FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

20MCA131- PROGRAMMING LAB

LABORATORY RECORD

Name: FARZEENA P A

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 Batch: SEMESTER -1 A Roll No: 57

FEDERAL INSTITUTE OF

SCIENCE AND TECHNOLOGY

(FISAT)™

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

'FOCUS ON EXCELLENCE'

Name : FARZEENA P A

Branch : MASTER OF COMPUTER APPLICATION

Semester : 1 **Roll No: 57**

University Exam.Reg. No: FIT21MCA-2057

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done and submitted to Kerala Technological University in partial fulfillment for the award of the Master Of Computer Applications is a record of the original research work done by **FARZEENA P A** in the **20MCA131- PROGRAMMING LAB** Laboratory of the Federal Institute of Science and Technology during the academic year 2020-2021.

Signature of Staff in Charge

Name:

Date:

Signature of H.O.D

Name:

Date of University practical examination ……

Signature of

Internal Examiner

Signature of

External Examiner

CONTENT

6	10-11-21	Enter two lists of integers. Check	17	
		a. Whether they are of same length.		
		b. Whether list sums to same value.		
		c. Whether any value occurs in both.		
7	10-11-21	Get a string from an input string where all occurrence of first character replaced with '$', except first character	19	
8	10-11-21	Create a string from given string where first and last characters exchanged.	20	
9	10-11-21	Accept the radius from user and find area of circle.	21	
10	10-11-21	Find biggest of 3 numbers entered.	22	
11	11-11-21	Accept a file name from user and print extension for that.	23	
12	11-11-21	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	24	
13	11-11-21	Accept an integer n and compute n+nn+nnn	25	
14	11-11-21	Print out all colors from color list1 not contained in color list2.	26	
15	11-11-21	Create a single string separated with space from two	27	

		strings by swapping the character at position 1.		
16	17-11-21	Sort dictionary in ascending and descending order.	28	
17	17-11-21	Merge two dictionaries.	29	
18	17-11-21	Find gcd of two numbers.	30	
19	17-11-21	Form a list of integers, create a list removing even numbers	31	
		COURSE OUTCOME 2	32	
1	25-11-21	Program to find the factorial of a number.	33	
2	25-11-21	Generate Fibonacci series of N terms.	34	
3	25-11-21	Find the sum of all items in a list.	35	
4	25-11-21	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	36	
5	2-12-21	Display the given pyramid with step numbers accepted from user.	38	
6	2-12-21	Count the number of characters (character frequency) in a string	39	

7	2-12-21	Add 'ing' at the end of a given string.If it already ends with 'ing' , then add 'ly'.	41	
8	9-12-21	Accept a list of words and return length of longest word.	42	
9	9-12-21	Construct following pattern using nested loop.	43	
10	9-12-21	Generate all factors of a number.	45	
		COURSE OUTCOME 3	46	
1	29-1-22	Create a package graphics with modules rectangle, circle and sub-package 3D graphics with module cuboid and sphere.	47	
		COURSE OUTCOME 4	49	
1	13-1-22	Create Rectangle class with attributes length and breadth and methods to find area and perimeter.	50	
2	13-1-22	Create a Bank account with members account number, name, type of account and balance..	51	
3	13-1-22	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of two rectangles	53	
4	20-1-22	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of two time.	54	

5	20-1-22	Create a class Publisher (name). Derive class Book from Publisher with attributes title and author.	55	
		COURSE OUTCOME 5	57	
1	3-2-22	Write a program to read a file line by line and store it into a list	58	
2	3-2-22	Write a Python program to read each row from a given csv file and print a list of strings.	59	

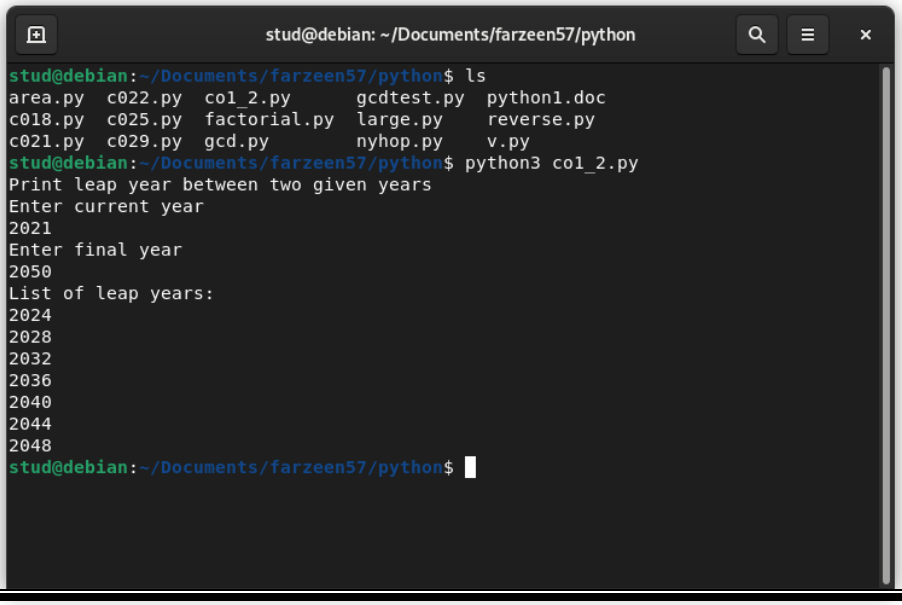
COURSE OUTCOME 1

1. DISPLAY FUTURE LEAP YEARS FROM CURRENT YEAR TO A FINAL YEAR ENTERED BY USER

SOURCE CODE:-

```
print ("Print leap year between two given years")
print ("Enter current year")
startYear = int(input())
print ("Enter final year")
endYear = int(input())
print ("List of leap years:")
for year in range(startYear, endYear):
if (0 == year % 4) and (0 != year % 100) or (0 == year % 400):
        print (year)
```

OUTPUT:-

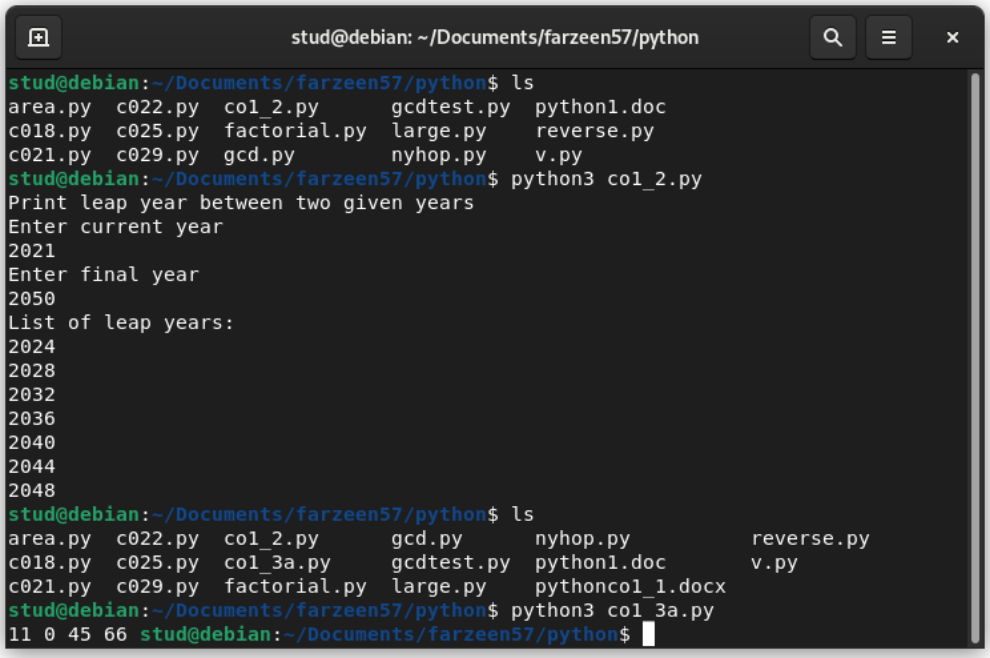


2. LIST COMPREHENSIONS: GENERATE POSITIVE LIST OF NUMBERS FROM A GIVEN LIST OF INTEGERS

SOURCE CODE:-

```
list1 = [11, -21, 0, 45, 66, -93]
for num in list1:
    if num >= 0:
        print(num, end = " ")
```

