FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)^{TM}

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

‘FOCUS ON EXCELLENCE’

20MCA131 PROGRAMMING LAB

………………………………………………………………………….

LABORATORY RECORD

Name: **MOHAMMED SHEKEEB .K**

Branch: **MASTER OF COMPUTER APPLICATIONS**

Semester: 1 **Branch: B** **Roll No: 31**

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)^{TM}

HORMIS NAGAR, MOOKKANNOOR

ANGAMALY-683577

‘FOCUS ON EXCELLENCE’

20MCA131 PROGRAMMING LAB

LABORATORY RECORD

Name: **MOHAMMED SHEKEEB .K**

Branch: **MASTER OF COMPUTER APPLICATIONS**

Semester: 1 **Roll No: 31**

University Exam Reg. No.:FIT22MCA-2090

CERTIFICATE

This is to certify that this is a Bonafide record of the Practical work done and submitted to APJ Abdul Kalam Technological University in partial fulfilment for the award of the Master of Computer Applications is a record of the original research work done by **MOHAMMED SHEKEEB .K** in the Laboratory of the Federal Institute of Science and Technology during the academic year 2022-2023.

Signature of Staff in Charge
Name:
Date:

Signature of HOD
Name:
Date:

Date of University Practical Examination: ……………………………………………

Signature of
Internal Examiner

Signature of
External Examiner

CONTENTS

COURSE OUTCOME 2

28		Program to construct the following pattern using nested loop *	39	
29		Program to generate all factors of a number	40	
30		Program with lambda functions to find area of square, rectangle, and triangle	41	
COURSE OUTCOME 3				
31		• Create a package graphics with modules rectangle, circle and a 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 find area and perimeter of figures by different importing statements.	42	
COURSE OUTCOME 4				
32		• Create a Rectangle class with attributes length and breadth and methods to find area and perimeter • Write a program to compare two Rectangle objects by their area.	44	
33		• Create a Bank account with members account_number, name, type_of_account, and balance • Write a constructor and methods to deposit an amount at the bank and withdraw an amount from the bank	45	
34		• Create a class Rectangle with private attributes length and width • Overload ‘< ‘operator to compare the area of two rectangles	46	
35		• Create a class Time with private attributes hour, minute, and second • Overload ‘+’ operator to find sum of 2 times	47	

36		• Create a class Publisher • 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 overriding	48	
COURSE OUTCOME 5				
37		Program to read a file line by line and store it into a list	49	
38		Program to copy odd lines of one file to another	50	
39		Program to read each row from a given .csv file and print a list of string	52	
40		Program to read specific columns of a given .csv file and print the content of the columns	53	
41		• Program to write a Python dictionary to a .csv file • After writing the .csv file, read the .csv file and display the content	54	

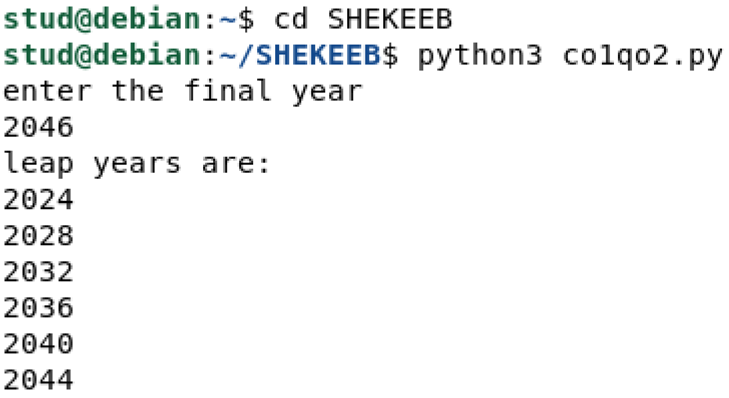
Experiment No. : **1**

Aim : Program to display future leap years from current year to a final year entered by user

Program Code:

```
year=2022
y=int(input("enter the final year\n"))
print("leap years are:")
for i in range(year,y):
       if((i%4==0 and i%100!=0) or i%400==0):
              print(i)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo2.py
enter the final year
2046
leap years are:
2024
2028
2032
2036
2040
2044
```

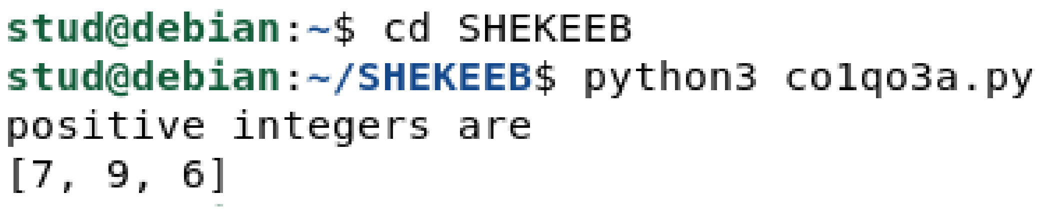
Experiment No. : **2a**

Aim : List Comprehension: Program to generate positive list of numbers from a given list of integers

Program Code:

```
l1=[7,-8,9,-3,6]
l2=[]
x=len(l1)
print("positive integers are")
for i in range(0,x):
        if(l1[i]>0):
                l2.append(l1[i])
print(l2)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo3a.py
positive integers are
[7, 9, 6]
```

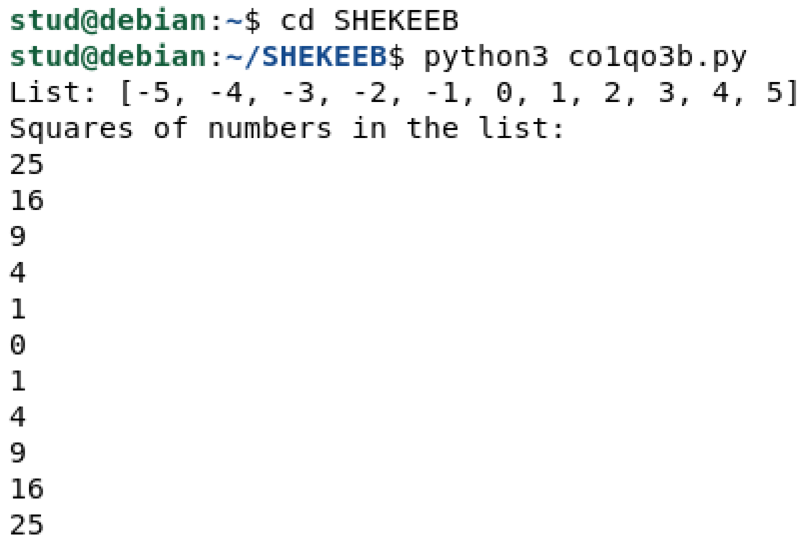
Experiment No. : **2b**

Aim : List Comprehension: Program to find the square of N numbers

Program Code:

```
a=[-5,-4,-3,-2,-1,0,1,2,3,4,5]
print("List:",a)
print("Squares of numbers in the list:")
for i in range(0,11):
       print(a[i]*a[i])
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo3b.py
List: [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
Squares of numbers in the list:
25
16
9
4
1
0
1
4
9
16
25
```

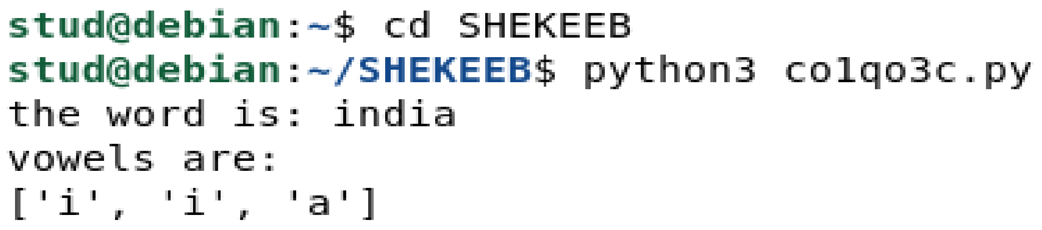
Experiment No. : **2c**

Aim : List Comprehension: Program to form a list of vowels selected from a given word

Program Code:

```
s1="india"
l1=['a','e','i','o','u','A','E','I','O','U']
l2=[]
print("the word is:",s1)
print("vowels are:")
for i in s1:
        if i in l1:
                l2.append(i)
print(l2)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo3c.py
the word is: india
vowels are:
['i', 'i', 'a']
```

