

# **FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)<sup>TM</sup>**

**HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577**



**FOCUS ON EXCELLENCE**

**20MCA131 PROGRAMMING LAB**

**LABORATORY RECORD**

**Name: AKHILADISON**

**Branch: MASTER OF COMPUTER APPLICATIONS**

**Semester: 1**

**Batch: A**

**Roll No: 09**

**University Registration Number: FIT21MCA-2009**

**MARCH 2022**

**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY  
(FISAT)<sup>TM</sup>**

**HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577**



**FOCUS ON EXCELLENCE**

**CERTIFICATE**

*This is to certify that this is a Bonafide record of the Practical work done by **AKHILA DISON (FIT21MCA-2009)** in the **20MCA131 PROGRAMMING LAB** Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.*

Signature of Staff in Charge

Name:

Signature of H O D

Name:

**Date of University practical examination .....**

Signature of  
Internal Examiner

Signature of  
External Examiner

**CONTENT**

SI No:	Date :	Name of Experiment:	Page No:	Signature of Staff –In – Charge:
<b>COURSE OUTCOME 1</b>				
1	28/10/2021	Display future leap years from current year to a final year entered by user.	7	
2	28/10/2021	List comprehensions: (a) Generate positive list of numbers from a given list of integers (b) Square of N numbers (c) Form a list of vowels selected from a given word (d) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)	7	
3	28/10/2021	Count the occurrences of each word in a line of text.	9	
4	28/10/2021	Prompt the user for a list of integers. For all values greater than 100, store 'over' instead	10	
5	10/11/2021	Store a list of first names. Count the occurrences of 'a' within the list	11	
6	10/11/2021	Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both.	12	
7	10/11/2021	Get a string from an input string where all occurrences of first character replaced with '\$', except first character	13	
8	10/11/2021	Create a string from given string where first and last characters exchanged. [eg: python -> nythop]	14	
9	10/11/2021	Accept the radius from user and find area of circle.	14	
10	11/11/2021	Find biggest of 3 numbers entered.	15	
11	11/11/2021	Accept a file name from user and print extension of that.	16	

12	11/11/2021	Create a list of colors from comma-separated color names entered by user. Display first and last colors.	16	
13	11/11/2021	Accept an integer n and compute $n+nn+nnn$ .	17	
14	11/11/2021	Print out all colors from color-list1 not contained in color-list2.	18	
15	17/11/2021	Create a single string separated with space from two strings by swapping the character at position 1.	18	
16	17/11/2021	Sort dictionary in ascending and descending order.	19	
17	17/11/2021	Merge two dictionaries.	20	
18	17/11/2021	Find gcd of 2 numbers.	20	
19	17/11/2021	From a list of integers, create a list removing even numbers.	21	
<b>COURSE OUTCOME 2</b>				
20	25/11/2021	Program to find the factorial of a number	22	
21	25/11/2021	Generate Fibonacci series of N terms	23	
22	25/11/2021	Find the sum of all items in a list	24	
23	25/11/2021	Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	23	
24	2/12/2021	Display the given pyramid with step number accepted from user.	25	
25	2/12/2021	Count the number of characters (character frequency) in a string.	26	
26	2/12/2021	Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'.	26	

27	9/12/2021	Accept a list of words and return length of longest word.	27	
28	9/12/2021	Construct following pattern using nested loop. *	28	
29	9/12/2021	Generate all factors of a number.	29	
<b>COURSE OUTCOME 3</b>				
30	29/01/2022	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)	30	
<b>COURSE OUTCOME 4</b>				
31	13/01/2022	Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	33	
32	13/01/2022	Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.	35	

33	13/01/2022	Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	36	
34	20/01/2022	Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time	37	
35	20/01/2022	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 overriding.	38	
<b>COURSE OUTCOME 5</b>				
36	03/02/2022	Write a Python program to read a file line by line and store it into a list.	40	
37	03/02/2022	Write a Python program to read each row from a given csv file and print a list of strings.	41	

**COURSE OUTCOME 1**

Sl no.:-1

**Aim:-** Display future leap years from current leap to a final year entered by the users.

**Program code:-**

```
currentYear=int(input("Enter the current year:"))
futureyear=int(input("Enter the future Year: "))
print("Leap Years in between the",currentYear,"and",futureyear)
for i in range(currentYear,futureyear):
    if i%4==0:
        print(i)
```

**Output:-**



```
===== RESTART: C:/Users/HP/OneDrive/Desktop/Akhila/python/CO1/1.py =====
Enter the current year:2000
Enter the future Year: 2021
Leap Years in between the 2000 and 2021
2000
2004
2008
2012
2016
2020
>>>
>>>
>>>
```

Sl no:-2

**Aim:-** List comprehension:-

- Generate positive list of numbers from a give list of integers.
- square of N numbers.
- Form a list of vowels selected from a give word.
- List ordinal values of each elements of a word.

a.

**Program code:-**

```
print("The given list is ")
list=[19,2,-3,0,6,-9,4,10]
print(list)
print("The positive integers are \n ")
for i in list:
```

```

if i>0:
    print(i)

```

**Output:-**

```

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/HP/OneDrive/Desktop/Akhila/python/CO1/3a.py =====
The given list is
[19, 2, -3, 0, 6, -9, 4, 10]
The positive integers are
19
2
6
4
10
>>>
Ln: 14 Col: 1

```

b.

**Program code:-**

```

print("The given list is ")
list=[19,2,-3,0,6,-9,4,10]
print(list)
print("The square's of given list is ")
for i in list:
    sqr=i*i
    print(sqr)

```

**Output:-**

```

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/HP/OneDrive/Desktop/Akhila/python/CO1/3b.py =====
The given list is
[19, 2, -3, 0, 6, -9, 4, 10]
The square's of given list is
361
4
9
0
36
81
16
100
>>>
Ln: 16 Col: 1

```

c.

**Program code:-**

```

word=input("Enter a word:")
list=['A','E','I','O','U','a','e','i','o','u']
temp=[]
for i in word:

```



```

if (i in list and i not in temp):
    temp.append(i)
print("The vowels in the word that you entered are ",temp)

```

**Output:-**

```

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\HP\OneDrive\Desktop\Akhila\python\CO1\3c.py =====
Enter a word:Akhila Dison
The vowels in the word that you entered are  ['A', 'i', 'a', 'o']
>>>
Ln: 7 Col: 0

```

d.

**Program code:-**

```

word=input("Enter a word:")
temp=[]
print("The ordinal values of each element of the word is ")
for i in word:
    x=ord(i)
    temp.append(x)
print(temp)

```

**Output:-**

```

IDLE Shell 3.10.0
File Edit Shell Debug Options Window Help
Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\HP\OneDrive\Desktop\Akhila\python\CO1\3d.py =====
Enter a word:Akhila Dison
The ordinal values of each element of the word is
[65, 107, 104, 105, 108, 97, 32, 68, 105, 115, 111, 110]
>>>
Ln: 8 Col: 0

```

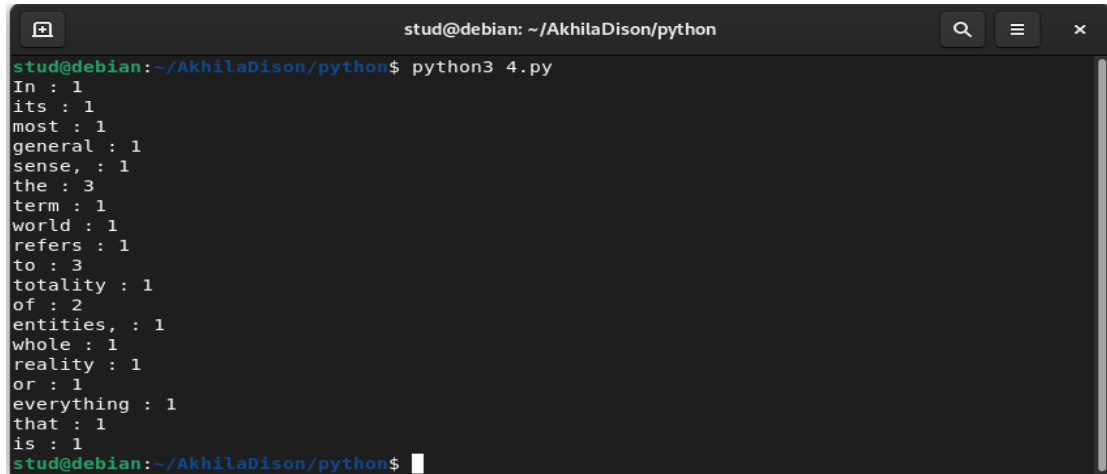
**Sl no.:-3****Aim:-**Count the occurrence of each word in a line of text.**Program code:-**

```

k="In its most general sense, the term world refers to the totality
of entities, to the whole of reality or to everything that is"
h=[]
for i in k.split(" "):
    if i not in h:
        h.append(i)