OUTPUT:-

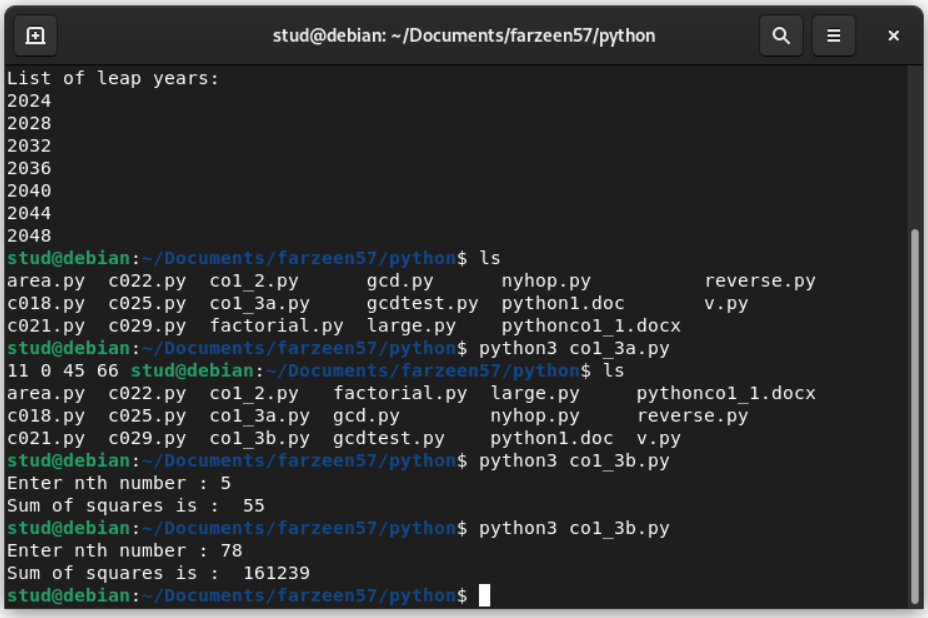


2. LIST COMPREHENSIONS: SQUARE OF NUMBERS

SOURCE CODE:-

```
n = int(input("Enter nth number : "))
sum = 0
for s in range(1, n+1):
   sum = sum + (s*s)
print("Sum of squares is : ", sum)
```

OUTPUT:-

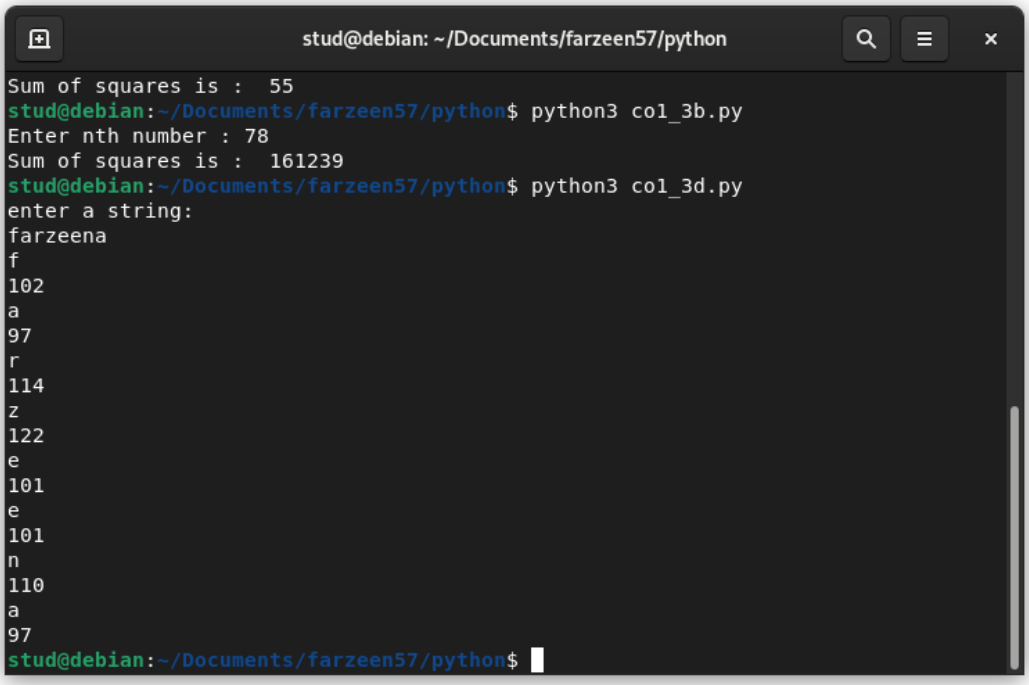


2.LIST COMPREHENSIONS: LIST ORDINAL VALUE OF EACH ELEMENT OF A WORD

SOURCE CODE:-

```
user_name = input("enter a string:\n")
for char in range(0,len(user_name)):
 print(user_name[char])
 print(ord(user_name[char]))
```

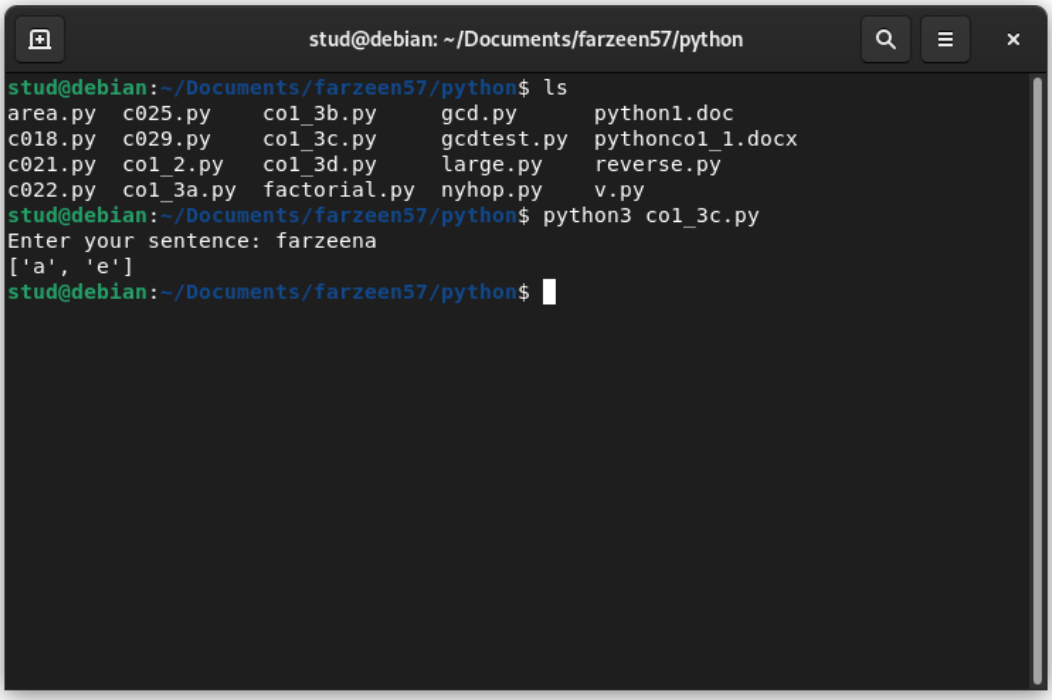
OUTPUT:-



2.LIST COMPREHENSIONS: FORM A LIST OF VOWELS SELECTED FROM A GIVEN WORD

```
string = input('Enter your sentence: ')
list1=[]
for x in 'aeiou':
   if x in string:
        list1.append(x)
print (list1)
```

OUTPUT:-

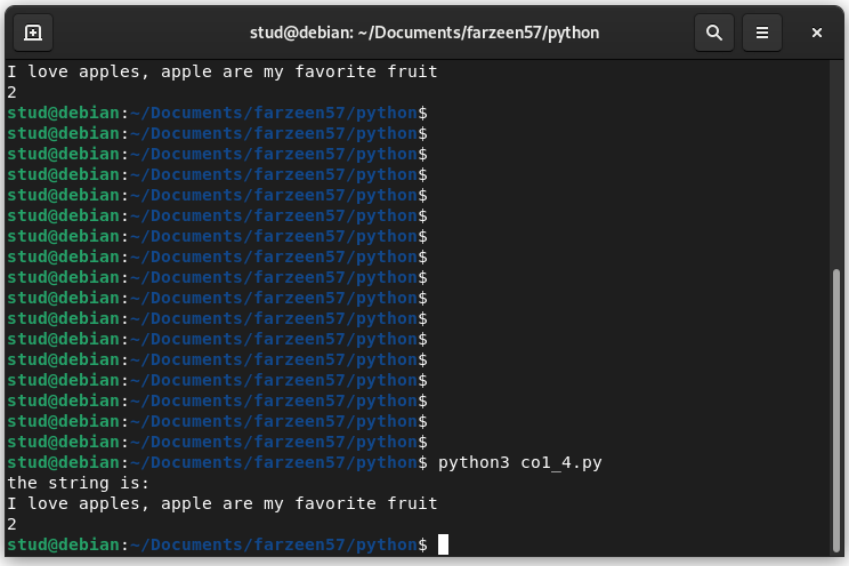


3. COUNT THE OCCURRENCES OF EACH WORD IN A LINE OF TEXT

SOURCE CODE:-

```
string = "I love apples, apple are my favorite fruit"
print ("the string is:")
print (string)
x = string.count("apple")
print(x)
```

