0-9]",p):
        break
    elif not re.search("[A-Z]",p):
        break
    elif not re.search("[$#@]",p):
        break
    elif re.search("\s",p):
        break
    else:
        print("Valid Password")
        x=False
        break

if x:
    print("Not a Valid Password")
```

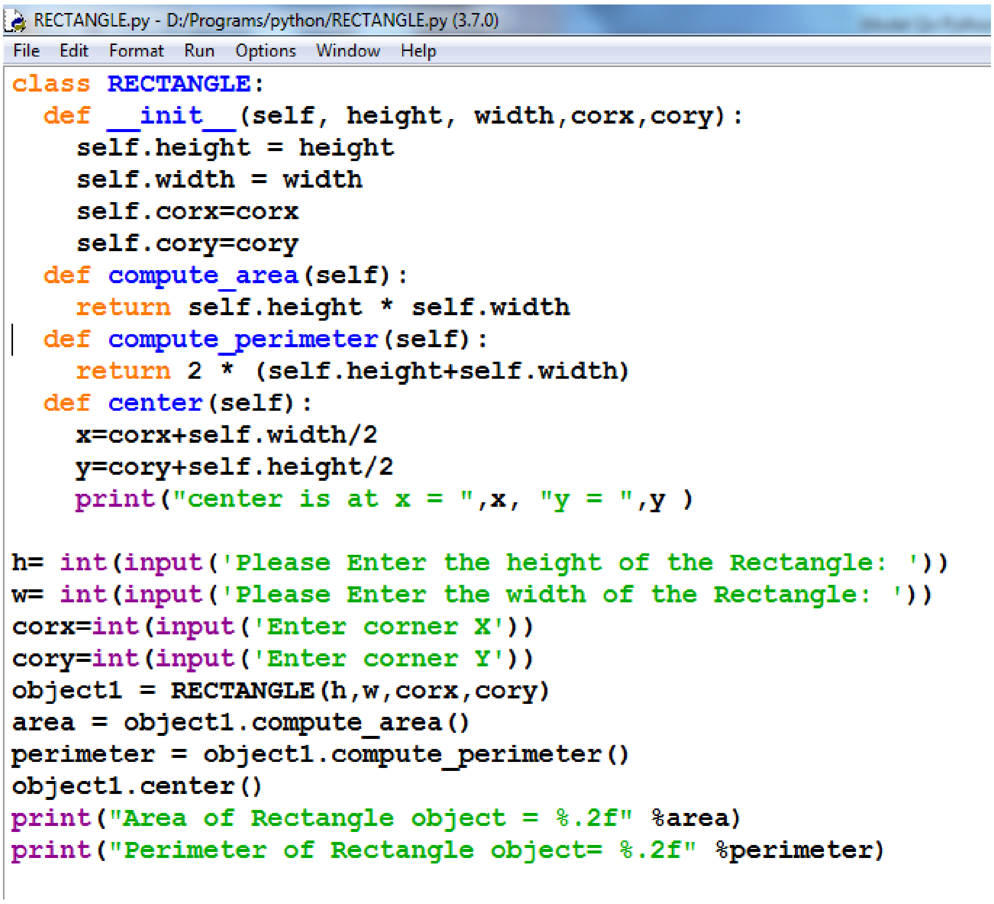
```
Python 3.7.0 (default, Aug
32
Type "copyright", "credits'
>>>
==================== RESTART:
Input your passwordrttA@
Not a Valid Password
>>> |
```

15 A) Write a program to draw a hexagon using turtle


hexa.py - D:/Programs/python/hexa.py (3.7.0)
File Edit Format Run Options Window Help