```

```
for i in h:
    print(i,":",k.split(" ").count(i))
```

**Output:-**


```
stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 4.py
In : 1
its : 1
most : 1
general : 1
sense, : 1
the : 3
term : 1
world : 1
refers : 1
to : 3
totality : 1
of : 2
entities, : 1
whole : 1
reality : 1
or : 1
everything : 1
that : 1
is : 1
stud@debian:~/AkhilaDison/python$
```

**Sl no.:-4**

**Aim:-**Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.

**Program code:-**

```
integers=int(input("Enter the limit:"))
temp=[]
over=0
for i in range(0,integers):
    user=int(input("Enter a number:"))
    if(user>=100):
        temp.append(user)
    else:
        temp.append(over)
print("List elements are",temp)
```

**Output:-**

```
stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 5.py
Enter the limit:5
Enter a number:-12
Enter a number:12
Enter a number:100
Enter a number:123
Enter a number:0
List elements are [0, 0, 100, 123, 0]
stud@debian:~/AkhilaDison/python$
```

**Sl no.:-5****Aim:-**store a list of first names.count the occurrence of “a” within the list.**Program code:-**

```
list= ['akhila','ashik','mary']
count=0
for i in list:
    value= i.count('a')
    count=count+value
print(list)
print(count)
```

**Output:-**

```
stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 6.py
['akhila', 'ashik', 'mary']
4
stud@debian:~/AkhilaDison/python$
```

**Sl no.:- 6**

**Aim:-** Enter 2 list of first integers. Check

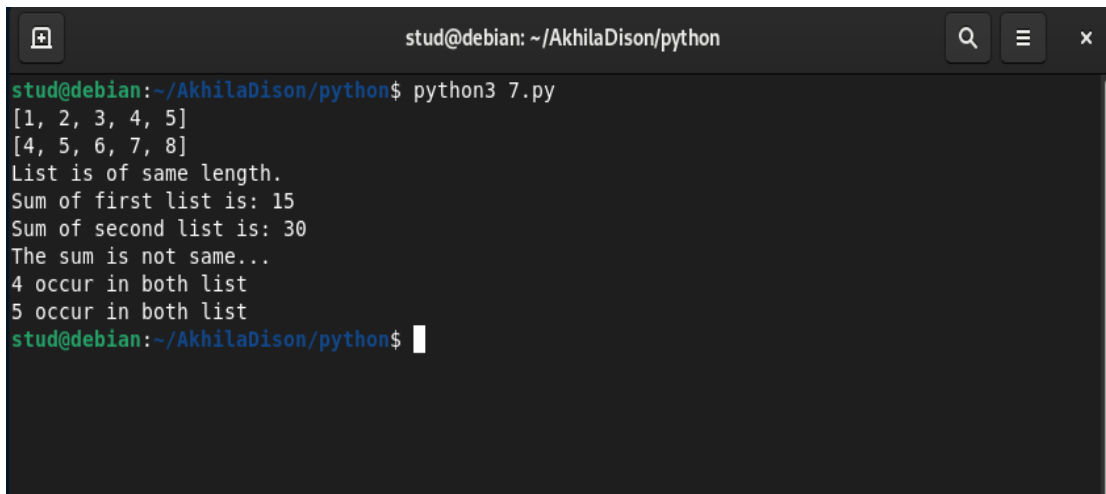
- a. whether list are of same length.
- b. Whether list sums to same value.
- b. whether any value occur in both.

**Program code:-**

```
#whether list are of same length.
list1=[1,2,3,4,5]
list2=[4,5,6,7,8]
print(list1)
print(list2)
x=len(list1)
y=len(list2)
if(x==y):
    print("List is of same length.")
else:
    print("List is not same length.")

#Whether list sums to same value.
sum1=0
sum2=0
for i in list1:
    sum1=sum1+i
print("Sum of first list is:",sum1)
for j in list2:
    sum2=sum2+j
print("Sum of second list is:",sum2)
if(sum1==sum2):
    print("The sum is same....")
else:
    print("The sum is not same...")

#whether any value occur in both.
for i in list1:
    if i in list2:
        print(i,"occur in both list")
```

**Output:-**


```

stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 7.py
[1, 2, 3, 4, 5]
[4, 5, 6, 7, 8]
List is of same length.
Sum of first list is: 15
Sum of second list is: 30
The sum is not same...
4 occur in both list
5 occur in both list
stud@debian:~/AkhilaDison/python$

```

Sl no:-7

**Aim:-**Get a string from an input string where all occurrence of first character replace with '\$', except first character.

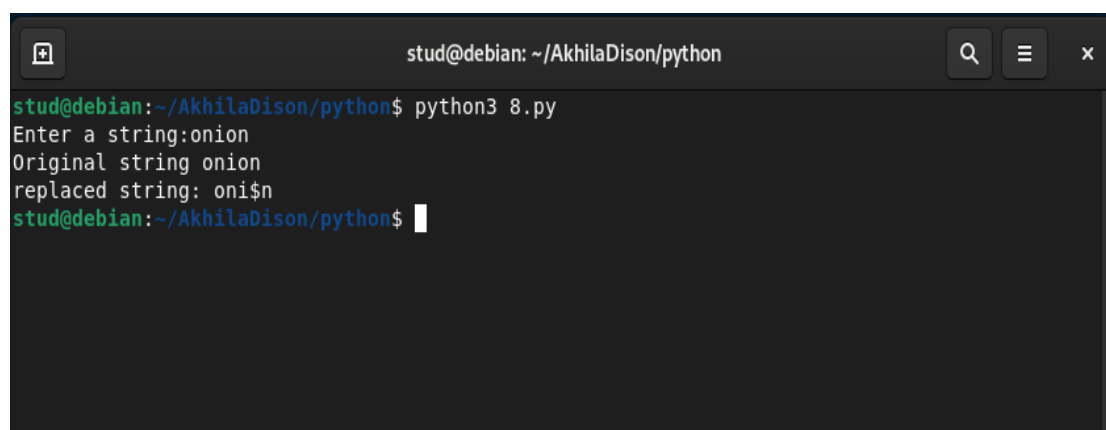
[eg:-onion->oni\$n]

**Programme code:-**

```

s=input("Enter a string:")
print("Original string",s)
char=s[0]
s=s.replace(char,'$')
s=char+s[1:]
print("replaced string:",s)

```

**Output:-**


```

stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 8.py
Enter a string:onion
Original string onion
replaced string: oni$n
stud@debian:~/AkhilaDison/python$

```

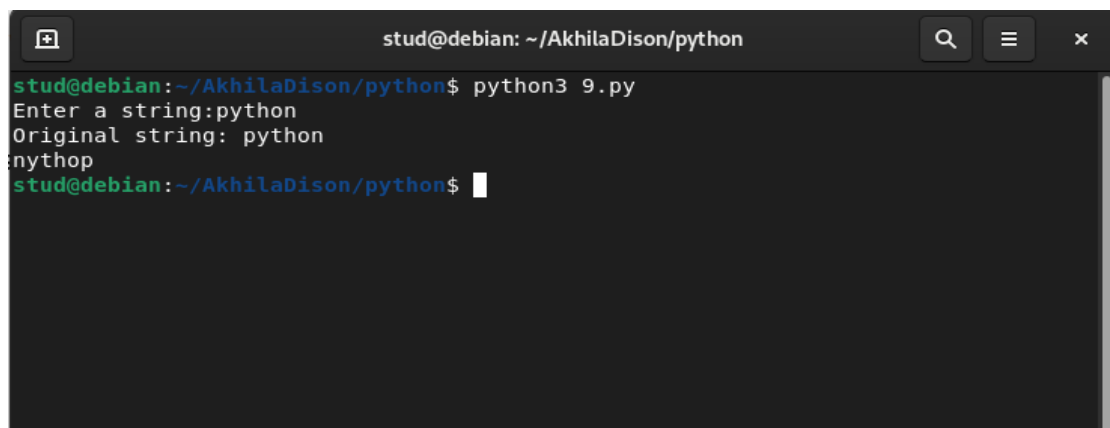
**Sl no.:-8**

**Aim:-**Create a string from give string where first and last charater exchanged.[eg:-python->nythop]

**Program Code:-**

```
s=input("Enter a string:")
print("Original string:",s)
sf=s[0]
sl=s[-1]
n=len(s)
ns=sl+s[1:n-1]+sf
print(ns)
```