OUTPUT:-

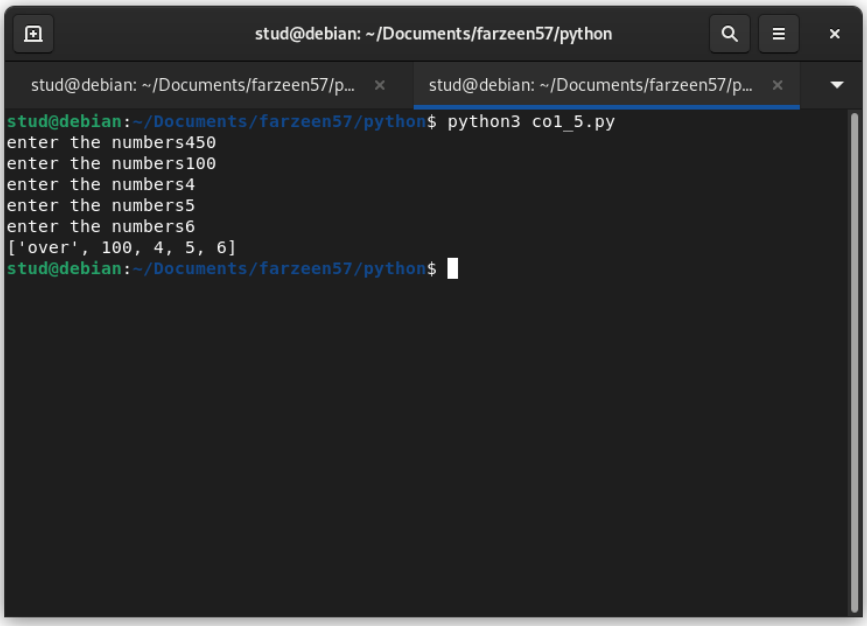


4. PROMPT THE USER FOR A LIST OF INTEGERS. FOR ALL VALUES GREATER THEN 100, STORE 'OVER' INSTEAD

SOURCE CODE:-

```
list1=[]
for i in range(5):
  x = int(input("enter the numbers"))
  if x>100:
        x="over"
  list1.append(x)
print(list1)
```

OUTPUT:-

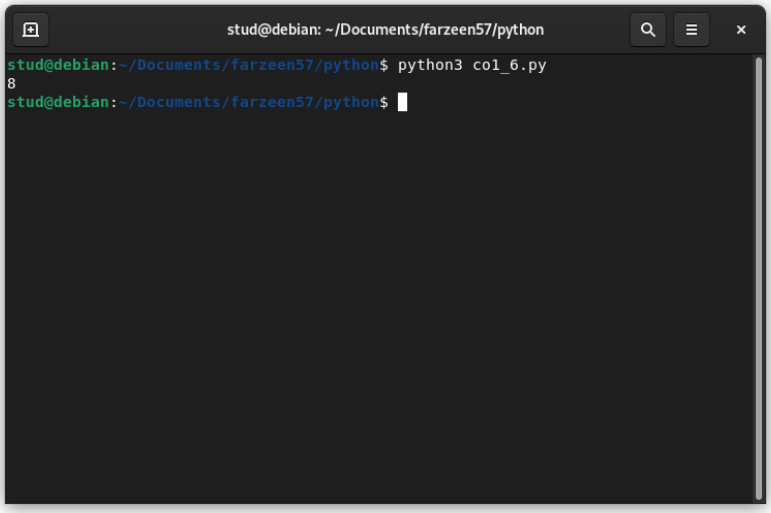


5. STORE A LIST OF FIRST NAMES. COUNT THE OCCURRENCES OF 'a' WITHIN THE LIST

SOURCE CODE:-

```
List1=["farzeena","emel","anupama","appu","zara"]
count=0
for x in List1:
        for k in x:
                if k=='a':
                        count=count+1
print(count)
```

OUTPUT:-



6. ENTER TWO LISTS OF INTEGERS. CHECK:

(A) WHETHER LISTS ARE OF SAME LENGTH,

(B) WHETHER LISTS SUM TO SAME VALUE,

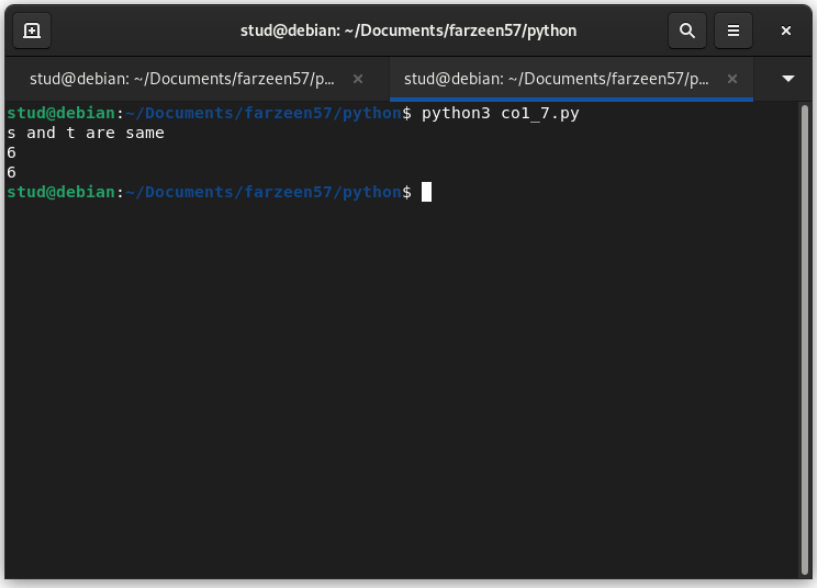
WHETHER ANY VALUE OCCUR IN BOTH

SOURCE CODE:-

```
l1 = [1,2,3]
l2 = [6]
s = 0
t = 0
for i in range(len(l1)):
        s = s + l1[i]

for i in range(len(l2)):

        t = t+l2[i]



if (s == t):
        print("s and t are same")
        print (s);
        print (t);
```

```
else:
        print(“not same”)
        print (“sum1=”);
        print (s);
        print (“sum2=”);
        print (t);
```

OUTPUT:-

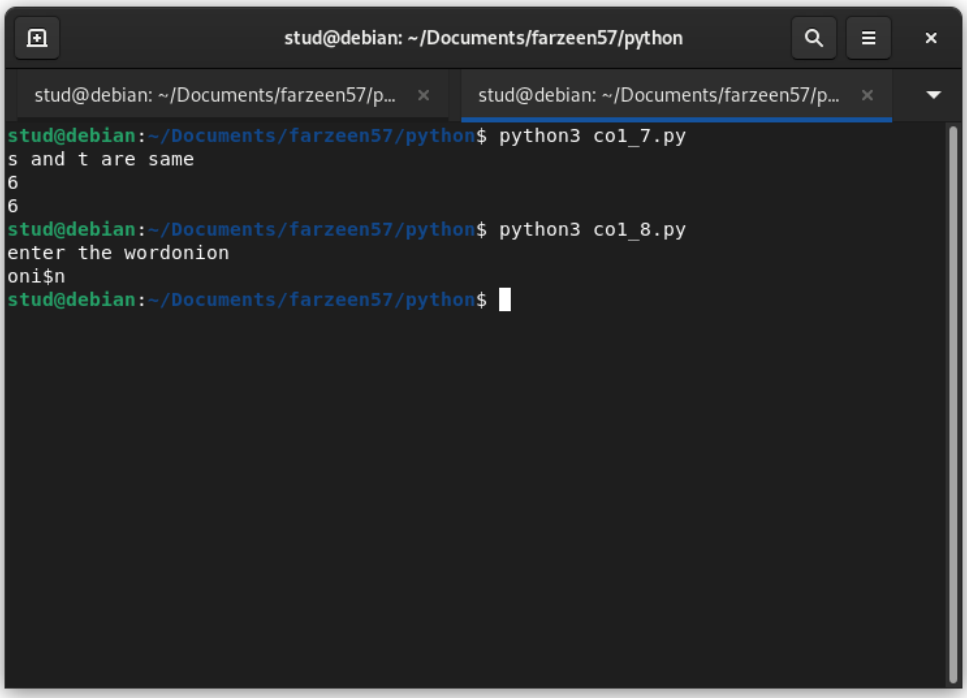


7. GET A STRING FROM AN INPUT STRING WHERE ALL OCCURRENCES OF FIRST CHARACTER '$', EXCEPT FIRST CHARACTER.

SOURCE CODE:-

```
string = input("enter the word")
a = string[0]
x = string[1:len(string)]
x = x.replace("o","$")
print (a+x)
```

OUTPUT:-



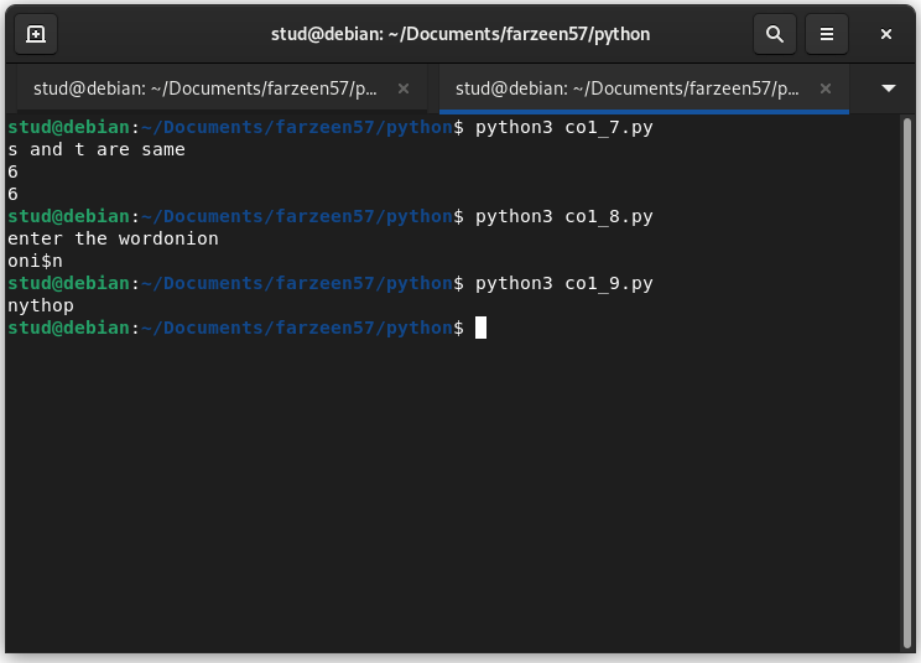
8. CREATE A STRING FROM A GIVEN STRING WHERE FIRST AND LAST CHARACTERS EXCHANGED