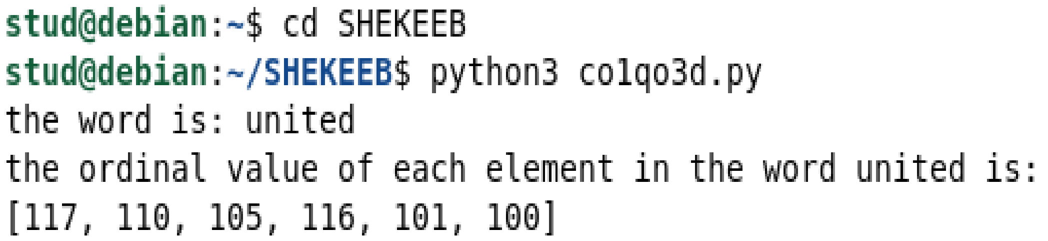
Experiment No. : **2d**

Aim : List Comprehension: Program to list ordinal value of each element of a word

Program Code:

```
s1="united"
l1=[]
print("the word is:",s1)
print("the ordinal value of each element in the word",s1,"is:")
for i in s1:
       x=ord(i)
       l1.append(x)
print(l1)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo3d.py
the word is: united
the ordinal value of each element in the word united is:
[117, 110, 105, 116, 101, 100]
```

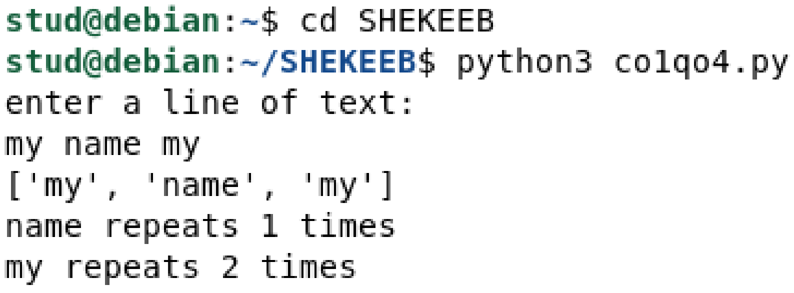
Experiment No. : 3

Aim : Program to count the occurrences of each word in a line of text

Program Code:

```
n=(str(input("enter a line of text:\n")))
l1=n.split()
print(l1)
l2=set(l1)
for i in l2:
        c=0
        for j in l1:
                if(i==j):
                        c=c+1
        print(i,"repeats",c,"times")
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo4.py
enter a line of text:
my name my
['my', 'name', 'my']
name repeats 1 times
my repeats 2 times
```

Experiment No. : **4**

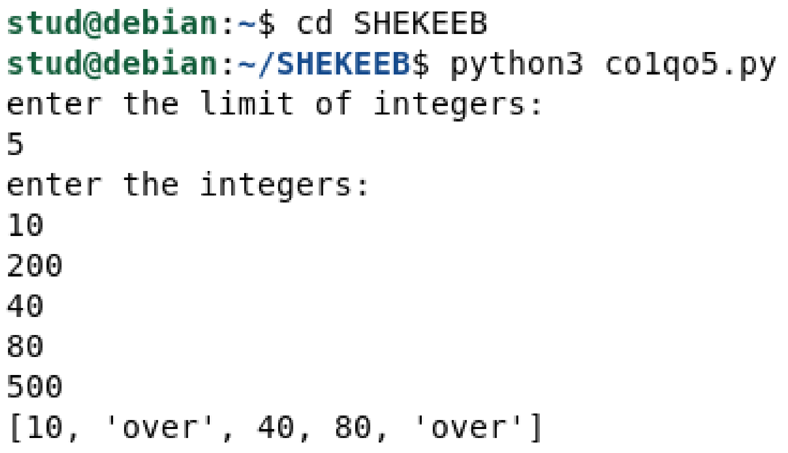
Aim :

- Program to prompt the user for a list of integers
- For all values greater than 100, store ‘over ’instead

Program Code:

```
n=(int(input("enter the limit of integers:\n")))
l1=[]
print("enter the integers:")
for i in range(0,n):
        x=int(input())
        if(x<100):
                l1.append(x)
        else:
                l1.append("over")
print(l1)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo5.py
enter the limit of integers:
5
enter the integers:
10
200
40
80
500
[10, 'over', 40, 80, 'over']
```

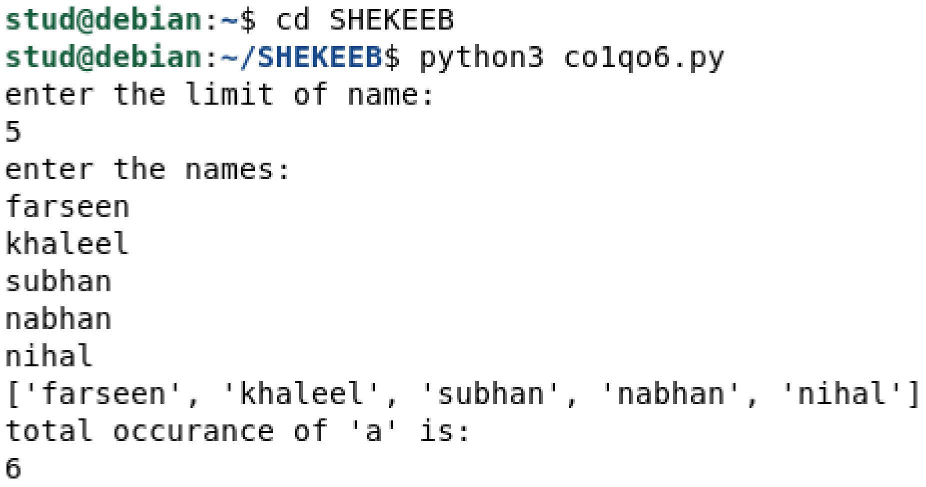
Experiment No. : **5**

Aim :
- Program to store a list of first names
- Count the occurrences of ‘a ’within the list

Program Code:

```
n=(int(input("enter the limit of name:\n")))
l1=[]
print("enter the names:")
for i in range(0,n):
        x=input()
        l1.append(x)
print(l1)
cnt=0
for i in l1:
        cnt=cnt+i.count("a")
print("total occurance of 'a' is:")
print(cnt)
```

Output:



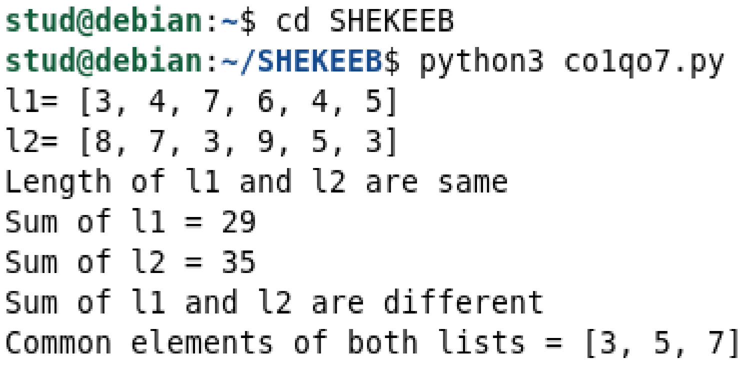
```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo6.py
enter the limit of name:
5
enter the names:
farseen
khaleel
subhan
nabhan
nihal
['farseen', 'khaleel', 'subhan', 'nabhan', 'nihal']
total occurance of 'a' is:
6
```

Experiment No.	**:**	**6**
Aim	**:**	Program to enter two lists of integers. Check whether: a) the lists are of the same length b) the lists sum to the same value c) any value occurs in both the lists

Program Code:

```
l1=[3,4,7,6,4,5]
l2=[8,7,3,9,5,3]
print("l1=",l1)
print("l2=",l2)
x=len(l1)
y=len(l2)
if(x==y):
        print("Length of l1 and l2 are same")
else:
        print("Length of l1 and l2 are different")
suml1=sum(l1)
suml2=sum(l2)
print("Sum of l1 =",suml1)
print("Sum of l2 =",suml2)
if(suml1==suml2):
        print("Sum of l1 and l2 are same")
else:
        print("Sum of l1 and l2 are different")
l3=set(l1)
l4=set(l2)
l5=[]
for i in l3:
        for j in l4:
                if(i==j):
                        l5.append(i)
x=len(l5)
if(x==0):
        print("Common elements are not occur in both lists")
else:
        print("Common elements of both lists =",l5)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo7.py
l1= [3, 4, 7, 6, 4, 5]
l2= [8, 7, 3, 9, 5, 3]
Length of l1 and l2 are same
Sum of l1 = 29
Sum of l2 = 35
Sum of l1 and l2 are different
Common elements of both lists = [3, 5, 7]
```