**Output:-**



```
stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 9.py
Enter a string:python
Original string: python
nythop
stud@debian:~/AkhilaDison/python$
```

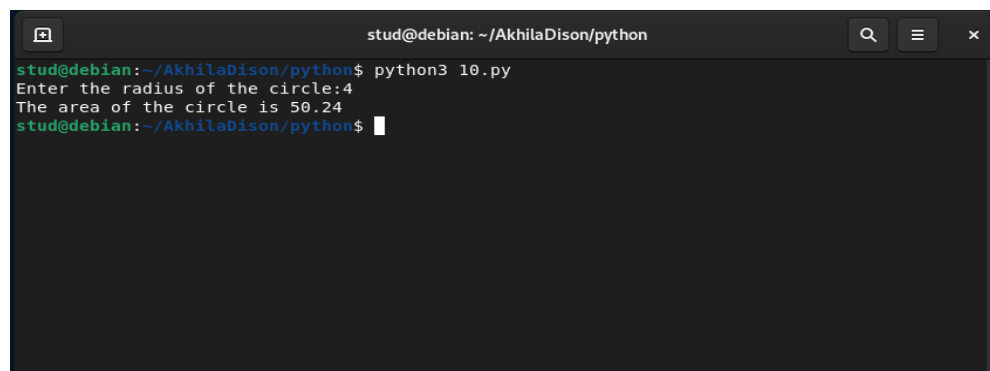
**Sl no.:-9**

**Aim:-**Accept the radius from the user and find the area of circle.

**Program Code:-**

```
y=int(input("Enter the radius of the circle:"))
a=3.14*y*y
print("The area of the circle is",a)
```

**Output:-**



```
stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 10.py
Enter the radius of the circle:4
The area of the circle is 50.24
stud@debian:~/AkhilaDison/python$
```

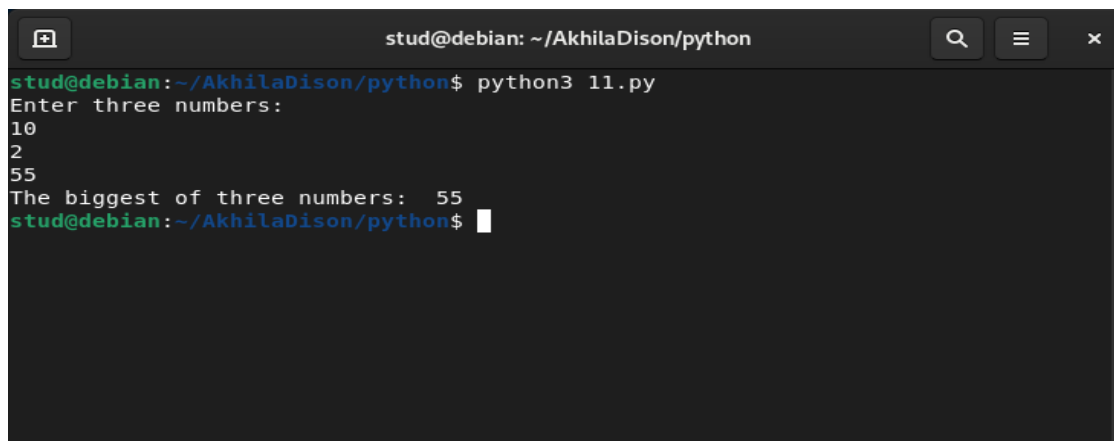
**Sl no.:-10**

**Aim:-**find the greatest of 3 entered.

**Program code:-**

```
print("Enter three numbers: ")
a=int(input())
b=int(input())
c=int(input())
if a>b and a>c:
    print("The biggest of three numbers: ",a)
if b>a and b>c:
    print("The biggest of three numbers: ",b)
if c>a and c>b:
    print("The biggest of three numbers: ",c)
```

**Output:-**



The screenshot shows a terminal window with the title 'stud@debian: ~/AkhilaDison/python'. The user enters the command 'python3 11.py'. The program prompts 'Enter three numbers:' and the user enters '10', '2', and '55' on separate lines. The program then outputs 'The biggest of three numbers: 55'.

```
stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 11.py
Enter three numbers:
10
2
55
The biggest of three numbers: 55
stud@debian:~/AkhilaDison/python$
```

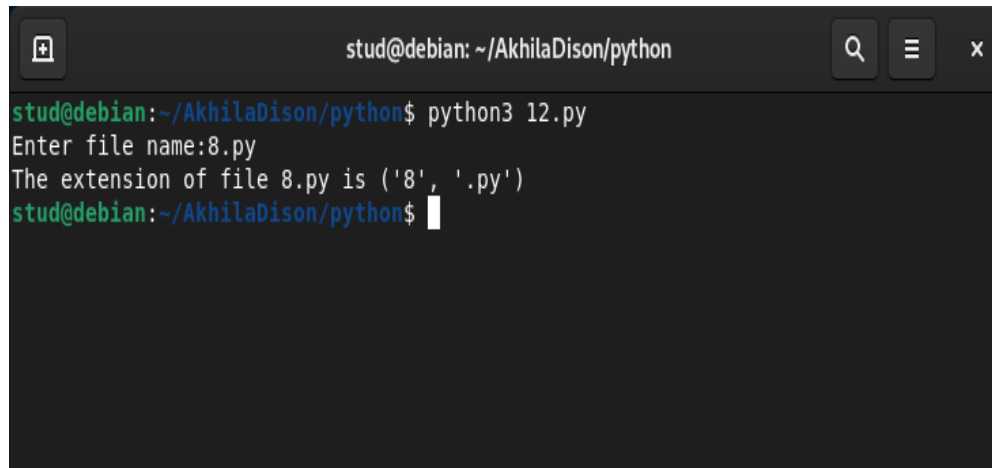
**Sl no.:-11**

**Aim:-**Accept file name from user and print extension of that.

**Program code:-**

```
import os
a=input("Enter file name:")
print("The extension of file",a,"is",os.path.splitext(a))
```

**Output:-**



```
stud@debian: ~/AkhilaDison/python
stud@debian:~/AkhilaDison/python$ python3 12.py
Enter file name:8.py
The extension of file 8.py is ('8', '.py')
stud@debian:~/AkhilaDison/python$
```

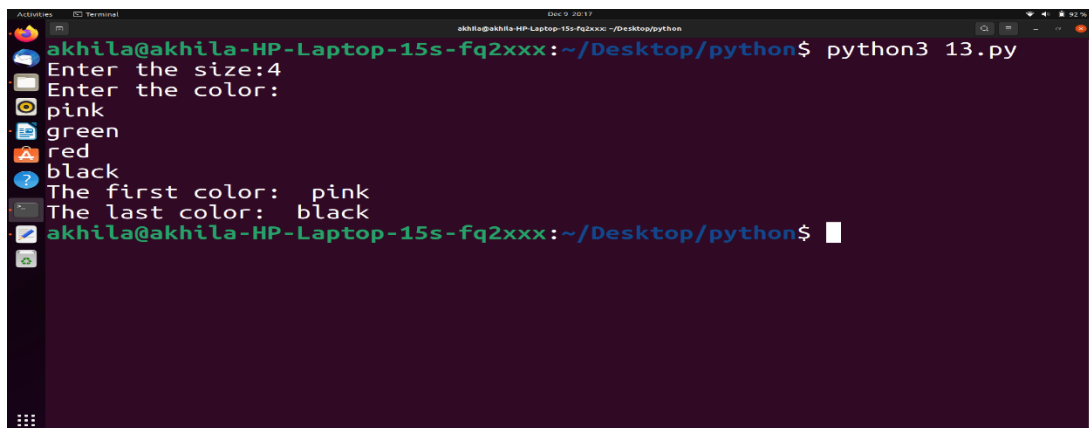
**Sl no.:-12**

**Aim:-**Create a list of colours from comma separated color names entered by users. Display first and last color.

**Program code:-**

```
n=int(input("Enter the size:"))
lt = []
l=[]
print("Enter the color:")
for i in range(0,n):
    lt = input()
    l.append(lt)
print("The first color: ",l[0])
print("The last color: ",l[n-1])
```