[PYTHON -> NYTHOP]

SOURCE CODE:-

```
s = "python"
x = s[0]
y = s[-1]
z = s[1:(len(s)-1)]
print (y+z+x)
```

OUTPUT:-

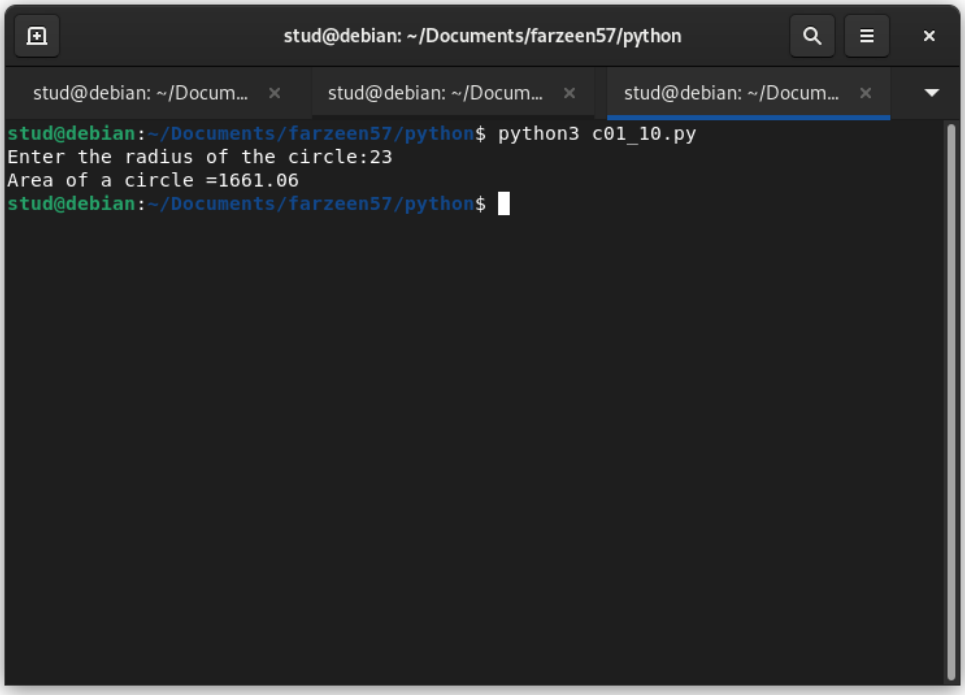


9. ACCEPT THE RADIUS FROM USER AND FIND AREA OF CIRCLE

SOURCE CODE:-

```
pi = 3.14
r = float(input("Enter the radius of the circle:"))
area = pi * r * r
print("Area of a circle =",end="")
print(area)
```

OUTPUT:-

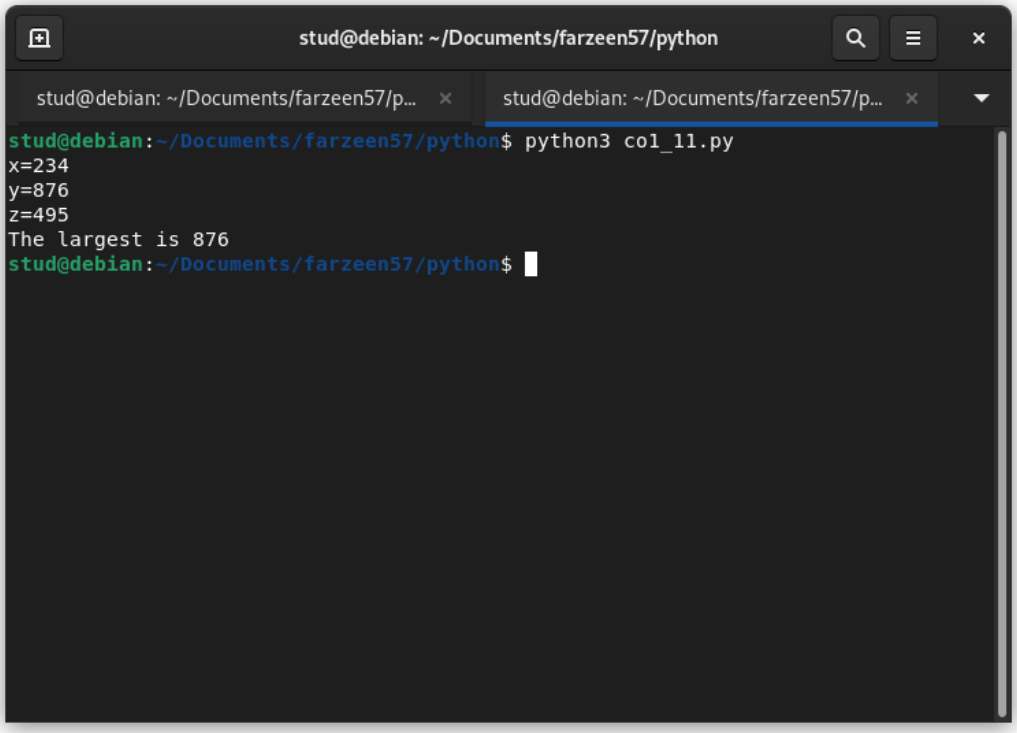


10. FIND BIGGEST OF 3 NUMBERS ENTERED

SOURCE CODE:-

```
x=int(input("x="))
y=int(input("y="))
z=int(input("z="))
if (x>=y) and (x>=z):
        A=x
elif (y>=x) and (y>=z):
        A=y
else:
        A=z
print("The largest is",A)
```

OUTPUT:-

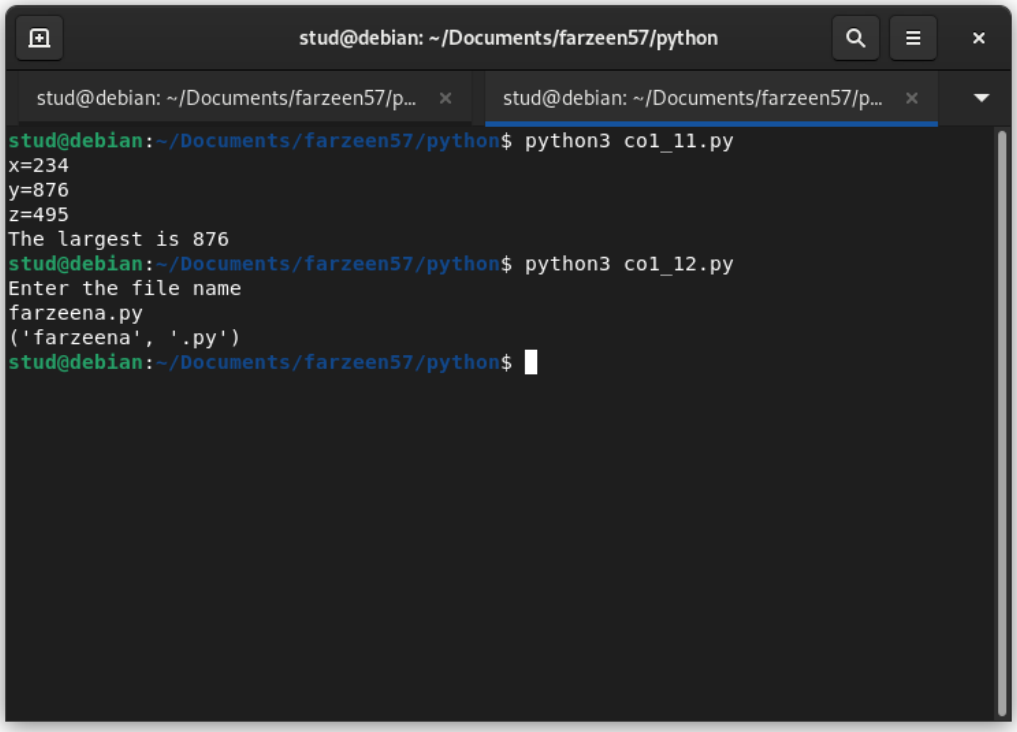


11. ACCEPT A FILE NAME FROM USER AND PRINT EXTENSION FOR THAT

SOURCE CODE:-

```
import os
a=input("Enter the file name\n")
print(os.path.splitext(a))
```