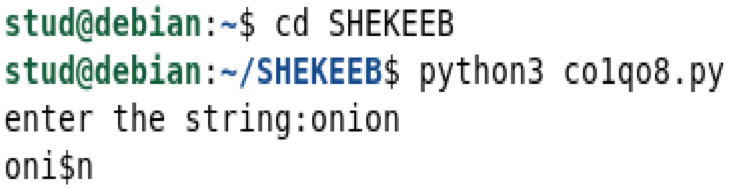
Experiment No. : **7**

Aim : Program to get a string from an input string where all occurrences of first character is replaced with ‘$’ except the first character

Program Code:

```
x=input("enter the string:")
y=x[0]
l=len(x)
for i in range(1,l):
        if(x[i]==x[0]):
                y=y+"$"
        else:
                y=y+x[i]
print(y)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo8.py
enter the string:onion
oni$n
```

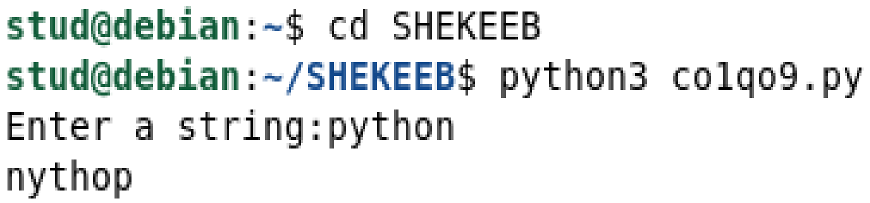
Experiment No. : **8**

Aim : Program to create a string from the given string where the first and last characters are exchanged

Program Code:

```
s=input("Enter a string:")
x=s[0]
y=s[-1]
z=s[1:-1]
s2=y+z+x
print(s2)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1qo9.py
Enter a string:python
nythop
```

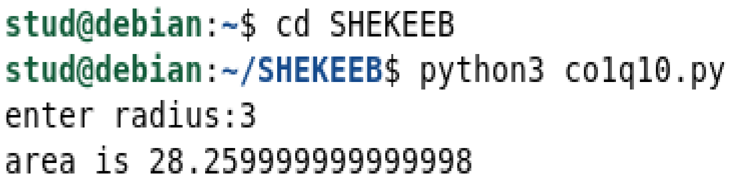
Experiment No. : **9**

Aim : Program to accept radius from the user and find the area of a circle

Program Code:

```
r=(int(input("enter radius:")))
a=3.14*r*r
print("area is",a)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q10.py
enter radius:3
area is 28.259999999999998
```

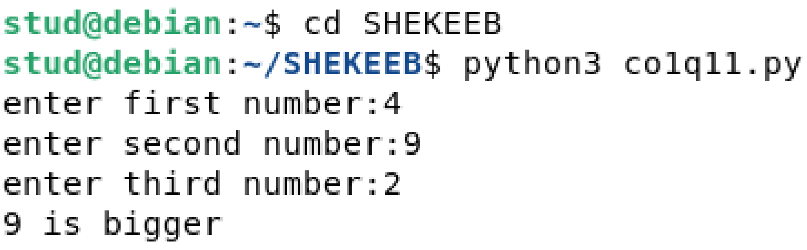
Experiment No. : **10**

Aim : Program to find the biggest of three numbers entered

Program Code:

```
a=(int(input("enter first number:")))
b=(int(input("enter second number:")))
c=(int(input("enter third number:")))
if(a>b):
        if(a>c):
                print(a,"is bigger")
        else:
                print(c,"is bigger")
elif(b>c):
        print(b,"is bigger")
else:
        print(c,"is bigger")
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q11.py
enter first number:4
enter second number:9
enter third number:2
9 is bigger
```

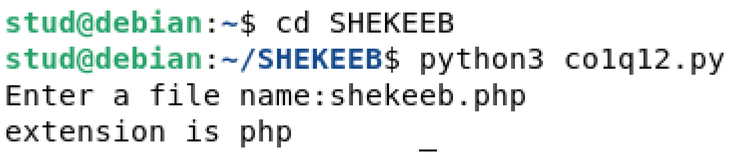
Experiment No. : **11**

Aim : Program to accept a file name from user and print its extension

Program Code:

```
fname=input("Enter a file name:")
c=fname.split(".")
print("extension is",c[-1])
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q12.py
Enter a file name:shekeeb.php
extension is php
```

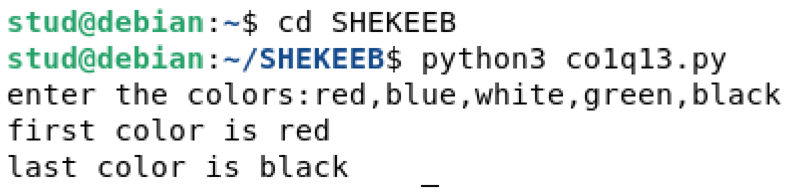
Experiment No. : **12**

Aim : • Program to create a list of colours from comma-separated colour names entered by user
• Display the first and last colours

Program Code:

```
color=input("enter the colors:")
c=color.split(",")
print("first color is",c[0])
print("last color is",c[-1])
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q13.py
enter the colors:red,blue,white,green,black
first color is red
last color is black
```

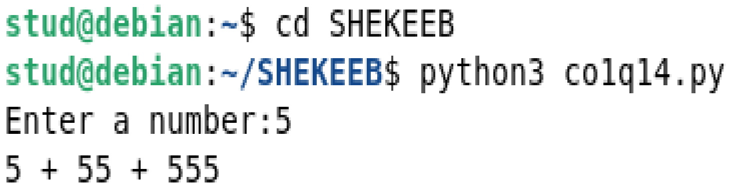
Experiment No. : **13**

Aim : Program to accept an integer n and compute n+nn+nnn

Program Code:

```
n=int(input("Enter a number:"))
print(n,"+",(n*10)+n,"+",(n*100)+(n*10)+n,)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q14.py
Enter a number:5
5 + 55 + 555
```

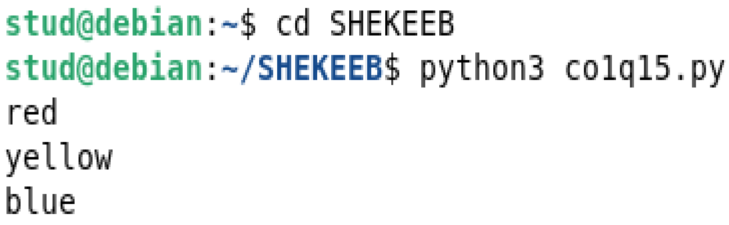
Experiment No. : **14**

Aim : Program to print out all colours from color-list1 not contained in color-list2

Program Code:

```
l1=['red','green','yellow','blue']
l2=['orange','black','white','green']
for i in l1:
        if i not in l2:
                print(i
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 colq15.py
red
yellow
blue
```

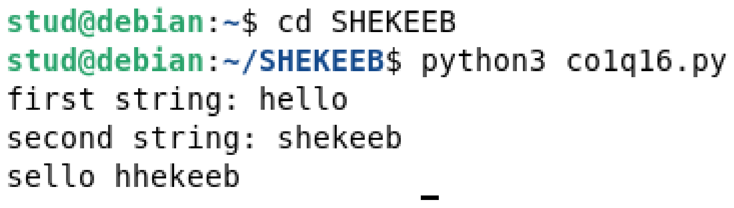
Experiment No. : **15**

Aim : Program to create a single string separated with space form two string by swapping the character at position 1

Program Code:

```
x=input("first string: ")
y=input("second string: ")
z=y[0]+x[1:]+" "+x[0]+y[1:]
print(z)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q16.py
first string: hello
second string: shekeeb
sello hhekeeb
```