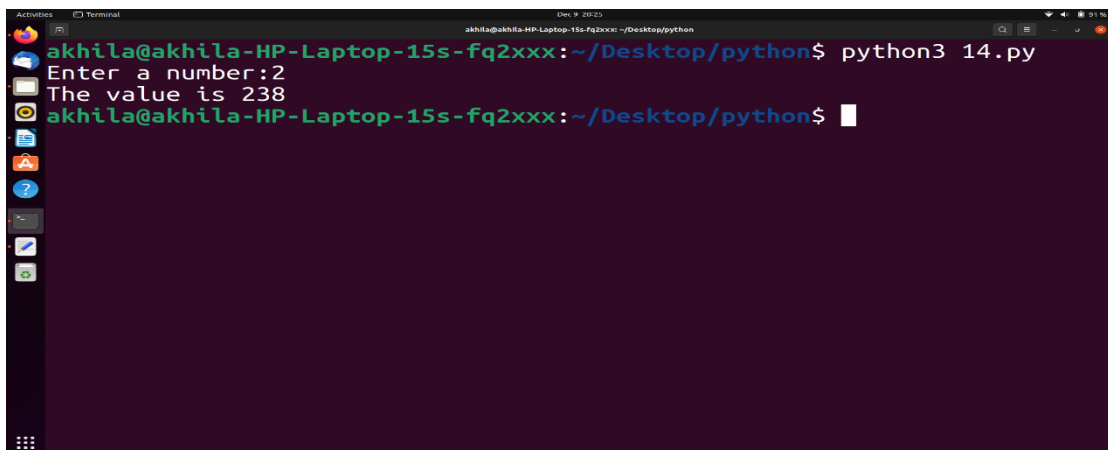
**Output:-**

```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$ python3 13.py
Enter the size:4
Enter the color:
pink
green
red
black
The first color: pink
The last color: black
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$
```

SI no:-13

**Aim:-**Accept an integer and compute  $n+nn+nnn$ .**Program code:-**

```
i=int(input("Enter a number:"))
value=i+((i*10)+i)+((i*100)+(i+10)+i)
print("The value is",value)
```

**Output:-**

```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$ python3 14.py
Enter a number:2
The value is 238
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$
```

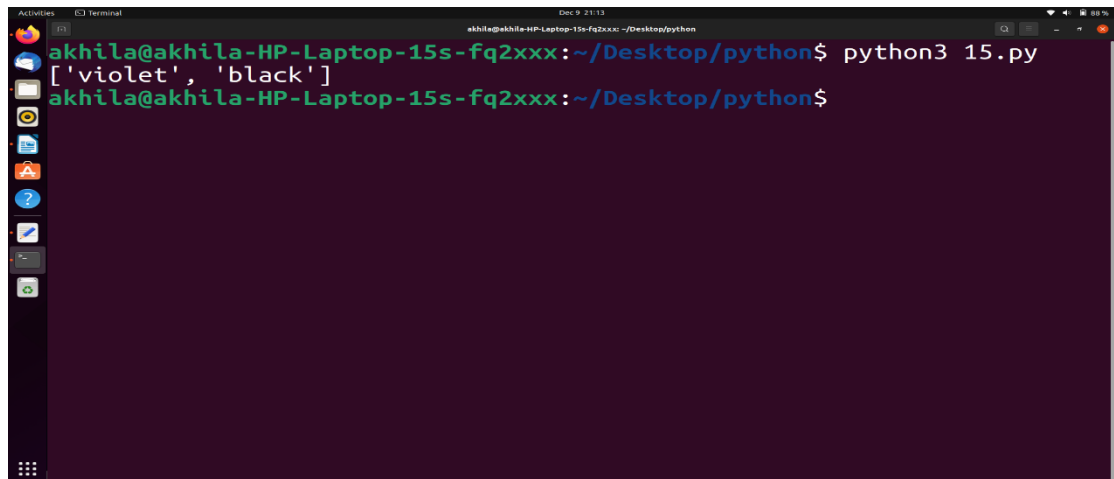
**Sl no.:-14**

**Aim:-**Print out all colors from color-list 1 not contained in color-list2

**Program code:-**

```
list1= ['red', 'violet', 'black']
list2= ['blue', 'pink', 'red']
list3=[]
for i in list1:
    if i not in list2:
        list3.append(i)
print(list3)
```

**Output:-**



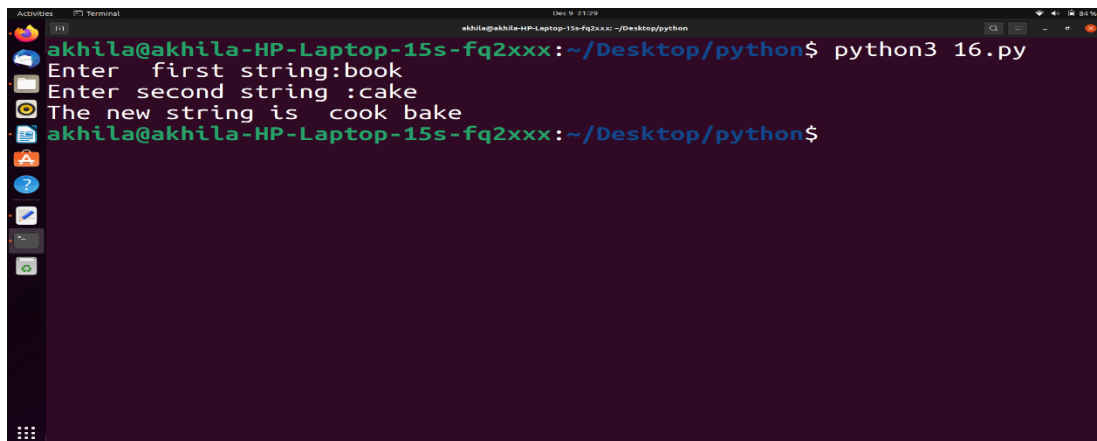
```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$ python3 15.py
['violet', 'black']
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$
```

**Sl no.:-15**

**Aim:-**Create a single string separated with space from two strings by swapping the character at position 1.

**Program code:-**

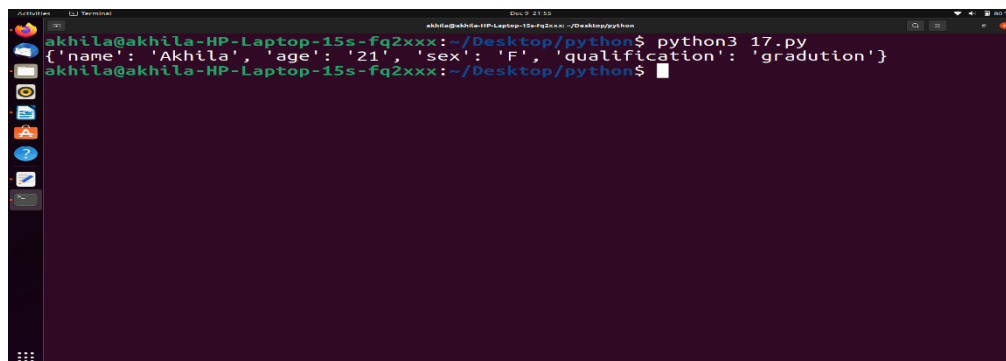
```
a=input("Enter string 1:")
b=input("Enter string 2:")
a1=b[0]+a[1:]
b1=a[0]+b[1:]
c=a1+' '+b1
print(c)
```

**Output:-**

```
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python$ python3 16.py
Enter first string:book
Enter second string :cake
The new string is  cook bake
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python$
```

**Sl no.:-16****Aim:-**Sort dictionary in ascending and descending order.**Program code:-**

```
d1={"name":"Akhila","age":"21"}
d2={"sex":"F","qualification":"gradution"}
d1.update(d2)
print(d1)
```

**Output:-**

```
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python$ python3 17.py
{'name': 'Akhila', 'age': '21', 'sex': 'F', 'qualification': 'gradution'}
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python$
```