OUTPUT:-



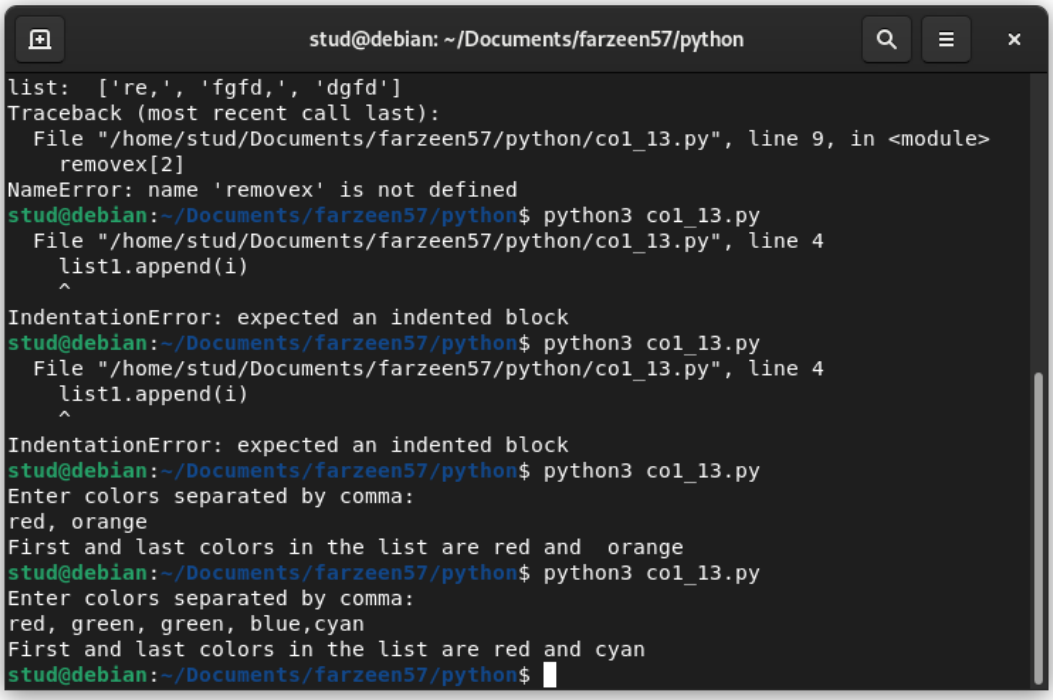
12. CREATE A LIST OF COLORS FROM COMMA-SEPARATED COLOR NAMES ENTERED BY THE USER. DISPLAY FIRST AND LAST COLORS

SOURCE CODE:-

```
list1=[]
string=input("Enter colors separated by comma:\n")
for i in string.split(","):
        list1.append(i)
```

```
print("First and last colors in the list are",list1[0],"and",list1[-1])
```

OUTPUT:-

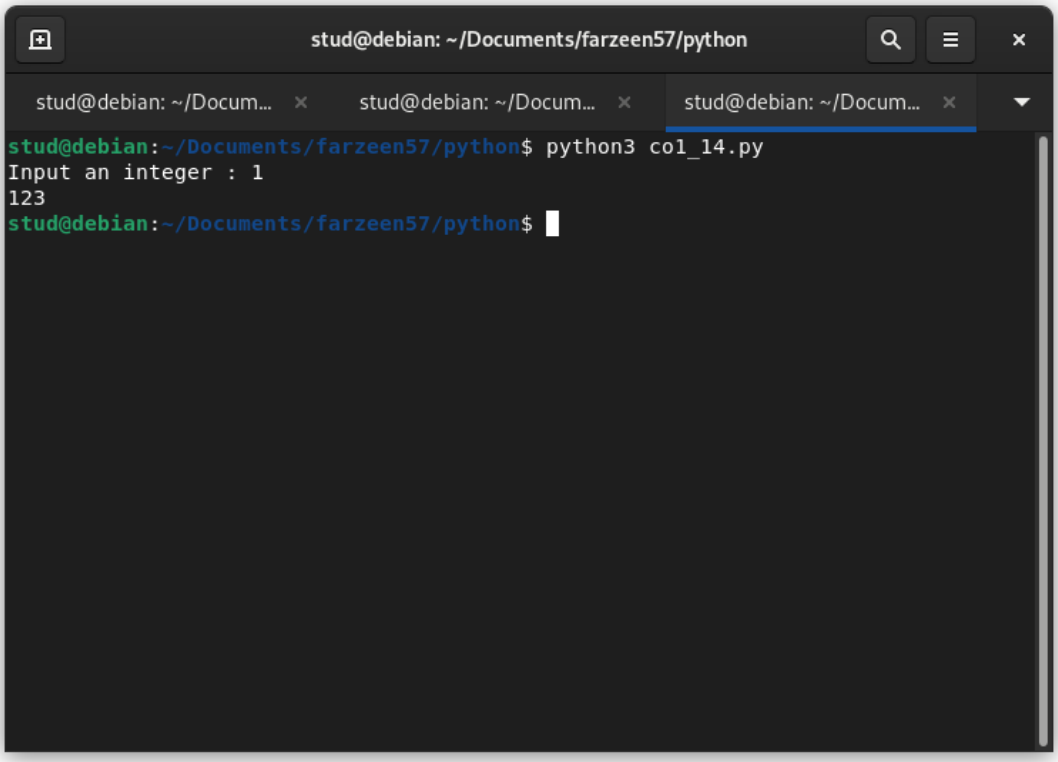


13. ACCEPT AN INTEGER N AND COMPUTE N+NN+NNN.

SOURCE CODE:-

```
a = int(input("Input an integer : "))

n1 = (a*1)

n2 = (a*11)

n3 = (a*111)

print (n1+n2+n3)
```

OUTPUT:-

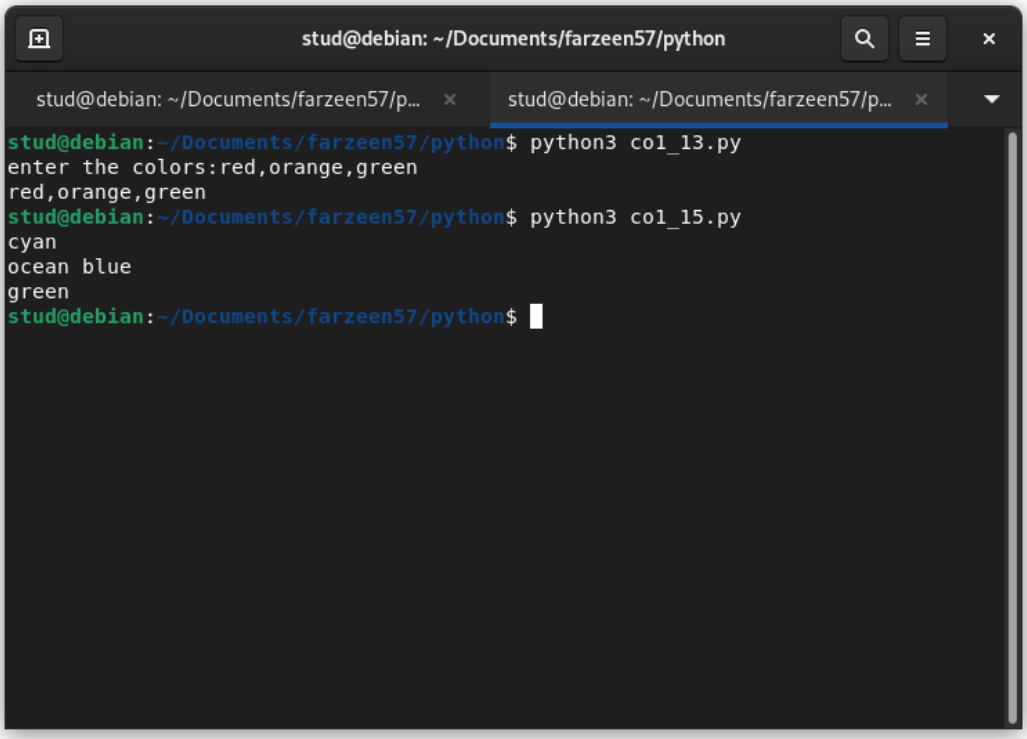


14. PRINT OUT ALL COLORS FROM COLOR-LIST 1 NOT CONTAINED IN COLOR-LIST 2

SOURCE CODE:-

```
l1 = ['red','orange','blue','cyan','ocean blue','green']
l2 = ['red','blue','magenta','orange']
for i in l1:
        if i not in l2:
                print (i)
```

OUTPUT:-



15. CREATE A SINGLE STRING SEPARATED WITH SPACE FROM TWO STRINGS BY SWAPPING THE CHARACTER AT THE POSITION 1

SOURCE CODE:-

```
string1="PROGRAMMING"
string2="LAB"
f1=string1[0]
f2=string2[0]
```

```
string=f2+string1[1:]+" "+f1+string2[1:]
print("The new string is :",string)
```

OUTPUT:-

```
The new string is : LROGRAMMING PAB


...Program finished with exit code 0
Press ENTER to exit console.
```

16. SORT DICTIONARY IN ASCENDING AND DESCENDING ORDER

SOURCE CODE:-

```
dict1={"F":15,"A":32,"R":26,"Z":41}
l=list(dict1.items())
print(l)
l.sort()
print("Ascending Order is \n",l)
l=list(dict1.items())
l.sort(reverse=True)
```

```
print("Descending order is \n",l)
```

OUTPUT:-

```
Ascending Order is
 [('A', 32), ('F', 15), ('R', 26), ('Z', 41)]
Descending order is
 [('Z', 41), ('R', 26), ('F', 15), ('A', 32)]


...Program finished with exit code 0
Press ENTER to exit console.
```

17. MERGE TWO DICTIONARIES

SOURCE CODE:-

```
dic1={"Name":"FARZEENA","Age":"20"}
dic2={"nationality":"Indian","Gender":"F"}
dic1.update(dic2)
print(dic1)
```