```
import turtle
ws = turtle.Screen()
MyTurtle = turtle.Turtle()
for i in range(6):
 MyTurtle.forward(90)
 MyTurtle.left(300)
```

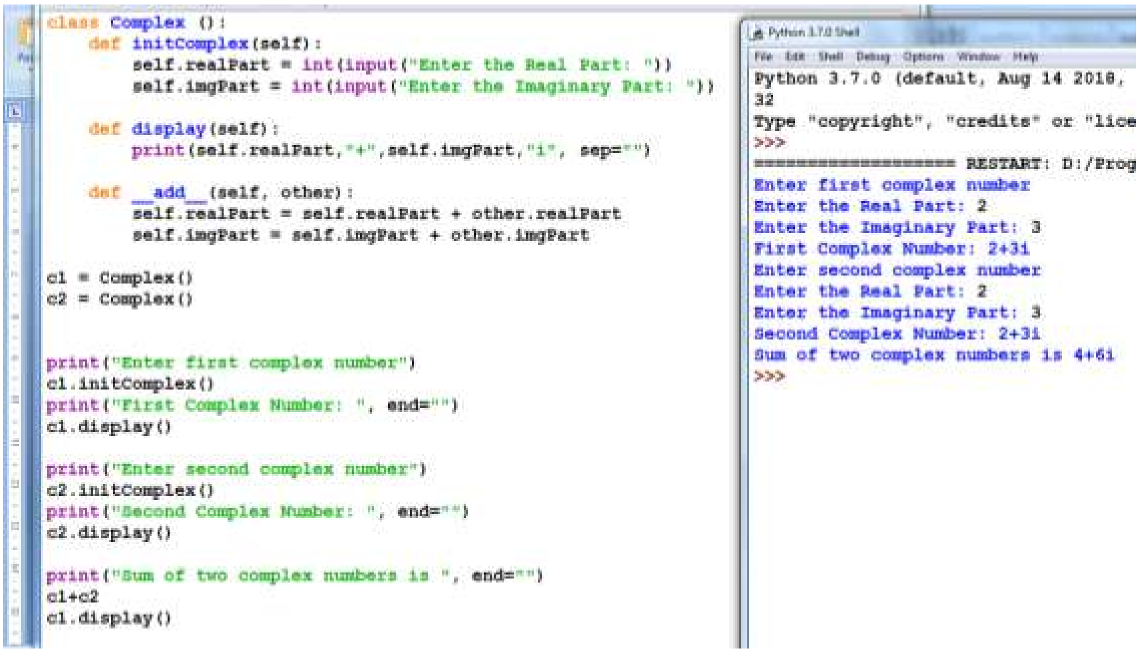
17 A) Write a Python program to express the instances as return values to define a class RECTANGLE with parameters *height, width, corner_x,* and *corner_y* and member functions to find center, area, and perimeter of an instance.


RECTANGLE.py - D:/Programs/python/RECTANGLE.py (3.7.0)
File Edit Format Run Options Window Help

```python
class RECTANGLE:
  def __init__(self, height, width,corx,cory):
    self.height = height
    self.width = width
    self.corx=corx
    self.cory=cory
  def compute_area(self):
    return self.height * self.width
  def compute_perimeter(self):
    return 2 * (self.height+self.width)
  def center(self):
    x=corx+self.width/2
    y=cory+self.height/2
    print("center is at x = ",x, "y = ",y )

h= int(input('Please Enter the height of the Rectangle: '))
w= int(input('Please Enter the width of the Rectangle: '))
corx=int(input('Enter corner X'))
cory=int(input('Enter corner Y'))
object1 = RECTANGLE(h,w,corx,cory)
area = object1.compute_area()
perimeter = object1.compute_perimeter()
object1.center()
print("Area of Rectangle object = %.2f" %area)
print("Perimeter of Rectangle object= %.2f" %perimeter)
```

18 A) same as 17 A (CIRCLE instead of RECTANGLE)

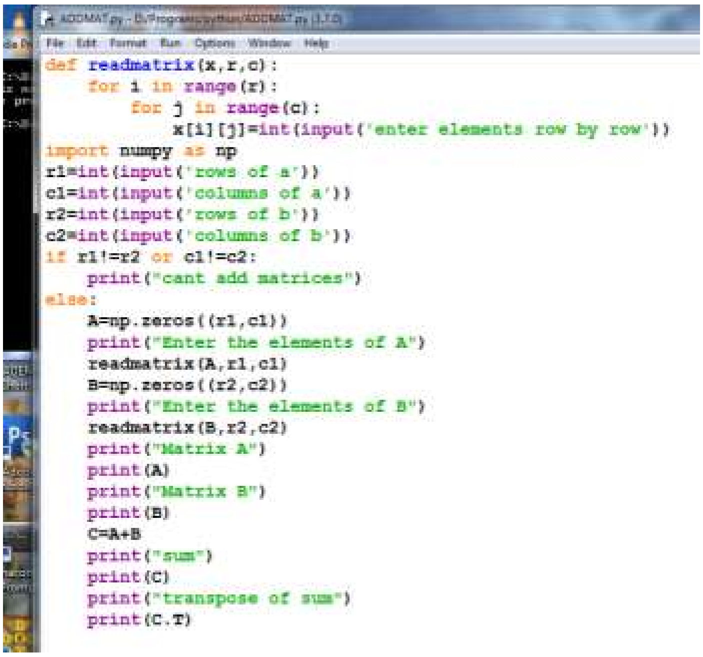
18 B) Write Python program to create a class called as **Complex** and implement *__add__( )* method to add two complex numbers. Display the result by overloading the + Operator