**Sl no.:-17**

**Aim:-**merge two dictionary

**Program code:-**

```
thisdict ={\n    "Name": "Akhila",\n    "age": 21,\n    "dob": "02/005/2000"\n}\nd={"College Name":'FISAT' , "rollno":'09'}\nd.update(thisdict)\nprint(d)
```

**Output:-**

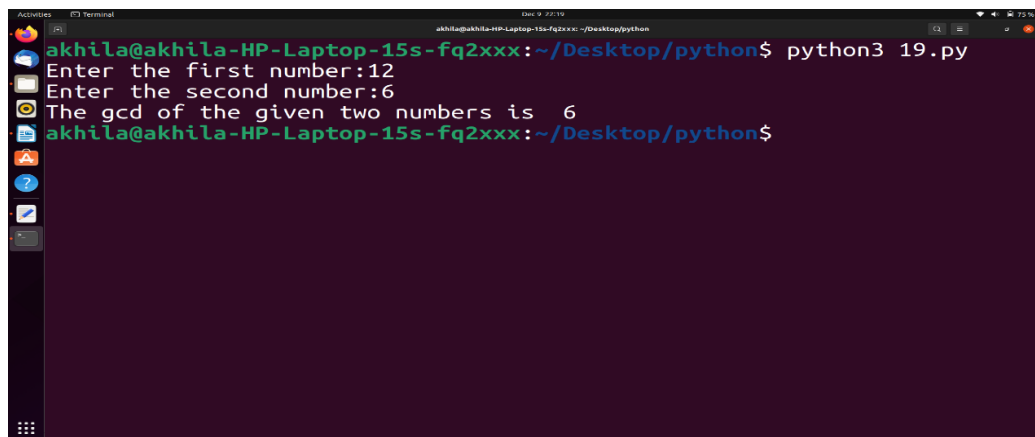


**Sl no.:-18**

**Aim:-**find GCD of 2 numbers

**Program Code:-**

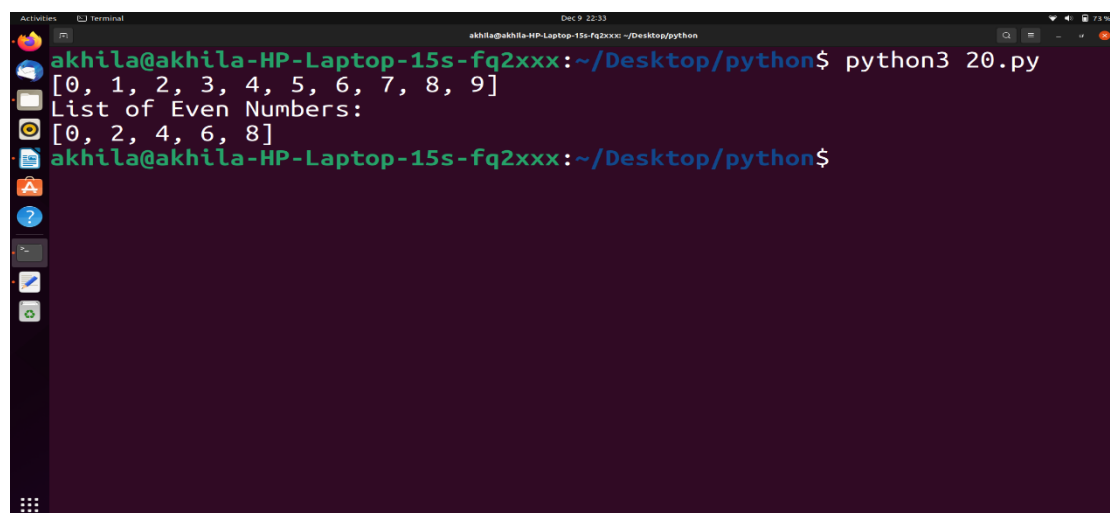
```
x=int(input("Enter the first number:"))\ny=int(input("Enter the second number:"))\nz=max(x,y)\nfor i in range(1,z+1):\n    if(x%i==0) and (y%i==0):\n        gcd=i\nprint("The gcd of the given two numbers is ",gcd)
```

**Output:-**

```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$ python3 19.py
Enter the first number:12
Enter the second number:6
The gcd of the given two numbers is 6
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$
```

**Sl no:-19****Aim:-**From a list of integers ,create a list removing even numbers.**Program code:-**

```
l=[0,1,2,3,4,5,6,7,8,9]
print(l)
li=[]
print("List of Even Numbers:")
for i in l:
    if(i%2==0):
        li.append(i)
print(li)
```

**Output:-**

```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$ python3 20.py
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
List of Even Numbers:
[0, 2, 4, 6, 8]
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python$
```

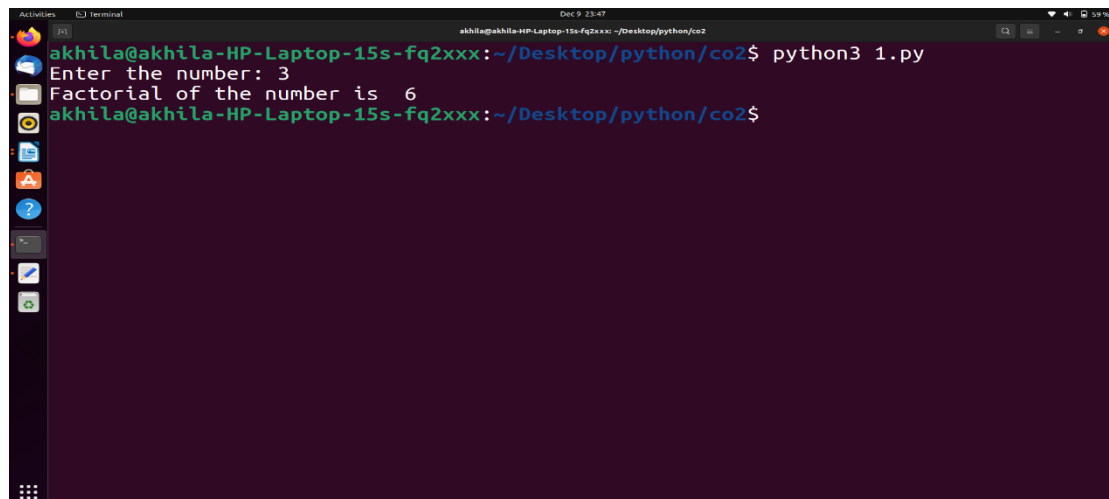
Sl no.:-20

Aim:-Program to find the factorial of a number.

Program code:-

```
num=int(input("Enter the number: "))  
fact=1  
for i in range(1,num+1):  
    fact=fact*i  
print("Factorial of the number is ",fact)
```

Output:-

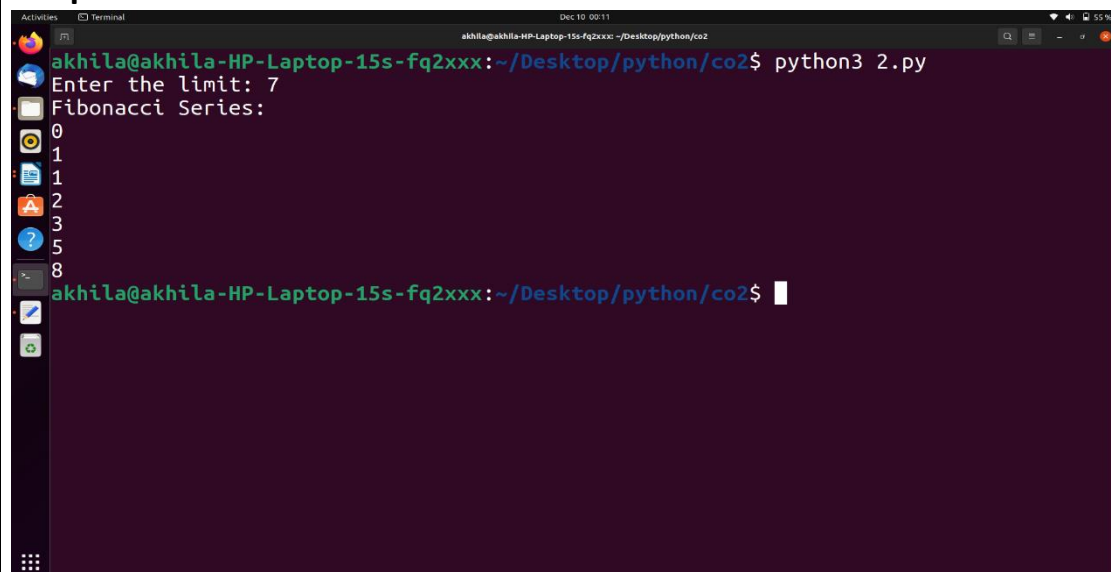


The screenshot shows a terminal window with a dark background. The prompt is 'akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2'. The user has entered 'python3 1.py'. The program prompts 'Enter the number: 3'. The user has entered '3'. The program outputs 'Factorial of the number is 6'. The prompt is now 'akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2\$'.

```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 1.py  
Enter the number: 3  
Factorial of the number is 6  
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$
```