OUTPUT:-

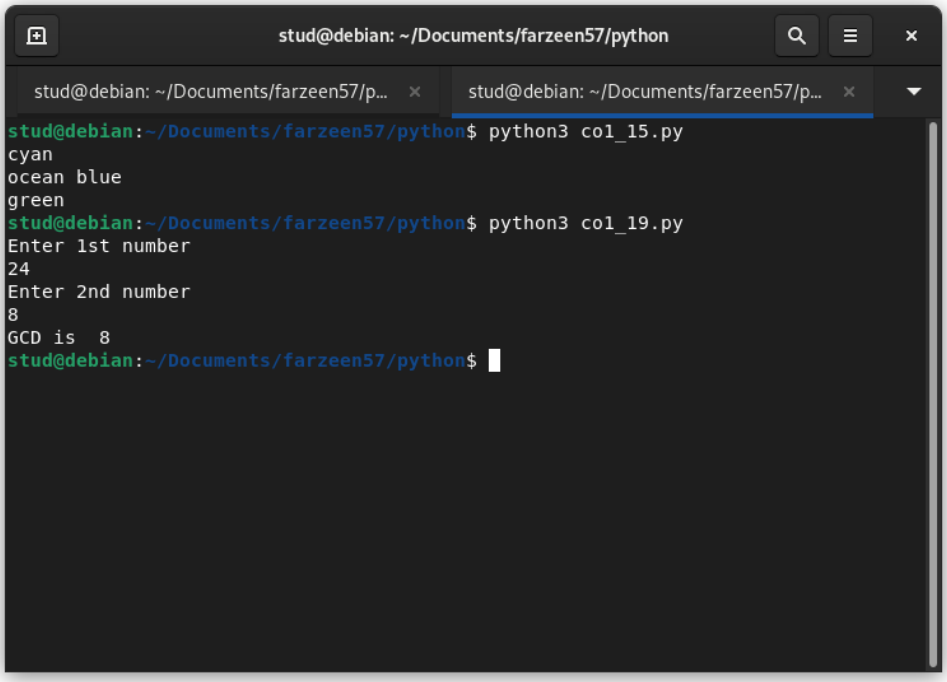
```
input
{'Name': 'FARZEENA', 'Age': '20', 'nationality': 'Indian', 'Gender': 'F'}


...Program finished with exit code 0
Press ENTER to exit console.
```

18. FIND GCD OF TWO NUMBERS

SOURCE CODE:-

```
x=(int(input("Enter 1st number\n")))
y=(int(input("Enter 2nd number\n")))
z = min(x,y)
for i in range(1,z+1):
        if((x%i)==0 and (y%i==0)):
                gcd=i
print("GCD is ",gcd)
```

OUTPUT:-

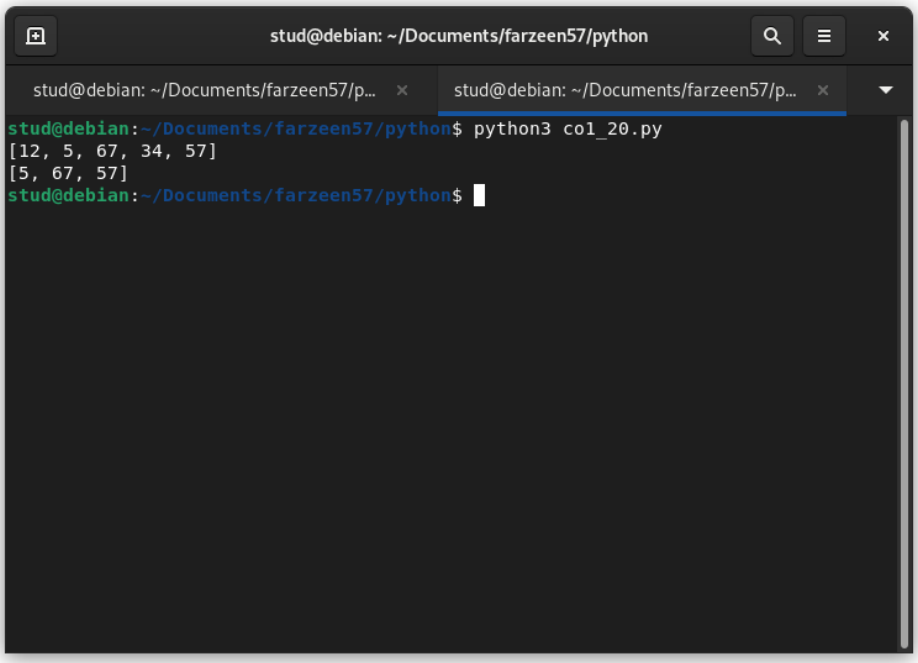


19. FROM A LIST OF INTEGERS, CREATE A LIST REMOVING EVEN NUMBERS

SOURCE CODE:-

```
l1 = [12,5,67,34,57]
print(l1)
for x in l1:
        if x%2==0:
                l1.remove(x)
print (l1)
```

OUTPUT:-



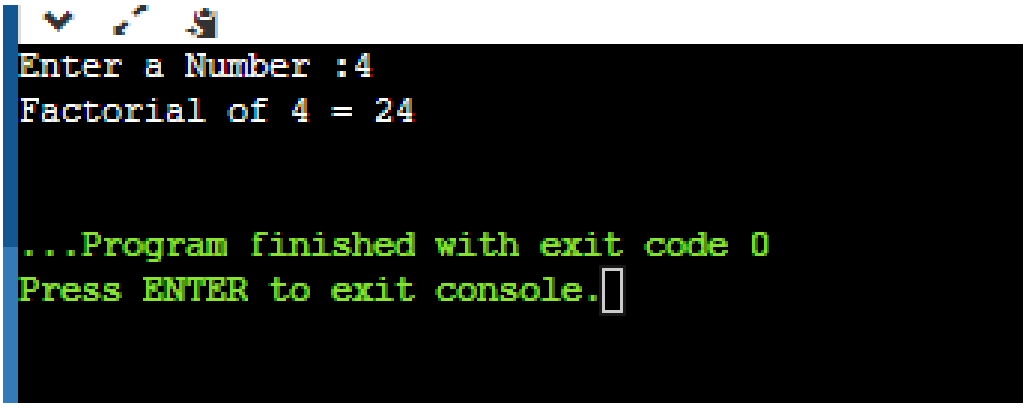
COURSE OUTCOME 2

1. PROGRAM TO FIND THE FACTORIAL OF A NUMBER

SOURCE CODE:-

```
n=int(input("Enter a Number :"))
factorial=1
for i in range(1,n+1):
    factorial=factorial*i
print("Factorial of",n,"=",factorial)
```

OUTPUT:-



2. GENERATE FIBONACCI SERIES OF N TERMS.

SOURCE CODE:-

```
n=int(input("Enter a Number :"))
print("The first",n,"fibonacci seriers is :")
f1=0
f2=1
for i in range(0,n):
    print(f1)
    f3=f1
    f1=f1+f2
    f2=f3
```

OUTPUT:-

```
Enter a Number :5
The first 5 fibonacci seriers is :
0
1
1
2
3
```

3. FIND THE SUM OF ALL ITEMS IN A LIST

SOURCE CODE:-

```
list1=[1,2,3,4,5,6,7]
summ=0
for i in list1:
    summ=summ+i
print("sum=",summ)
```

OUTPUT:-

```
sum= 28


...Program finished with exit code 0
Press ENTER to exit console.
```

4. GENERATE A LIST OF FOUR DIGIT NUMBERS IN A GIVEN RANGE WITH ALL THEIR DIGITS EVEN AND THE NUMBER IS A PERFECT SQUARE.

SOURCE CODE:-

```
limit1=1000
limit2=9999
list1=[]
for i in range(limit1,limit2):
   j=i
   digit=[]
   while(i!=0):
      digit.append(i%10)
      i=int(i/10)
   count=0
   for n in digit:
      if n%2==0:
         count=count+1

   if count==4:
      for k in range(31,100):
```

```
        if((k**2)==j):
            list1.append(j)


print(list1)
```

OUTPUT:-

```
[4624, 6084, 6400, 8464]


...Program finished with exit code 0
Press ENTER to exit console.
```

5. DISPLAY THE GIVEN PYRAMID WITH STEP NUMBERS ACCEPTED FROM USER.

EG : 4

1

2 4

3 6 9

4 8 12 16

SOURCE CODE:-

```
n=int(input("Enter a number :"))
for i in range(1,n+1):
   for j in range(i,(i*i)+1,i):
      print(j,"\t",end="")
   print("\n")
```

OUTPUT:-

```
Enter a number :6
1

2       4

3       6       9

4       8       12      16

5       10      15      20      25

6       12      18      24      30      36


...Program finished with exit code 0
Press ENTER to exit console.
```

6. COUNT THE NUMBER OF CHARACTERS (CHARACTER FREQUENCY) IN A STRING.

SOURCE CODE:-

```
string=input("Enter a string :")
ulist=[]
for i in string:
```

```
    if i not in ulist:
        ulist.append(i)


for i in ulist:
    count=0
    for j in string:
        if(i==j):
            count+=1
    print(i,"\t:",count)
```