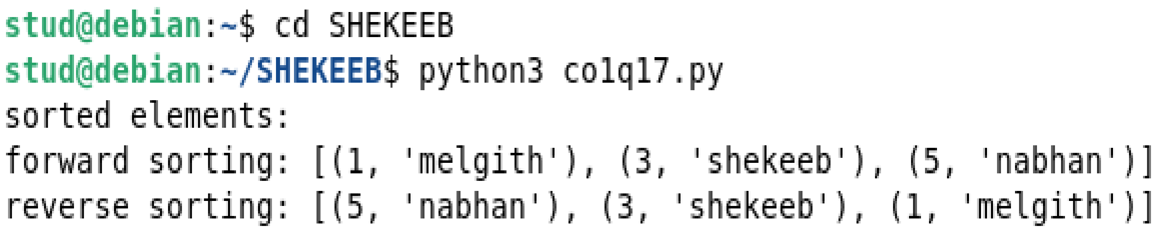
Experiment No. : **16**

Aim : Program to sort a dictionary in ascending and descending order

Program Code:

```
a={1:"melgith",3:"shekeeb",5:"nabhan",}
b={6:"farseen",7:"nihal",9:"subhan"}
x=sorted(a.items())
y=sorted(a.items(),reverse=True)
print("sorted elements:")
print("forward sorting:",x)
print("reverse sorting:",y)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q17.py
sorted elements:
forward sorting: [(1, 'melgith'), (3, 'shekeeb'), (5, 'nabhan')]
reverse sorting: [(5, 'nabhan'), (3, 'shekeeb'), (1, 'melgith')]
```

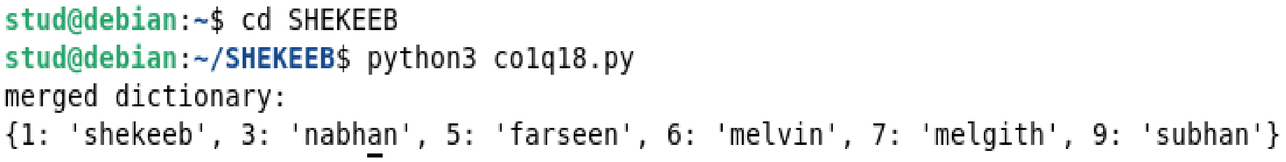
Experiment No. : **17**

Aim : Program to merge two dictionaries

Program Code:

```
a={1:"shekeeb",3:"nabhan",5:"farseen"}
b={6:"melvin",7:"melgith",9:"subhan"}
print("merged dictionary:")
print(a|b)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q18.py
merged dictionary:
{1: 'shekeeb', 3: 'nabhan', 5: 'farseen', 6: 'melvin', 7: 'melgith', 9: 'subhan'}
```

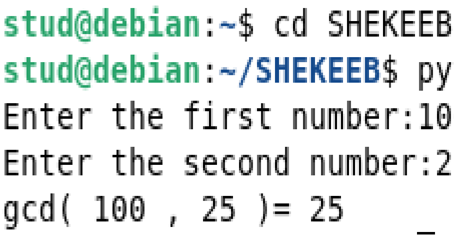
Experiment No. : **18**

Aim : Program to find the Greatest Common Divisor (GCD) of two numbers

Program Code:

```
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
import math
g=math.gcd(a,b)
print("gcd(",a,",",b,")=",g)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co1q19.py
Enter the first number:100
Enter the second number:25
gcd( 100 , 25 )= 25
```

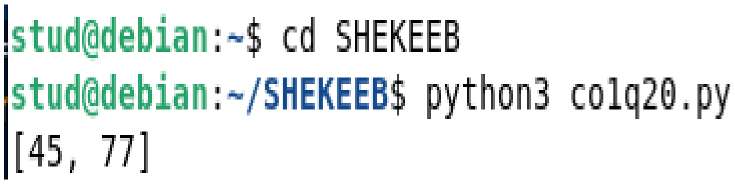
Experiment No. : **19**

Aim : Program to create a list removing even numbers from a list of integers

Program Code:

```
l1=[45,60,56,77,8]
l2=[]
for i in l1:
        if(i%2==0):
                l1.remove(i)
        else:
                l2.append(i)
print(l2)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 colq20.py
[45, 77]
```

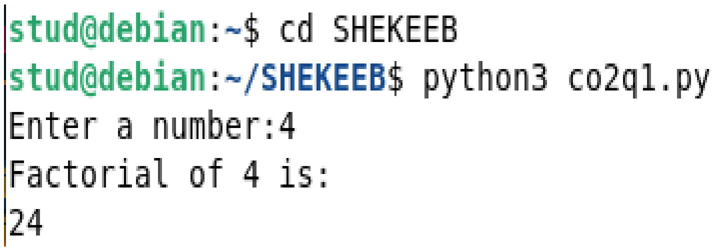
Experiment No. : **20**

Aim : Program to find the factorial of a number

Program Code:

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

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q1.py
Enter a number:4
Factorial of 4 is:
24
```

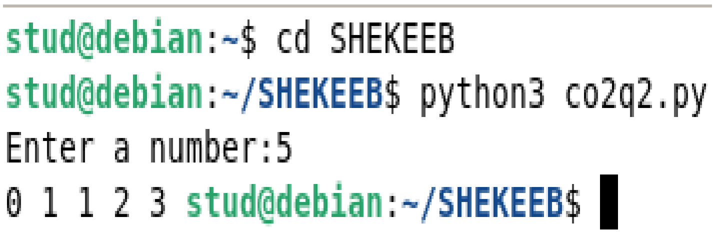
Experiment No. : **21**

Aim : Program to generate Fibonacci series of N terms

Program Code:

```
x=0
y=1
c=0
n=int(input("Enter a number:"))
for i in range(0,n):
        print(c,end=' ')
        x=y
        y=c
        c=x+y
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q2.py
Enter a number:5
0 1 1 2 3 stud@debian:~/SHEKEEB$
```

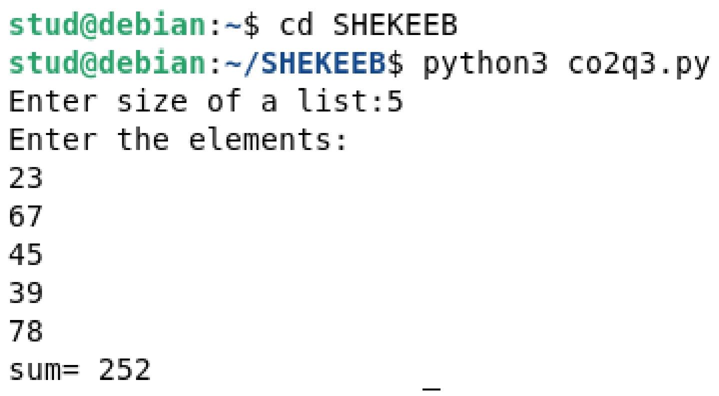
Experiment No. : 22

Aim : Program to find the sum of all items in a list

Program Code:

```
l1=[]
n=int(input("Enter size of a list:"))
print("Enter the elements:")
for i in range(1,n+1):
       m=int(input())
       l1.append(m)
s=sum(l1)
print("sum=",s)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q3.py
Enter size of a list:5
Enter the elements:
23
67
45
39
78
sum= 252
```

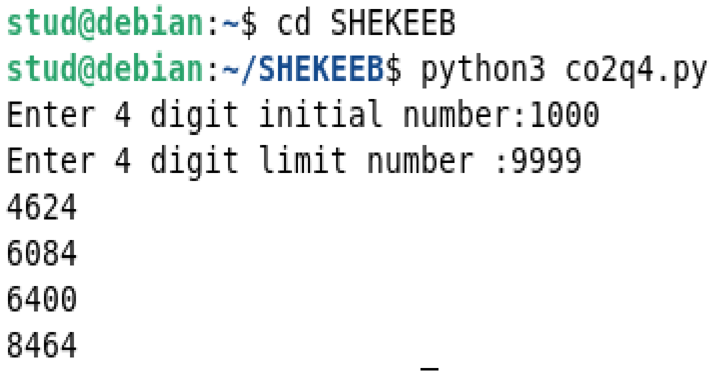
Experiment No. : **23**

Aim : Program to generate a list of four digit numbers in a given range with all their digits even and the numbers being perfect squares

Program Code:

```
import math
a=int(input("Enter 4 digit initial number:"))
b=int(input("Enter 4 digit limit number :"))
for i in range (a,b+1):
        c=i//1000
        d=i%1000
        e=d//100
        f=d%100
        g=f//10
        h=f%10
        s=math.sqrt(i)
        if(c%2==0 and e%2==0 and g%2==0 and h%2==0 and float.is_integer(s) and 1000<=i<10000):
                print(i)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q4.py
Enter 4 digit initial number:1000
Enter 4 digit limit number :9999
4624
6084
6400
8464
```

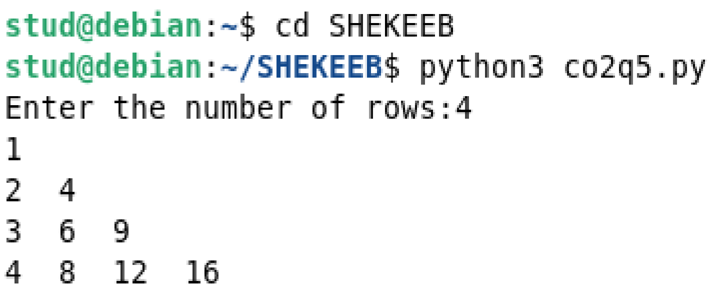
Experiment No. : **24**

Aim : Program to display the given pyramid with step number accepted from the user
e.g. N=4
1
2 4
3 6 9
4 8 12 16

Program Code:

```
n=int(input("Enter the number of rows:"))
for i in range(1,n+1):
        for j in range(1,i+1):
                print(i*j,"",end=" ")
        print("")
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q5.py
Enter the number of rows:4
1
2  4
3  6  9
4  8  12  16
```

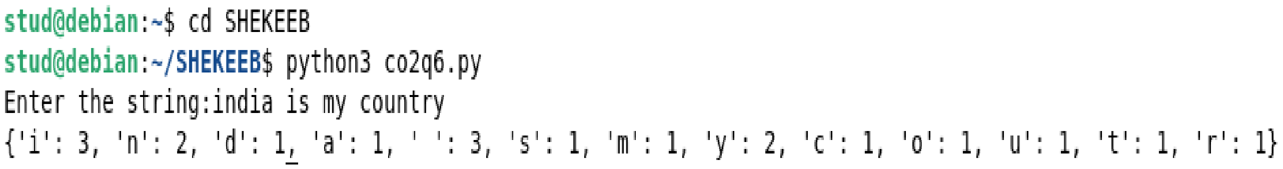
Experiment No. : **25**

Aim : Program to count the number of characters (character frequency) in a string

Program Code:

```
s=input("Enter the string:")
c=dict()
for i in s:
        if i in c:
                c[i]+=1
        else:
                c[i]=1
print(c)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q6.py
Enter the string:india is my country
{'i': 3, 'n': 2, 'd': 1, 'a': 1, ' ': 3, 's': 1, 'm': 1, 'y': 2, 'c': 1, 'o': 1, 'u': 1, 't': 1, 'r': 1}
```

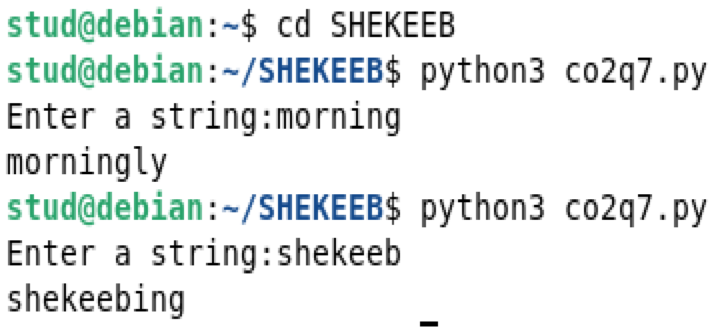
Experiment No. : **26**

Aim : Program to add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’

Program Code:

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

Output:



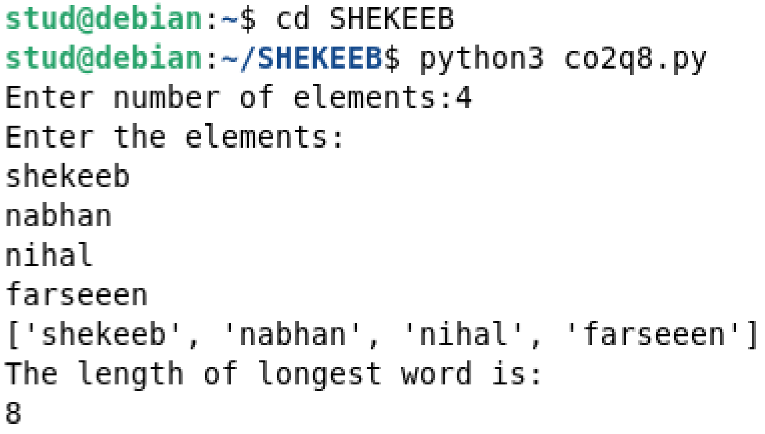
```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q7.py
Enter a string:morning
morningly
stud@debian:~/SHEKEEB$ python3 co2q7.py
Enter a string:shekeeb
shekeebing
```

Experiment No.	**:**	**27**
Aim	**:**	Program to accept a list of words and return the length of the longest word

Program Code:

```
n=int(input("Enter number of elements:"))
l1=[]
l2=[]
print("Enter the elements:")
for i in range(0,n):
        m=input()
        l1.append(m)
        l2.append(len(m))
print(l1)
print("The length of longest word is:")
print(max(l2))
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q8.py
Enter number of elements:4
Enter the elements:
shekeeb
nabhan
nihal
farseeen
['shekeeb', 'nabhan', 'nihal', 'farseeen']
The length of longest word is:
8
```

Experiment No. : **28**

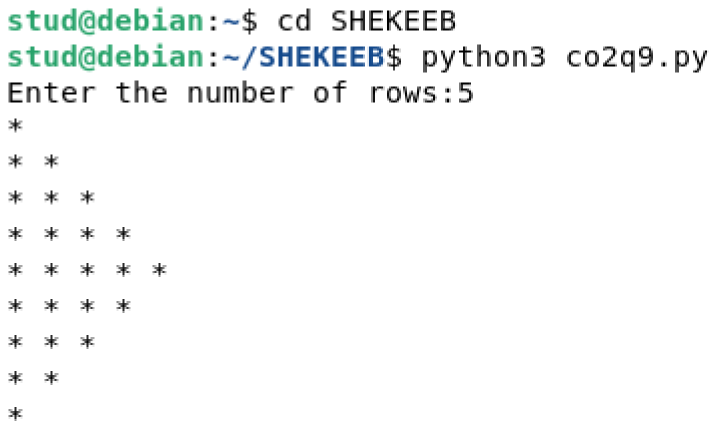
Aim : Program to construct the following pattern using nested loop

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

Program Code:

```
n=int(input("Enter the number of rows:"))
for i in range(1,n+1):
        for j in range(1,i+1):
                print("*","",end='')
        print("")
for i in range(n-1,0,-1):
        for j in range(1,i+1):
                print("*","",end='')
        print("")
```

Output:



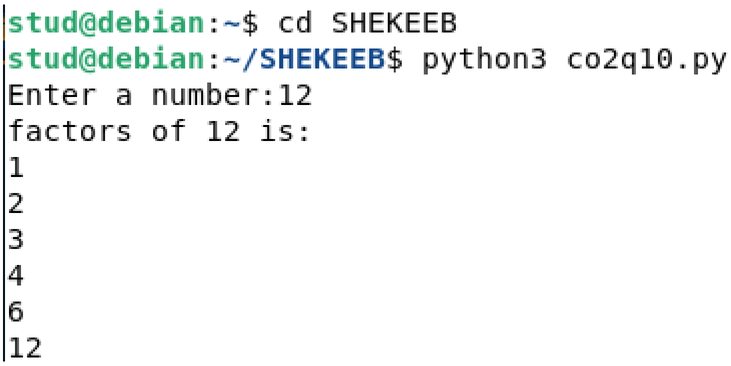
```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q9.py
Enter the number of rows:5
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Experiment No.	**:**	**29**
Aim	**:**	Program to generate all factors of a number

Program Code:

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

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co2q10.py
Enter a number:12
factors of 12 is:
1
2
3
4
6
12
```

Experiment No.	**:**	**30**
Aim	**:**	Program with lambda functions to find area of square, rectangle, and triangle

Program Code:

```
print("area of square:")
x=lambda a:a*a
print(x(4))

print("area of rectangle:")
x=lambda a,b:a*b
print(x(2,3))

print("area of triangle:")
x=lambda a,b:.5*a*b
print(x(2,3))
```

Output:

```
area of square:
16
area of rectangle:
6
area of triangle:
3.0
```

Experiment No. : **31**

Aim :

- Create a package graphics with modules rectangle, circle and a sub-poackage 3D-graphics with modules cuboid and sphere.
- Include methods to find area and perimeter of respective figures in each module.
- Write programs that find area and perimeter of figures by different importing statements.

Program Code:

Co3.py

```
from graphics.circle import*
from graphics.rectangle import*
from graphics.graphics3d.sphere import*
from graphics.graphics3d.cuboid import*
Circle(r)
Rectangle(l,b)
Cuboid(a)
Sphere(r)
```

Circle.py