**COURSE OUTCOME 2****Sl no.:-21****Aim:-**Generate Fibonacci series of N terms.**Program Code:-**

```
n=int(input("Enter the limit: "))
a=0
b=1
c=0
print("Fibonacci Series:")
print(a)
print(b)
for i in range(3,n+1):
    c=a+b
    print(c)
    a=b
    b=c
```

**Output:-**A screenshot of a Linux terminal window. The title bar shows 'Terminal' and 'Dec 10 00:11'. The terminal content shows the user 'akhila' at 'akhila@akhila-HP-Laptop-15s-fq2xxx' in the directory '~/Desktop/python/co2'. The user runs 'python3 2.py'. The program prompts 'Enter the limit: 7' and then prints 'Fibonacci Series:'. The output shows the numbers 0, 1, 1, 2, 3, 5, 8 on separate lines. The prompt 'akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python/co2\$' is visible at the bottom.

```
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python/co2$ python3 2.py
Enter the limit: 7
Fibonacci Series:
0
1
1
2
3
5
8
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python/co2$
```

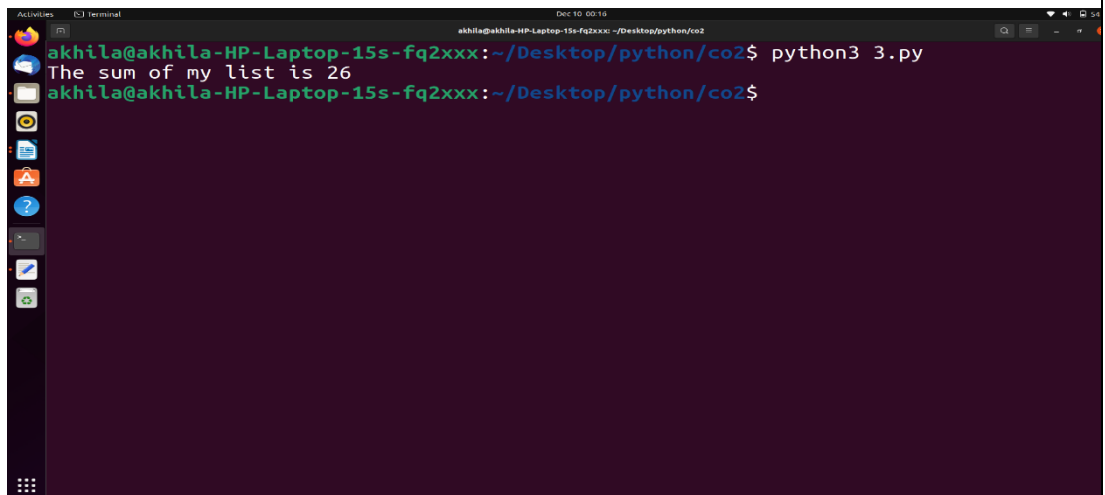
**Sl no.:-22**

**Aim:-**Find the sum of all items in a list.

**Program code:-**

```
def sum_of_list(l):
    total = 0
    for val in l:
        total = total+val
    return total
my_list = [3,5,7,9,2]
print("The sum of my list is", sum_of_list(my_list))
```

**Output:-**



```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 3.py
The sum of my list is 26
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$
```

**Sl no.:-23**

**Aim:-**Generate a list of 4 digit number in a give range with all their digits even and the number is a perfect square.

**Program code:-**

```
limit1=4286
limit2=8642
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(k)
print(list1)

```

**Output:-**

```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 4.py
68
78
80
92
[4624, 6084, 6400, 8464]
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$

```

**Sl no:-24**

**Aim:-**Display the given pyramid with step number accepted from user.

Eg=4

```

1
2 4
3 6 9
4 8 12 16

```

**Program code:-**

```

for i in range(1,5):
    for j in range(1,i+1):
        print(i*j,end=" ")
    print("\n")

```

**Output:-**

```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 5.py
1
2 4
3 6 9
4 8 12 16
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$

```

**Sl no.:-25**

**Aim:-**Count the number of character in a string.

**Program Code:-**

```

string = "hello world";
count = 0;
for i in range(0, len(string)):
    if(string[i] != ' '):
        count = count + 1;
print("Total number of characters in a string: " + str(count));

```

**Output:-**

```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 6.py
Total number of characters in a string: 10
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$

```

**Sl no.:-26**

**Aim:-**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:")
if s[-3:]=='ing':
    s=s+'ly'
else:
    s=s+'ing'
print("New string:",s)

```

**Output:-**

```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 7.py
Enter a string: beggingly
New string: beggingly
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 7.py
Enter a string: kinly
New string: kinly
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$

```

**Sl no:-27****Aim:-**Accept a list of words and return length of longest word.**Program code:-**

```

l1=[]
num=int(input("Enter the range:"))
print("Enter the words:")
for i in range(0,num):
    l1.append(input(""))
longest=l1[0]
for i in range(1,num):
    if(len(l1[i])>len(longest)):
        longest=l1[i]
print("Length of longest word is",len(longest))

```

**Output:-**

```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 8.py
Enter the range:5
Enter the words:
hello
welcome
to
new
day
Length of longest word is 7
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$

```

**Sl no.:-28**

**Aim:-**Construct the pattern using nested loop.

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

**Program code:-**

```
for i in range(0,5):
    for j in range(0,i):
        print("*",end=" ")
    print("\n")
for i in range(5,0,-1):
    for j in range(0,i):
        print("*",end=" ")
    print("\n")
```

**Output:-**

```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/python/co2$ python3 9.py
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

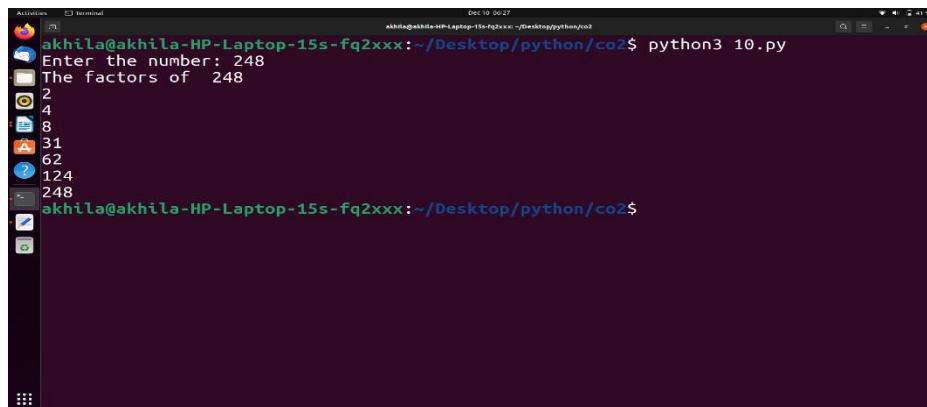
Sl no:-29

Aim:-Generate all factors of a number.