```python
class Complex ():
    def initComplex(self):
        self.realPart = int(input("Enter the Real Part: "))
        self.imgPart = int(input("Enter the Imaginary Part: "))

    def display(self):
        print(self.realPart,"+",self.imgPart,"i", sep="")

    def __add__(self, other):
        self.realPart = self.realPart + other.realPart
        self.imgPart = self.imgPart + other.imgPart

c1 = Complex()
c2 = Complex()


print("Enter first complex number")
c1.initComplex()
print("First Complex Number: ", end="")
c1.display()

print("Enter second complex number")
c2.initComplex()
print("Second Complex Number: ", end="")
c2.display()

print("Sum of two complex numbers is ", end="")
c1+c2
c1.display()
```

```
Python 3.7.0 (default, Aug 14 2018,
32
Type "copyright", "credits" or "lice
>>>
====================== RESTART: D:/Prog
Enter first complex number
Enter the Real Part: 2
Enter the Imaginary Part: 3
First Complex Number: 2+3i
Enter second complex number
Enter the Real Part: 2
Enter the Imaginary Part: 3
Second Complex Number: 2+3i
Sum of two complex numbers is 4+6i
>>>
```

19 A) Write a Python program to add two matrices and also find the transpose of the resultant matrix



```
def readmatrix(x,r,c):
    for i in range(r):
        for j in range(c):
            x[i][j]=int(input('enter elements row by row'))
import numpy as np
r1=int(input('rows of a'))
c1=int(input('columns of a'))
r2=int(input('rows of b'))
c2=int(input('columns of b'))
if r1!=r2 or c1!=c2:
    print("cant add matrices")
else:
    A=np.zeros((r1,c1))
    print("Enter the elements of A")
    readmatrix(A,r1,c1)
    B=np.zeros((r2,c2))
    print("Enter the elements of B")
    readmatrix(B,r2,c2)
    print("Matrix A")
    print(A)
    print("Matrix B")
    print(B)
    C=A+B
    print("sum")
    print(C)
    print("transpose of sum")
    print(C.T)
```

19 B) Given a file “auto.csv” of automobile data with the fields *index, company, body-style, wheel-base, length, engine-type, num-of-cylinders, horsepower, average-mileage,* and *price***,** write Python codes using Pandas to ....

Reading the data file and showing the first five records

import pandas as pd
df = pd.read_csv("Automobile_data.csv")
df.head(5)

1) Clean and Update the CSV file

```
import pandas as pd
df = pd.read_csv("Automobile_data.csv",
   na_values={
'price':["?","n.a"],
'stroke':["?","n.a"],
'horsepower':["?","n.a"],
'peak-rpm':["?","n.a"],
'average-mileage':["?","n.a"]})
print (df)
df.to_csv("Automobile_data.csv")
```

2) Find the highest priced car of all companies