```
def Circle(r):
	r=int(input("Enter the radius of the cirle:"))
	print("Circle Area:",3.14*r*r)
	print("Circle Circumfrence:",2*3.14*r)
```

Rectangle.py

```
def Rectangle(l,b):
	l=int(input("Enter the length of the Rectangle:"))
	b=int(input("Enter the Breadth of the Rectangle:"))
	print("Rectangle Area:",l*b)
	print("Rectangle perimeter:",2*(l+b))
```

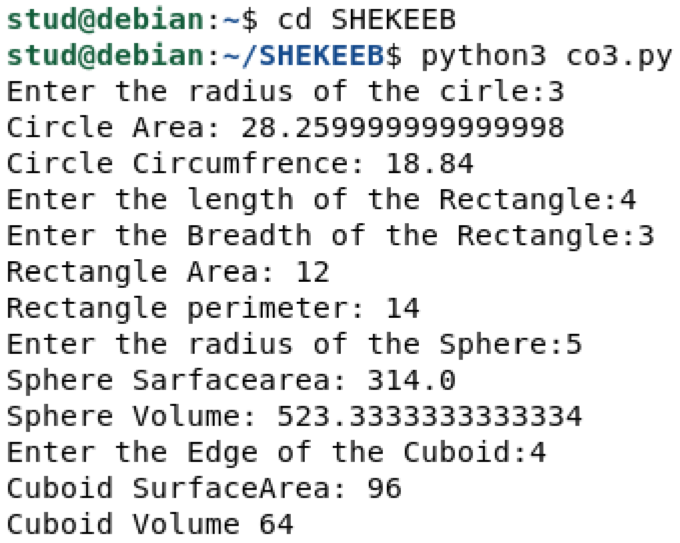
Cuboid.py

```
def  Cuboid(a):
	s=int(input("Enter the radius of the Sphere:"))
	print("Cuboid SurfaceArea:",6*a*a)
	print("Cuboid Volume",a*a*a)
```

<u>Sphere.py</u>

```
def Sphere(r):
        e=int(input("Enter the Edge of the Cuboid:"))
        print("Sphere Sarfacearea:",4*3.14*r*r)
        print("Sphere Volume:",(4/3)*3.14*r*r*r)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co3.py
Enter the radius of the cirle:3
Circle Area: 28.259999999999998
Circle Circumfrence: 18.84
Enter the length of the Rectangle:4
Enter the Breadth of the Rectangle:3
Rectangle Area: 12
Rectangle perimeter: 14
Enter the radius of the Sphere:5
Sphere Sarfacearea: 314.0
Sphere Volume: 523.3333333333334
Enter the Edge of the Cuboid:4
Cuboid SurfaceArea: 96
Cuboid Volume 64
```

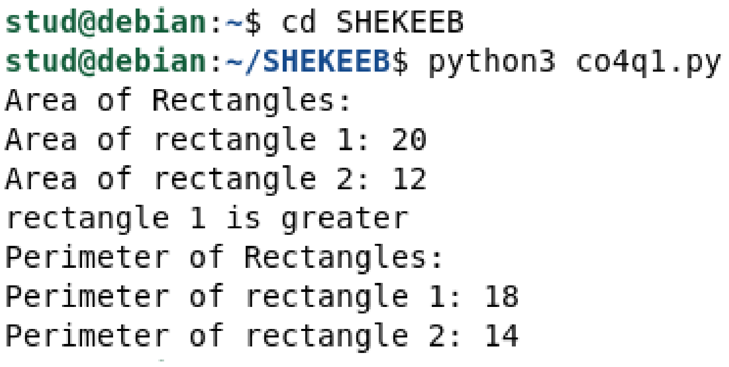
Experiment No. : **32**

Aim : • Create a Rectangle class with attributes length and breadth and methods to find area and perimeter
• Write a program to compare two Rectangle objects by their area.

Program Code:

```
class Rectangle:
        def __init__(self,length,width):
                self.length=length
                self.width=width
        def area(self):
                area=self.length*self.width
                return(area)
        def peri(self):
                peri=2*(self.length+self.width)
                return(peri)
obj1=Rectangle(4,5)
obj2=Rectangle(4,3)
print("Area of Rectangles:")
area1=obj1.area()
area2=obj2.area()
print("Area of rectangle 1:",area1)
print("Area of rectangle 2:",area2)
if (area1==area2):
        print("Areas are same")
elif (area1>area2):
        print("rectangle 1 is greater")
else:
        print("rectangle 2 is greater")
print("Perimeter of Rectangles:")
peri1=obj1.peri()
peri2=obj2.peri()
print("Perimeter of rectangle 1:",peri1)
print("Perimeter of rectangle 2:",peri2)
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co4q1.py
Area of Rectangles:
Area of rectangle 1: 20
Area of rectangle 2: 12
rectangle 1 is greater
Perimeter of Rectangles:
Perimeter of rectangle 1: 18
Perimeter of rectangle 2: 14
```

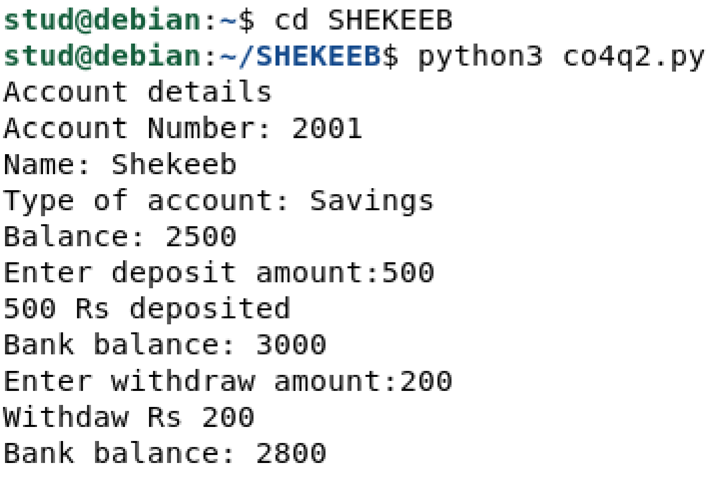
Experiment No. : **33**

Aim : • Create a Bank account with members account_number, name, type_of_account, and balance
• Write a constructor and methods to deposit an amount at the bank and withdraw an amount from the bank

Program Code:

```
class Bank:
	def __init__(self,acno,name,tyofac,bal):
		self.acno=acno
		self.name=name
		self.tyofac=tyofac
		self.bal=bal
	def display(self):
		print("Account details")
		print("Account Number:",self.acno,"")
		print("Name:",self.name,"")
		print("Type of account:",self.tyofac,"")
		print("Balance:",self.bal,"")
	def deposite(self):
		dep=int(input("Enter deposit amount:"))
		self.bal=self.bal+dep
		print(dep,"Rs deposited\nBank balance:",self.bal)
	def withdraw(self):
		draw=int(input("Enter withdraw amount:"))
		if(draw>self.bal):
			print("Balance is lessthan withdraw amount")
		else:
			self.bal=self.bal-draw
			print("Withdaw Rs",draw,"\nBank balance:",self.bal)
user1=Bank(2001,'Shekeeb','Savings',2500)
user1.display()
user1.deposite()
user1.withdraw()
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co4q2.py
Account details
Account Number: 2001
Name: Shekeeb
Type of account: Savings
Balance: 2500
Enter deposit amount:500
500 Rs deposited
Bank balance: 3000
Enter withdraw amount:200
Withdaw Rs 200
Bank balance: 2800
```

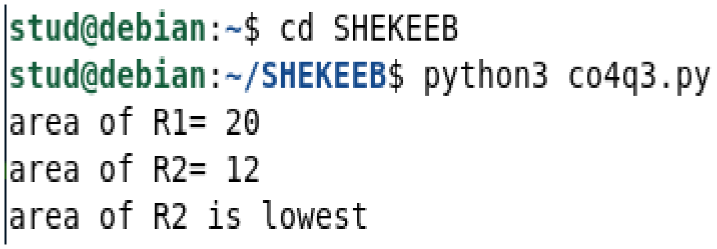
Experiment No. : **34**

Aim : • Create a class Rectangle with private attributes length and width
• Overload ‘< ‘operator to compare the area of two rectangles

Program Code:

```
class Rectangle:
        def __init__(self,length,width):
                self.__length=length
                self.__width=width
        def area(self):
                area=self.__length*self.__width
                return(area)
        def __lt__(self,t):
                if(self.__length * self.__width < t.__length * t.__width):
                        print("area of R1 is lowest")
                else:
                        print("area of R2 is lowest")
obj1=Rectangle(4,5)
obj2=Rectangle(4,3)
area1=obj1.area()
print("area of R1=",area1)
area2=obj2.area()
print("area of R2=",area2)
obj1<obj2
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co4q3.py
area of R1= 20
area of R2= 12
area of R2 is lowest
```

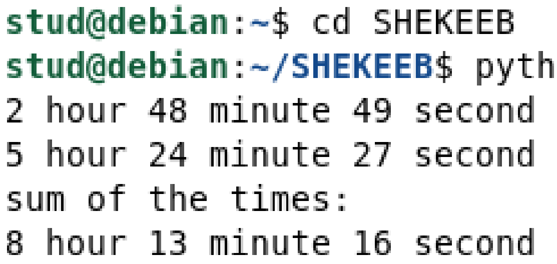
Experiment No. : **35**

Aim : • Create a class Time with private attributes hour, minute, and second
• Overload '+' operator to find sum of 2 times

Program Code:

```
class Time:
       def __init__(self,hour,minute,second):
              self.__hour=hour
              self.__minute=minute
              self.__second=second
       def display(self):

       print(self.__hour,"hour",self.__minute,"minute",self.__second,"second
")
       def __add__(self, other):
              h1 = self.__hour + other.__hour
              m1 = self.__minute + other.__minute
              s1 = self.__second + other.__second
              if(m1>60):
                     m1=m1-60
                     h1=h1+1
              if(s1>60):
                     s1=s1-60
                     m1=m1+1
              print(h1,"hour",m1,"minute",s1,"second")
t1=Time(2,48,49)
t2=Time(5,24,27)
t1.display()
t2.display()
print("sum of the times:")
t1 + t2
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co4q4.py
2 hour 48 minute 49 second
5 hour 24 minute 27 second
sum of the times:
8 hour 13 minute 16 second
```

Experiment No. : **36**

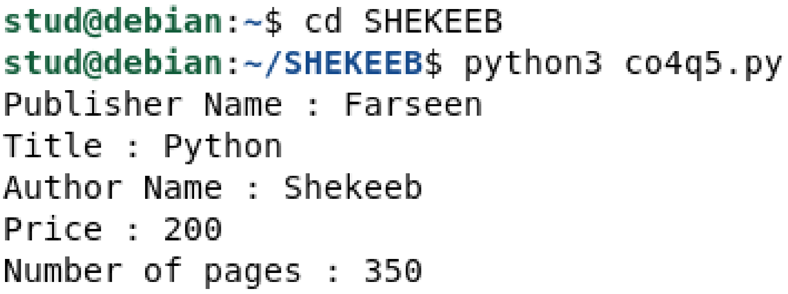
Aim :

- Create a class Publisher
- 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 overriding

Program Code:

```
class Publisher:
        def __init__(self,name):
                self.name=name
        def display(self):
                print("Name of Publisher is ",self.name)
class Book(Publisher):
        def __init__(self,name,title,author):
                super().__init__(name)
                self.title=title
                self.author=author
        def display(self):
                print("Publisher Name is",self.name)
                print("Title is ",self.title)
                print("Author Name is ",self.author)
class Python(Book):
        def __init__(self,name,title,author,price,noofpage):
                super().__init__(name,title,author)
                self.price=price
                self.noofpage=noofpage
        def display(self):
                print("Publisher Name :",self.name)
                print("Title :",self.title)
                print("Author Name :",self.author)
                print("Price :",self.price)
                print("Number of pages :",self.noofpage)
p1=Python("Farseen","Python","Shekeeb",200,350)
p1.display()
```

Output:



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co4q5.py
Publisher Name : Farseen
Title : Python
Author Name : Shekeeb
Price : 200
Number of pages : 350
```

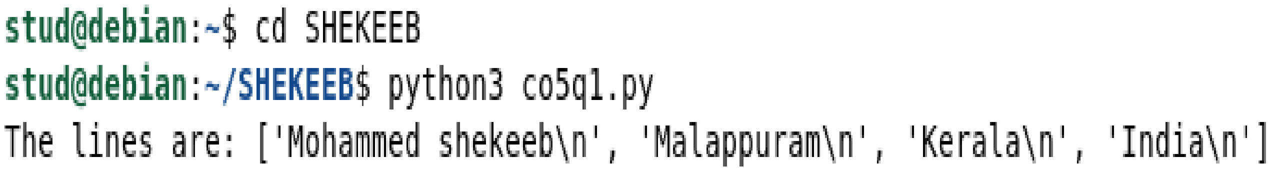
Experiment No. : **37**

Aim : Program to read a file line by line and store it into a list

Program Code:

```
filehandle=open('file.txt','r')
x=filehandle.readlines()
print("The lines are:",x)
filehandle.close()
```

Output:



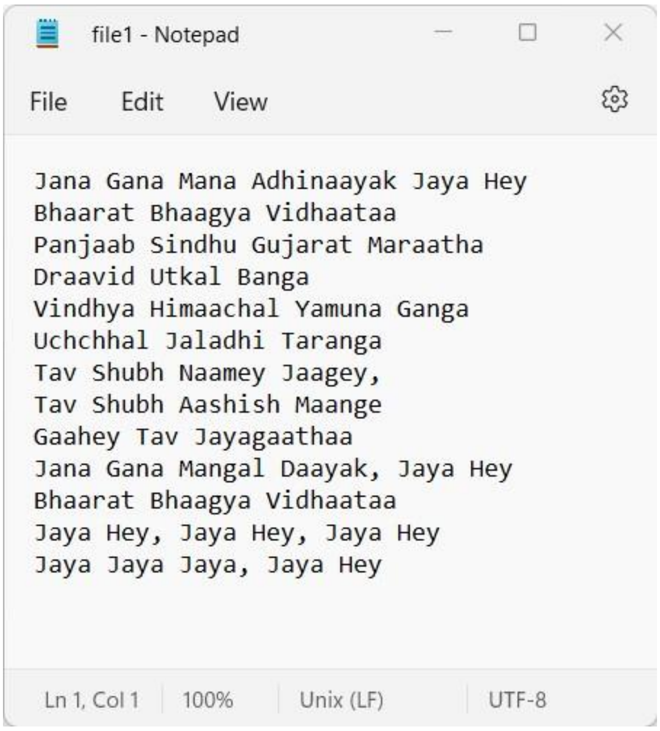
```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co5q1.py
The lines are: ['Mohammed shekeeb\n', 'Malappuram\n', 'Kerala\n', 'India\n']
```

Experiment No.	**:**	**38**
Aim	**:**	Program to copy odd lines of one file to another

Program Code:

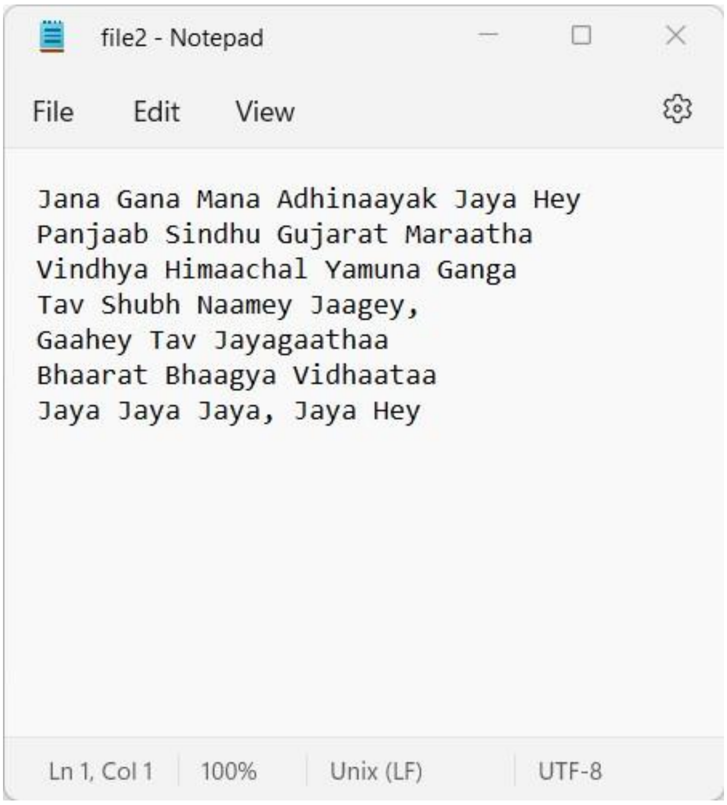
```
f1=open('file1.txt','r')
f2=open('file2.txt','w')
x=f1.readlines()
j=1
for i in range(0,len(x)):
        if(j%2!=0):
                         f2.write(x[i])
                         print(x[i])
               j=j+1

f1.close()
f2.close()
```