Program code:-

```
n=int(input("Enter the number: "))
i=2
print("The factors of ",n)
while i<=n:
    if (n % i==0):
        print(i)
    i = i + 1
```

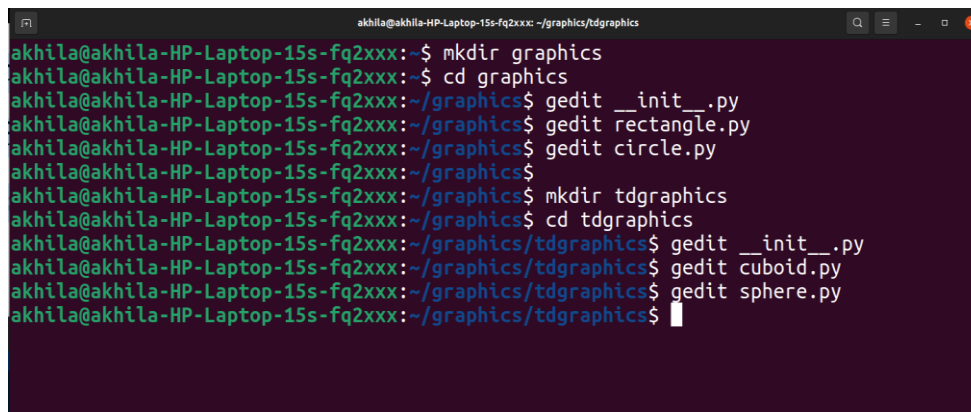
Output:-

A screenshot of a terminal window on a Linux system. The terminal shows the execution of a Python script named 10.py. The user enters the number 248, and the program outputs the factors of 248: 2, 4, 8, 31, 62, 124, and 248. The terminal window has a dark background and a light-colored text. The prompt is akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python/co2\$. The output is as follows:

```
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python/co2$ python3 10.py
Enter the number: 248
The factors of 248
2
4
8
31
62
124
248
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/python/co2$
```

**COURSE OUTCOME 3****Sl no.:-30**

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

**Terminal Commands:-**


```

akhila@akhila-HP-Laptop-15s-fq2xxx:~$ mkdir graphics
akhila@akhila-HP-Laptop-15s-fq2xxx:~$ cd graphics
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics$ gedit __init__.py
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics$ gedit rectangle.py
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics$ gedit circle.py
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics$
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics$ mkdir tdgraphics
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics$ cd tdgraphics
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics/tdgraphics$ gedit __init__.py
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics/tdgraphics$ gedit cuboid.py
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics/tdgraphics$ gedit sphere.py
akhila@akhila-HP-Laptop-15s-fq2xxx:~/graphics/tdgraphics$

```

**Program Code:-****graphics\rectangle.py**

```

class Rectangle:
    def __init__(self,length,width):
        self.length=length
        self.width=width
    def area(self):
        return (self.length*self.width)
    def perimeter(self):
        return (2*(self.length+self.width))

```

**graphics\circle.py**

```

global pi
pi=3.1416
class Circle:
    global pi
    pi=3.1416
    def __init__(self,radius):
        self.radius=radius
    def area(self):
        return (pi*(self.radius**2))

```

```
def perimeter(self):
    return (2*pi*self.radius)
```

### tdgraphics\cuboid.py

```
class Cuboid:
def __init__(self,length,width,height):
    self.l=length
    self.w=width
    self.h=height
def volume(self):
    return (self.l*self.w*self.h)
def area(self):
    l=self.l
    w=self.w
    h=self.h
    return (2*((l*w)+(w*h)+(l*h)))
```

### tdgrapics\sphere.py

```
global pi
pi=3.1416
class Sphere:
    def __init__(self,radius):
        self.radius=radius
    def volume(self):
        r=self.radius
        return ((4/3)*pi*(r**3))
    def area(self):
        r=self.radius
        return (4*pi*(r**2))
```

### 1.py

```
from graphics import rectangle as rt
from graphics import circle
from graphics.tdgraphics import sphere
from graphics.tdgraphics import cuboid
```

```
#Rectangle
r=rt.Rectangle(10,12)
```

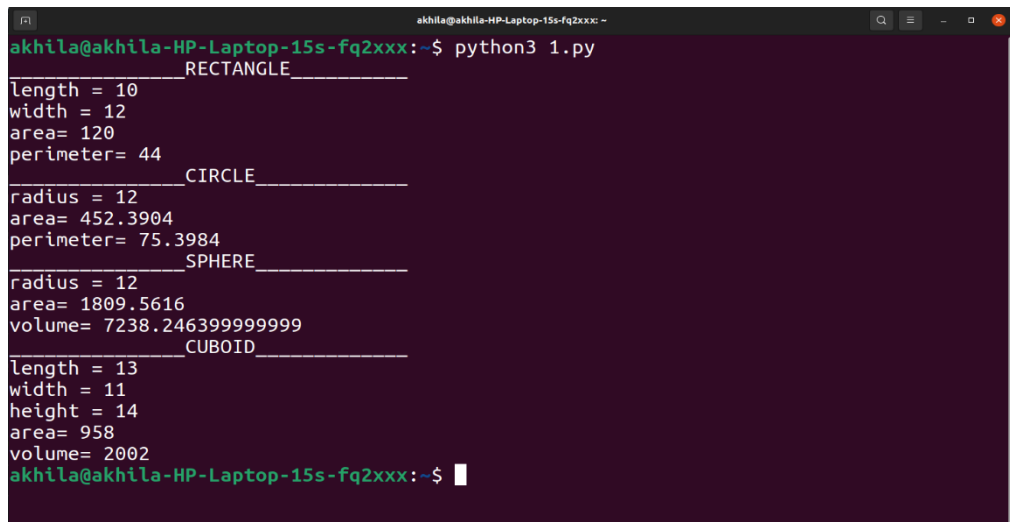
```

print("_____RECTANGLE_____")
print("length =",r.length)
print("width =",r.width)
print("area=",r.area())
print("perimeter=",r.perimeter())

#Circle
c=Circle.Circle(12)
print("_____CIRCLE_____")
print("radius =",c.radius)
print("area=",c.area())
print("perimeter=",c.perimeter())

#Sphere
s=sphere.Sphere(12)
print("_____SPHERE_____")
print("radius =",s.radius)
print("area=",s.area())
print("volume=",s.volume())
#Cuboid
cu=cuboid.Cuboid(13,11,14)
print("_____CUBOID_____")
print("length =",cu.l)
print("width =",cu.w)
print("height =",cu.h)
print("area=",cu.area())
print("volume=",cu.volume())

```

**Output:-**


```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~$ python3 1.py
_____RECTANGLE_____
length = 10
width = 12
area= 120
perimeter= 44
_____CIRCLE_____
radius = 12
area= 452.3904
perimeter= 75.3984
_____SPHERE_____
radius = 12
area= 1809.5616
volume= 7238.246399999999
_____CUBOID_____
length = 13
width = 11
height = 14
area= 958
volume= 2002
akhila@akhila-HP-Laptop-15s-fq2xxx: ~$

```



**COURSE OUTCOME 4****Sl no.:-31****Aim:-**Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two rectangle objects by their area.**Program code:-**

```
class Rectangle:
    def __init__(self,length,breadth):
        self.len=length
        self.brea=breadth

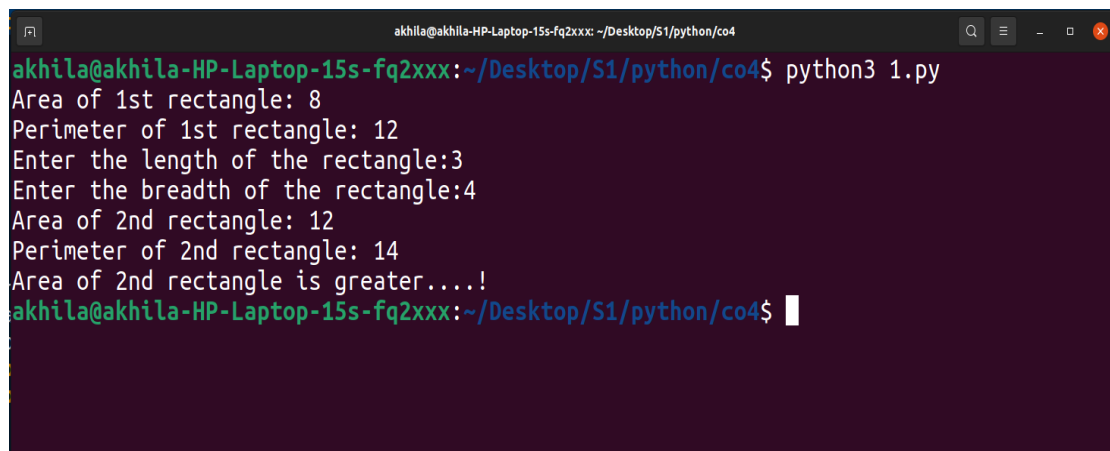
    def area(self):
        return self.len*self.brea

    def perimeter(self):
        return 2*(self.len+self.brea)

r1=Rectangle(2,4)
a1=r1.area()
p1=r1.perimeter()
print("Area of 1st rectangle:",a1)
print("Perimeter of 1st rectangle:",p1)

l=int(input("Enter the length of the rectangle:"))
b=int(input("Enter the breadth of the rectangle:"))
r2=Rectangle(l,b)
a2=r2.area()
p2=r2.perimeter()
print("Area of 2nd rectangle:",a2)
print("Perimeter of 2nd rectangle:",p2)
```

```
if (a1>a2):  
    print("Area of 1st rectangle is greater.....!")  
elif(a1==a2):  
    print("Area of both rectangle is same.....!")  
else:  
    print("Area of 2nd rectangle is greater....!")
```