```
import pandas as pd
df = pd.read_csv("Automobile_data.csv")
df.groupby('company')[['company','price']].max()
```

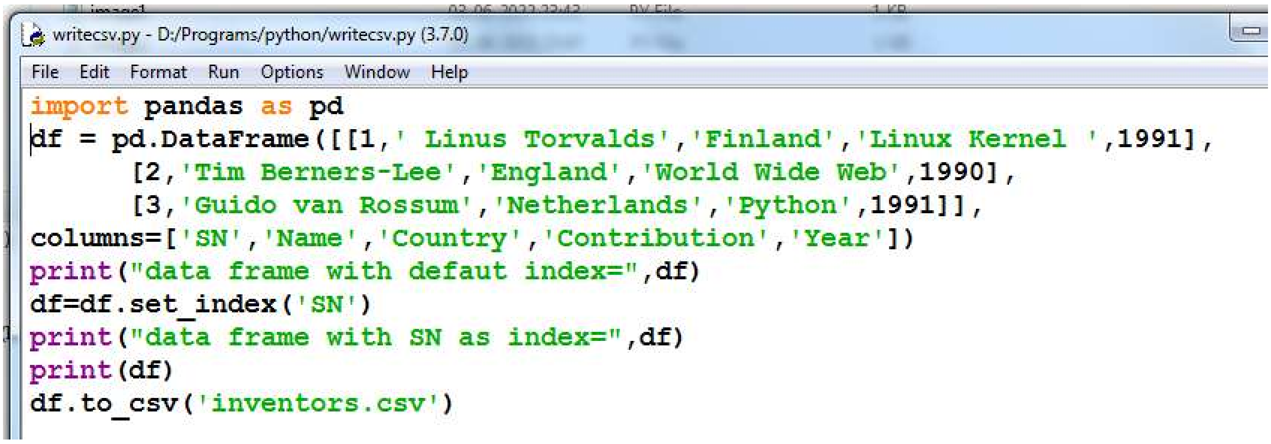
3)Print total cars of all companies

```
import pandas as pd
df = pd.read_csv("Automobile_data.csv")
df.groupby('company')['company'].count()
```

4) Find the average mileage of all companies

```
import pandas as pd
df = pd.read_csv("Automobile_data.csv")
df.groupby('company')[['company','average-mileage']].mean()
```

20 A) Write Python program to write the data given below to a CSV file.


writecsv.py - D:/Programs/python/writecsv.py (3.7.0)
File Edit Format Run Options Window Help

```
import pandas as pd
df = pd.DataFrame([[1,' Linus Torvalds','Finland','Linux Kernel ',1991],
       [2,'Tim Berners-Lee','England','World Wide Web',1990],
       [3,'Guido van Rossum','Netherlands','Python',1991]],
columns=['SN','Name','Country','Contribution','Year'])
print("data frame with defaut index=",df)
df=df.set_index('SN')
print("data frame with SN as index=",df)
print(df)
df.to_csv('inventors.csv')
```

20 B) Given the sales information of a company as CSV file with the following fields *month_number, facecream, facewash, toothpaste, bathingsoap, shampoo, moisturizer, total_units, total_profit*. Write Python codes to visualize the data as follows

1) Toothpaste sales data of each month and show it using a scatter plot

2) Face cream and face wash product sales data and show it using the bar chart

3) Calculate total sale data for last year for each product and show it using a Pie chart.

```
import pandas as pd
import matplotlib.pyplot as plt
import os
import numpy as np
import matplotlib

comp_sales_df = pd.read_csv('company_sales_data.csv')
comp_sales_df

# Toothpaste sales data of each month and show it using a scatter plot

sns.scatterplot(x=comp_sales_df.month_number,
y=comp_sales_df.toothpaste)
plt.grid(True, linewidth= 2, linestyle = "-")
plt.xlabel("Months Number")
plt.ylabel("Toothpastes sold")
plt.title(" Toothpaste sale data each month ")
plt.xticks(np.arange(1, 13))
plt.show()
```

```
# Face cream and face wash product sales data and show it using the bar
chart
plt.bar(comp_sales_df.month_number, comp_sales_df.facecream,
label='Facecream',color ='g')
plt.bar(comp_sales_df.month_number, comp_sales_df.facewash, label=
'Facewash', color = "r" )
plt.legend()
plt.xlabel("Months")
plt.ylabel("Sold units number ")
plt.xticks(np.arange(1, 13))
plt.show()
```

#Calculate total sale data for last year for each product and show it using a Pie chart.

```
new_comp_sales_df =
pd.read_csv('company_sales_data.csv').set_index('month_number')
new_set = new_comp_sales_df[['facecream', 'facewash', 'toothpaste',
'bathingsoap', 'shampoo', 'moisturizer']]
new1 = ['facecream', 'facewash', 'toothpaste', 'bathingsoap', 'shampoo',
'moisturizer']
new_set.sum(axis=0).plot(kind='pie', label= '', figsize=(6,6), autopct
='%1.1f%%')
plt.title(" Sales data")
plt.show()
```