Output:



```
Jana Gana Mana Adhinaayak Jaya Hey
Bhaarat Bhaagya Vidhaataa
Panjaab Sindhu Gujarat Maraatha
Draavid Utkal Banga
Vindhya Himaachal Yamuna Ganga
Uchchhal Jaladhi Taranga
Tav Shubh Naamey Jaagey,
Tav Shubh Aashish Maange
Gaahey Tav Jayagaathaa
Jana Gana Mangal Daayak, Jaya Hey
Bhaarat Bhaagya Vidhaataa
Jaya Hey, Jaya Hey, Jaya Hey
Jaya Jaya Jaya, Jaya Hey
```

```
Jana Gana Mana Adhinaayak Jaya Hey

Panjaab Sindhu Gujarat Maraatha

Vindhya Himaachal Yamuna Ganga

Tav Shubh Naamey Jaagey,

Gaahey Tav Jayagaathaa

Bhaarat Bhaagya Vidhaataa

Jaya Jaya Jaya, Jaya Hey
```



```
Jana Gana Mana Adhinaayak Jaya Hey
Panjaab Sindhu Gujarat Maraatha
Vindhya Himaachal Yamuna Ganga
Tav Shubh Naamey Jaagey,
Gaahey Tav Jayagaathaa
Bhaarat Bhaagya Vidhaataa
Jaya Jaya Jaya, Jaya Hey
```

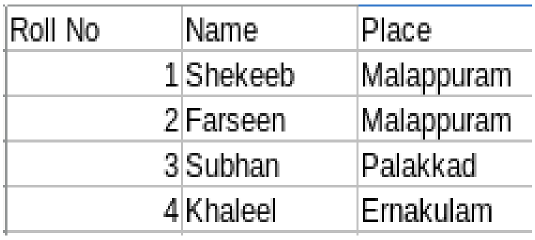
Experiment No. : **39**

Aim : Program to read each row from a given .csv file and print a list of string

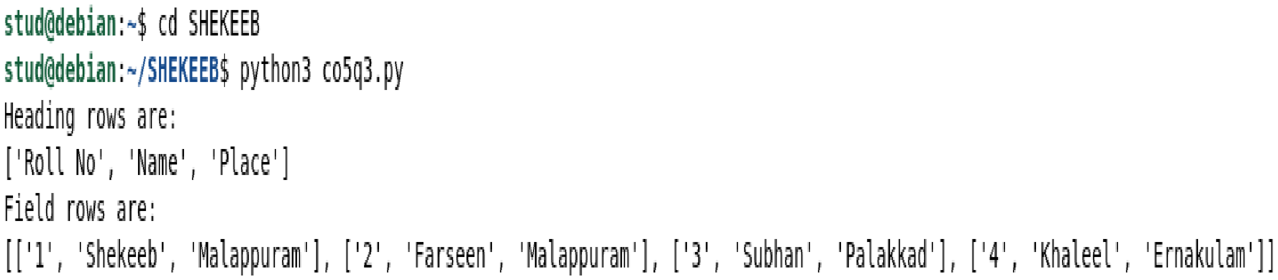
Program Code:

```
import csv
file = open('file.csv')
print("Heading rows are:")
csvreader = csv.reader(file)
header = []
header = next(csvreader)
print(header)
print("Field rows are:")
rows = []
for row in csvreader:
        rows.append(row)
print(rows)
```

Output:



Roll No	Name	Place
1	Shekeeb	Malappuram
2	Farseen	Malappuram
3	Subhan	Palakkad
4	Khaleel	Ernakulam



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co5q3.py
Heading rows are:
['Roll No', 'Name', 'Place']
Field rows are:
[['1', 'Shekeeb', 'Malappuram'], ['2', 'Farseen', 'Malappuram'], ['3', 'Subhan', 'Palakkad'], ['4', 'Khaleel', 'Ernakulam']]
```

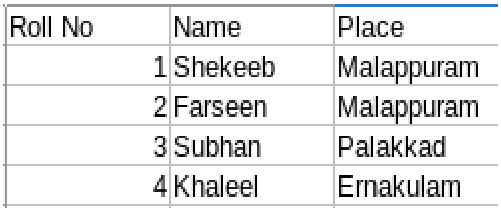
Experiment No. : **40**

Aim : Program to read specific columns of a given .csv file and print the content of the columns

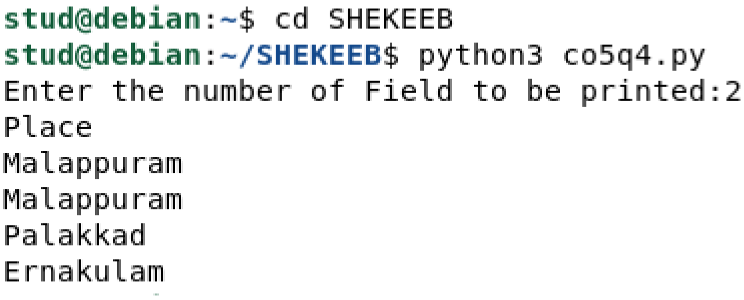
Program Code:

```
import csv
file = open('file.csv')
csvreader = csv.reader(file)
n=int(input("Enter the number of Field to be printed:"))
rows = []
for row in csvreader:
        rows.append(row)
for i in range(0,len(rows)):
        print(rows[i][n])
```

Output:



Roll No	Name	Place
1	Shekeeb	Malappuram
2	Farseen	Malappuram
3	Subhan	Palakkad
4	Khaleel	Ernakulam



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co5q4.py
Enter the number of Field to be printed:2
Place
Malappuram
Malappuram
Palakkad
Ernakulam
```

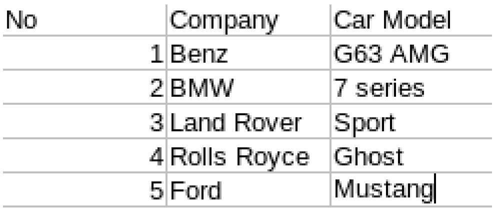
Experiment No. : **41**

Aim : • Program to write a Python dictionary to a .csv file
• After writing the .csv file, read the .csv file and display the content

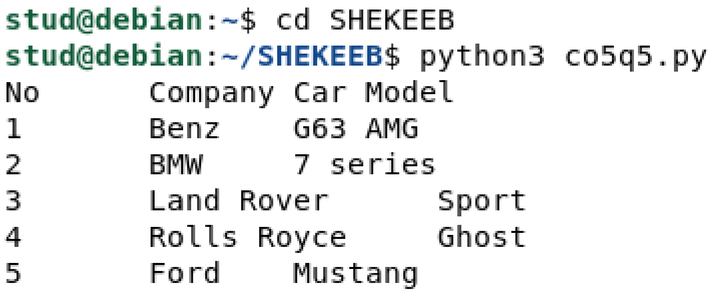
Program Code:

```
import csv
field_names = ['No', 'Company', 'Car Model']
cars = [
{'No': 1, 'Company': 'Benz', 'Car Model': 'G63 AMG'},
{'No': 2, 'Company': 'BMW', 'Car Model': '7 series'},
{'No': 3, 'Company': 'Land Rover', 'Car Model': 'Sport'},
{'No': 4, 'Company': 'Rolls Royce', 'Car Model': 'Ghost'},
{'No': 5, 'Company': 'Ford', 'Car Model': 'Mustang'},
]
csvfile=open('cars.csv', 'w')
writer = csv.DictWriter(csvfile, fieldnames = field_names)
writer.writeheader()
writer.writerows(cars)
csvfile.close()
csvfile=open('cars.csv','r')
csvreader=csv.reader(csvfile)
for i in csvreader:
        print(i[0],end='\t')
        print(i[1],end='\t')
        print(i[2])
csvfile.close()
```

Output:



No	Company	Car Model
1	Benz	G63 AMG
2	BMW	7 series
3	Land Rover	Sport
4	Rolls Royce	Ghost
5	Ford	Mustang



```
stud@debian:~$ cd SHEKEEB
stud@debian:~/SHEKEEB$ python3 co5q5.py
No      Company Car Model
1       Benz    G63 AMG
2       BMW     7 series
3       Land Rover      Sport
4       Rolls Royce     Ghost
5       Ford    Mustang
```