OUTPUT:-

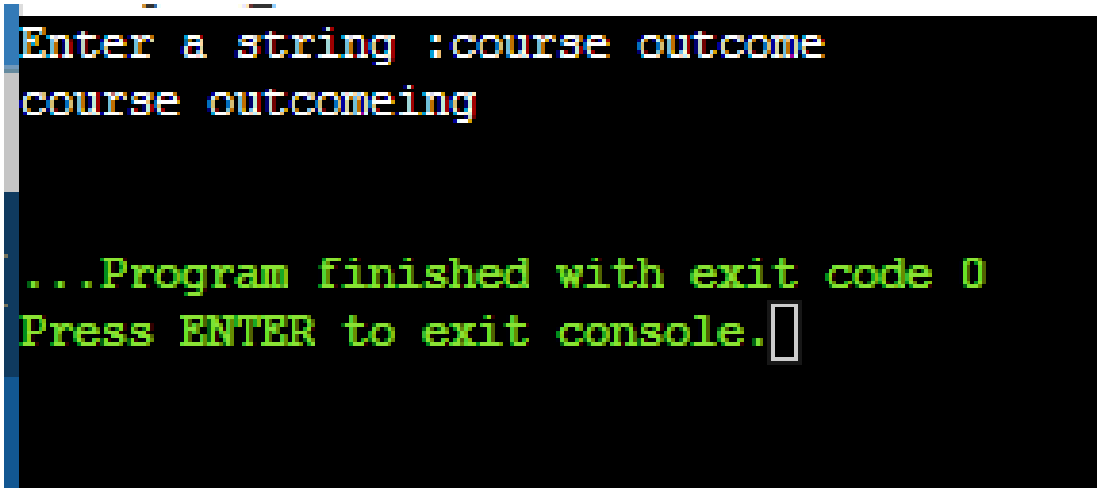
```
Enter a string :COURSE OUTCOME
C       : 2
O       : 3
U       : 2
R       : 1
S       : 1
E       : 2
        : 2
T       : 1
M       : 1


...Program finished with exit code 0
Press ENTER to exit console.
```

7. ADD 'ING' AT THE END OF A GIVEN STRING.IF IT ALREADY ENDS WITH 'ING' , THEN ADD 'LY'.

SOURCE CODE:-

```
string=input("Enter a string :")
if(string[-3:]=="ing"):
   string+="ly"
else:
   string+="ing"
print(string)
```

OUTPUT:-



8. ACCEPT A LIST OF WORDS AND RETURN LENGTH OF LONGEST WORD.

SOURCE CODE:-

```
wlist=[]
print("Enter 5 words :")
for i in range(0,5):
    wlist.append(input(""))
    temp=wlist[0]
for i in range(1,5):
    if len(wlist[i])>len(temp):
        temp=wlist[i]

print("Length of longest word is",len(temp))
```

OUTPUT:-

```
Enter 5 words :
course
outcome
one
and
two
Length of longest word is 7


...Program finished with exit code 0
Press ENTER to exit console.
```

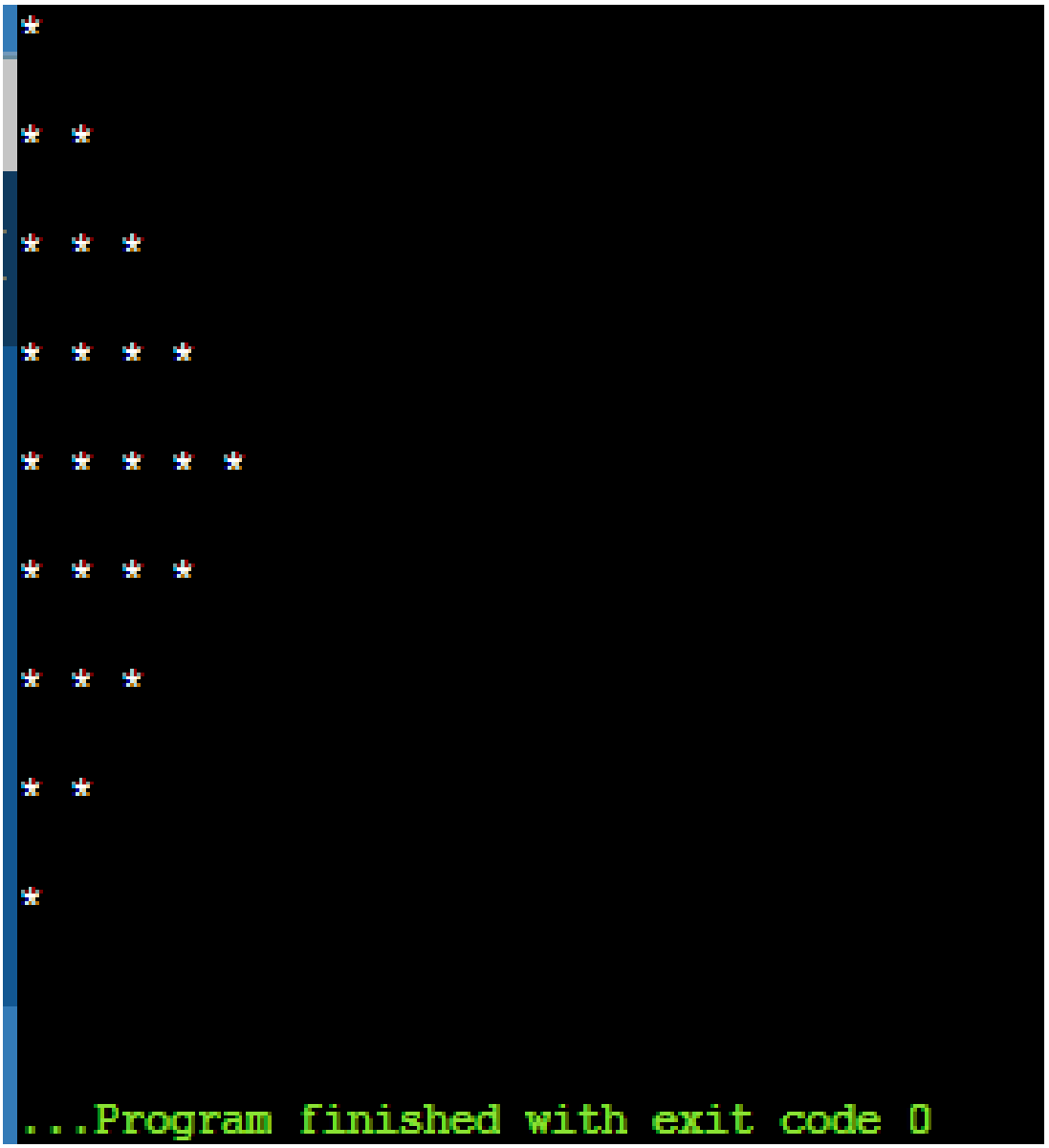
9. CONSTRUCT FOLLOWING PATTERN USING NESTED LOOP.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

SOURCE CODE:-

```
for i in range(1,6):
    for j in range(0,i):
        print("* ",end="")
    print("\n")
for i in range(4,0,-1):
    for j in range(0,i):
        print("* ",end="")
    print("\n")
```

OUTPUT:-



10. GENERATE ALL FACTORS OF A NUMBER.

SOURCE CODE:-

```
n=int(input("Enter a number :"))
print("The factors are :")
for i in range(1,n+1):
    if(n%i)==0:
        print(i)
```

OUTPUT:-

```
Enter a number :8
The factors are :
1
2
4
8


...Program finished with exit code 0
Press ENTER to exit console.
```

COURSE OUTCOME 3

1. CREATE A PACKAGE GRAPHICS WITH MODULES RECTANGLE, CIRCLE AND SUB PACKAGE 3D-GRAPHICS WITH MODULES CUBOID AND SPHERE

:-Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import statements).*

SOURCE CODE:-

>>graphics\circle.py

```
from math import pi
def area_circle(radius):
    return pi*radius*radius
def peri_circle(radius):
    return 2*pi*radius
```

>>graphics\rectangle.py

```
def area_rec(length,width):
    return length*width
def peri_rec(length,width):
    return 2*(length+width)
```

>>graphics\tdgraphics\cuboid.py

```
def area_cuboid(l,b,h):
    return 2*(l*h+b*h+l*b)
def volume_cuboid(l,b,h):
```

```
    return l*b*h
```

>>graphics\tdgraphics\sphere.py

```
from math import pi
def area_sphere(radius):
    return 4*(pi*radius*radius)
def peri_sphere(radius):
    return 2*pi*radius
```

OUTPUT:-

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> md graphics


    Directory: C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        03-03-2022  12.04 PM                graphics


PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> cd graphics
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics> notepad __init__.py
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics> notepad rectangle.py
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics> notepad circle.py
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics> md tdgraphics


    Directory: C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        03-03-2022  12.05 PM                tdgraphics


PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics> cd tdgraphics
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics\tdgraphics> notepad __init__.py
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics\tdgraphics> notepad sphere.py
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics\tdgraphics> notepad cuboid.py
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics\tdgraphics> cd..
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\graphics> cd..
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> notepad graphics.py
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python graphics.py
```

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python graphics.py

--------------circle-------------

area of circle with radius 10 is: 314.1592653589793
perimeter of circle with radius 10 is: 62.83185307179586


--------------rectangle--------------

area of rectangle with length and width 10 is: 100
perimeter of rectangle with length and breadth 10 is: 40


---------------cuboid-----------------

area of cuboid with length breadth and height 10 is : 600
volume of cuboid with length breadth and height 10 is: 1000


--------------------sphere-----------

area of the sphere having radius 10 is: 1256.6370614359173
perimeter of sphere having radius 10 is: 62.83185307179586
```

COURSE OUTCOME 4

1. CREATE RECTANGLE CLASS WITH ATTRIBUTES LENGTH AND BREADTH AND METHODS TO FIND AREA AND PERIMETER. COMPARE TWO RECTANGLE OBJECTS BY THEIR AREA

SOURCE CODE:-

```
class rectangle:
    def __init__(self,length,breadth):
        self.length = length
        self.breadth = breadth
    def area(self):
        return self.length*self.breadth
    def perimeter(self):
        return 2*(self.length+self.breadth)
#-------------------------------------------------------------
l = int(input("enter length of the rectangle1:"))
b = int(input("enter breadth of the rectangle1:"))
rect1 = rectangle(l,b)
a1 = rect1.area()
p1 = rect1.perimeter()
print("Area:",a1)
print("perimeter:",p1)
#-------------------------------------------------------------
l = int(input("enter length of the rectangle2:"))
b = int(input("enter breadth of the rectangle2:"))
rect2 = rectangle(l,b)
a2 = rect2.area()
p2 = rect2.perimeter()
print("Area:",a2)
print("perimeter:",p2)
#-------------------------------------------------------------
if (a1>a2):
    print("First rectangle has larger area\n")
elif(a1==a2):
```

```
    print("both rectangles are of same area\n")
else:
    print("second rectangle has larger area\n")
```

OUTPUT:-

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python -u "c:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\co4.1.py"
enter length of the rectangle1:10
enter breadth of the rectangle1:15
Area: 150
perimeter: 50
enter length of the rectangle2:50
enter breadth of the rectangle2:15
Area: 750
perimeter: 130
second rectangle has larger area

PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes>
```

2. CREATE A BANK ACCOUNT NUMBER, NAME, TYPE OF ACCOUNT AND ACCOUNT BALANCE. WRITE CONSTRUCTOR AND METHODS TO DEPOSIT AT THE BANK AND WITHDRAW AN AMOUNT FROM THE BANK.

SOURCE CODE:-

```
class BANK:
    def
__init__(self,account_number,name,type_of_account,balance):
        self.account_number = account_number
        self.name = name
        self.type_of_account = type_of_account
        self.balance = balance
    def deposit(self,amount):
        self.balance = self.balance+amount
        return self.balance
    def withdraw(self,amount):
```

```
        if(amount>self.balance):
            print("insufficient balance\n")
        else:
            self.balance = (self.balance-amount)
        return self.balance
#-------------------------------------------------------------
a = input("enter your account number: ")
n = input("enter your name: ")
t = input("enter the type of your account: ")
b = int(input("enter your current balance: "))
person1 = BANK(a,n,t,b)
val = input("1:Deposit\n2:Withdraw\n")
if (int(val)==1):
    amt = int(input("amount to deposit:-"))
    Bal = person1.deposit(amt)
elif(int(val)==2):
    amt = int(input("amount to withdraw:-"))
    Bal = person1.withdraw(amt)
else:
    print("invalid action\n")

print("current balance is: ",Bal)
```

OUTPUT:-

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python -u "c:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\co4.2.py"
enter your account number: 1100070413
enter your name: farzeena p a
enter the type of your account: fdrl knjr
enter your current balance: 5000
1:Deposit
2:Withdraw
1
amount to deposit:-5000
current balance is:  10000
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python -u "c:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\co4.2.py"
enter your account number: 1100070414
enter your name: haezal
enter the type of your account: fdrl kldy
enter your current balance: 100000
1:Deposit
2:Withdraw
2
amount to withdraw:-10000
current balance is:  90000
```

3. CREATE A CLASS RECTANGLE WITH PRIVATE ATTRIBUTES LENGTH AND WIDTH.

:-Overload '<' operator to compare the area of two rectangles

SOURCE CODE:-

```
class Rectangle:
    def __init__(self,l,b):
        self.__length=l
        self.__width=b
    def __lt__(self,ob):
        if((self.__length*self.__width)<(ob.__length *
ob.__width)):
            return True
        else:
            return False
#-------------------------------------------------------------
x1 = int(input("enter length of r1: \n"))
y1 = int(input("enter the breadth of r1: \n"))
x2 = int(input("enter length of r2: \n"))
y2 = int(input("enter the breadth of r2: \n"))
r1=Rectangle(x1,y1)
r2=Rectangle(x2,y2)
if(r1<r2):
    print("Area of r1<area of r2")
elif(r2<r1):
    print("Area of r2<area of r1")
else:
    print("Area of r1=area of r2")
```

OUTPUT:-

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python -u "c:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\co4.3.py"
enter length of r1:
10
enter the breadth of r1:
10
enter length of r2:
20
enter the breadth of r2:
1
Area of r2<area of r1
```

4. CREATE A CLASS TIME WITH PRIVATE ATTRIBUTES HOUR, MINUTE AND SECOND. OVERLOAD '+' OPERATOR TO FIND SUM OF TWO TIMES

SOURCE CODE:-

```
class Time:
    def __init__(self,h,m,s):
        self.__hour=h
        self.__minute=m
        self.__second=s
    def __add__(self,ob):
        hour=self.__hour+ob.__hour
        minute=self.__minute+ob.__minute
        second=self.__second+ob.__second
        t=Time(hour,minute,second)
        return t
    def print_it(self):
        print("Hour :",self.__hour)
        print("Minute :",self.__minute)
        print("Second :",self.__second)
t1=Time(10,10,10)
t2=Time(20,20,20)
t3=t1+t2
t3.print_it()
```

OUTPUT:-

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python -u "c:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\co4.4.py"
Hour : 30
Minute : 30
Second : 30
```

5. CREATE A CLASS PUBLISHER(NAME). DERIVE CLASS BOOK FROM PUBLISHER WITH ATTRIBUTES TITLE AND AUTHOR. DERIVE CLASS PYTHON FROM BOOK WITH ATTRIBUTES PRICE AND NO_OF_PAGES.

:-Write a program that displays information about a Python book. Use base class constructor invocation and method overreading

SOURCE CODE:-

```
class Publisher:
    def __init__(self,name):
        self.name = name
class Book(Publisher):
    def __init__(self, name,title,author):
        super().__init__(name)
        self.title = title
        self.author = author
    def print_fn(self):
        print("this fn is a member fn of class publisher \n")
class Python(Book):
    def __init__(self,name, title, author,price,nop):
        super().__init__(name,title,author)
        self.price = price
        self.nop = nop
    def print_fn(self):
        print("type of the book: ",self.name)
```

```
        print("title of the book: ",self.title)
        print("author of the book: ",self.author)
        print("price of the book; ",self.price)
        print("number of pages: ",self.nop)
#-------------------------------------------------------------
p1 = Python("TextBook", " The story of Sirius Black"  , "Albus
Dumbledore", "799", "1000")
p1.print_fn()
print("\n")
p2 = Python("Encyclopedia", "Around the mighty Hogwarts",
"Mad-eye Moody", "1000", "3000")
p2.print_fn()
print("\n")
```

OUTPUT:-

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python -u "c:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\test.py"
type of the book:  TextBook
title of the book:   The story of Sirius Black
author of the book:  Albus Dumbledore
price of the book;  799
number of pages:  1000


type of the book:  Encyclopedia
title of the book:  Around the mighty Hogwarts
author of the book:  Mad-eye Moody
price of the book;  1000
number of pages:  3000
```

COURSE OUTCOME 5

1. WRITE A PROGRAM TO READ A FILE LINE BY LINE AND STORE IT INTO A LIST

SOURCE CODE:-

```
file=open("text.txt","r")
lines=[]
for line in file:
    lines.append(line.strip())
print(lines)
```

>>text.txt

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range.

They are small, inexpensive, low-power, easy to use and don't wear out.

For that reason they are commonly found in appliances and gadgets used in homes or businesses.

They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.

OUTPUT:-

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python -u "c:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\co5.1.py"
['PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range.', "They are small, ine
xpensive, low-power, easy to use and don't wear out.", 'For that reason they are commonly found in appliances and gadgets used in homes or businesses.
', 'They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.']
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes>
```

2. WRITE A PYTHON PROGRAM TO READ EACH ROW FROM A GIVEN CSV FILE AND PRINT A LIST OF STRINGS

SOURCE CODE:-

```
import csv
with open("text.csv","r") as file:
    reader=csv.reader(file)
    for row in reader:
        print(row)
```

>>text.csv

```
Id,Name,Desig,Salary
01,Farzeena,CEO,200000
02,Haezal,CTO,100000
03,Hanna,CFO,100000
```

OUTPUT:-

```
PS C:\Users\LENOVO\Desktop\_The_ignited_kid_\codes> python -u "c:\Users\LENOVO\Desktop\_The_ignited_kid_\codes\co5.2.py"
['Id', 'Name', 'Desig', 'Salary']
['01', 'Farzeena', 'CEO', '200000']
['02', 'Haezal', 'CTO', '100000']
['03', 'Hanna', 'CFO', '100000']
```