**Output:-**

```
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/S1/python/co4  
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/S1/python/co4$ python3 1.py  
Area of 1st rectangle: 8  
Perimeter of 1st rectangle: 12  
Enter the length of the rectangle:3  
Enter the breadth of the rectangle:4  
Area of 2nd rectangle: 12  
Perimeter of 2nd rectangle: 14  
Area of 2nd rectangle is greater....!  
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/S1/python/co4$
```

Sl no.:-32

**Aim:-**Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

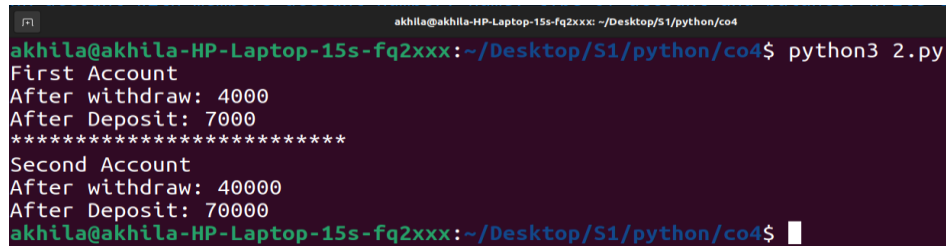
**Program code:-**

```
class Bank:
    def __init__(self,acc_no,name,type_of_acc,balance):
        self.acc_no=acc_no
        self.name=name
        self.type_of_acc=type_of_acc
        self.balance=balance

    def deposit(self,x):
        return self.balance+x

    def withdraw(self,y):
        return self.balance-y

print('First Account')
acc1=Bank(111,"Emma","personal",5000)
w1=acc1.withdraw(1000)
d1=acc1.deposit(2000)
print('After withdraw:',w1)
print('After Deposit:',d1)
print('*****')
print('Second Account')
acc2=Bank(222,"Emliy","personal",50000)
w2=acc2.withdraw(10000)
d2=acc2.deposit(20000)
print('After withdraw:',w2)
print('After Deposit:',d2)
```

**Output:-**


```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/S1/python/co4$ python3 2.py
First Account
After withdraw: 4000
After Deposit: 7000
*****
Second Account
After withdraw: 40000
After Deposit: 70000
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/S1/python/co4$

```

**Sl no.:-33**

**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 __lt__(self,rec2):
        if self.__length * self.__width < rec2.__length *
rec2.__width :
            return True
        else:
            return False
rec1=Rectangle(2,3)
rec2=Rectangle(1,3)

if rec1 < rec2:
    print("Second rectangle is larger...!")
else:
    print("First rectangle is larger...!")

```

**Output:-**

```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/S1/python/co4$ python3 3.py
First rectangle is larger...!
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/S1/python/co4$

```

**Sl no.:-34**

**Aim:-**Create a class Time with private attributes hour,minute and second.  
Overload '+' operator to find sum of two time.

**Program code:-**

```

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

    def __add__(self,t2):
        x=t1.__hour+t2.__hour
        y=t1.__minute+t2.__minute
        z=t1.__second+t2.__second
        print('Time is.....')
        print(x,":",y,":",z)

t1=Time(4,10,5)
t2=Time(4,20,15)
t3=t1+t2

```

**Output:-**

```

akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/S1/python/co4$ python3 4.py
Time is.....
8 : 30 : 20
akhila@akhila-HP-Laptop-15s-fq2xxx: ~/Desktop/S1/python/co4$

```

Sl no.:-35

**Aim:-**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.

**Program code:-**

```
class Publisher:
    def __init__(self,name):
        self.name=name

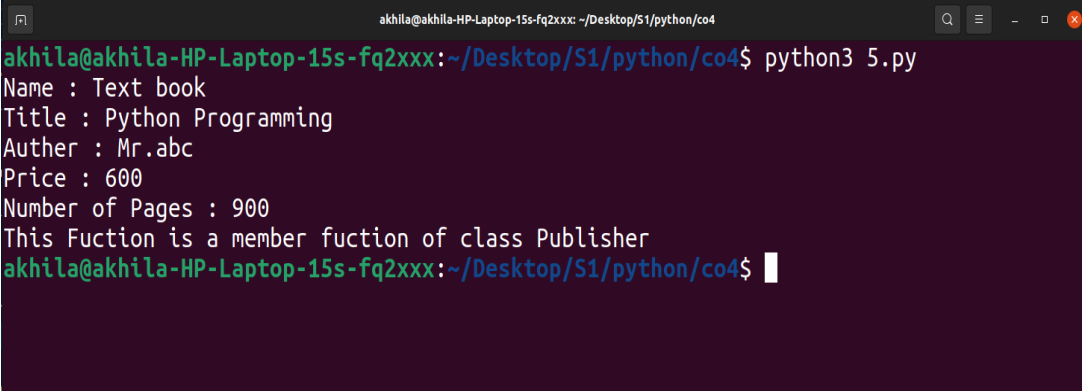
class Book(Publisher):
    def __init__(self,name,title,author):
        super().__init__(name)
        self.title=title
        self.author=author

class Python(Book):
    def __init__(self,name,title,author,price,pages):
        super().__init__(name,title,author)
        self.price=price
        self.pages=pages

    def print_function(self):
        print("Name :",self.name)
        print("Title :",self.title)
        print("Auther :",self.author)
        print("Price :",self.price)
        print("Number of Pages :",self.pages)

p1=Python("Text book","Python Programming","Mr.abc",600,900)
p1.print_function()
p2=Book("a","b","c")
p2.print_function()
```

**Output:-**

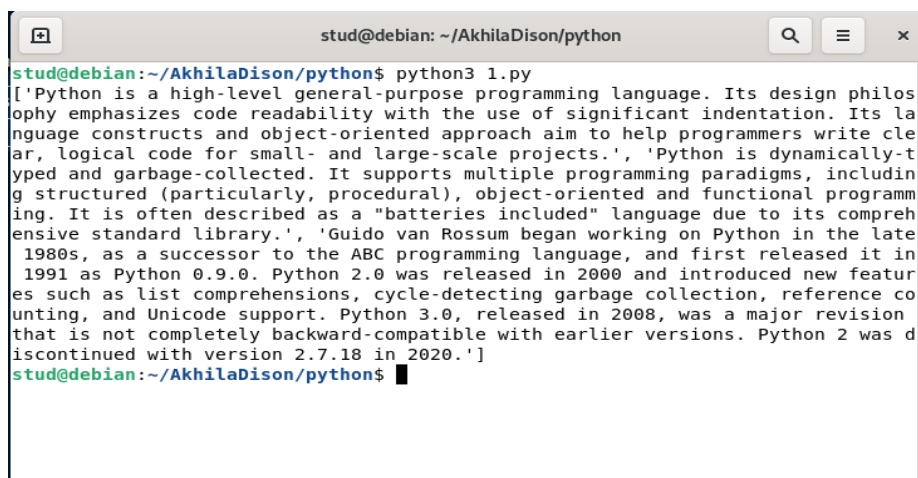
A terminal window with a dark purple background and white text. The window title bar shows the user 'akhila' on a laptop 'akhila-HP-Laptop-15s-fq2xxx' in the directory '~/Desktop/S1/python/co4'. The command 'python3 5.py' has been executed, resulting in the following output: 'Name : Text book', 'Title : Python Programming', 'Auther : Mr.abc', 'Price : 600', 'Number of Pages : 900', and 'This Fuction is a member fuction of class Publisher'. The prompt 'akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/S1/python/co4\$' is visible at the bottom.

```
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/S1/python/co4$ python3 5.py
Name : Text book
Title : Python Programming
Auther : Mr.abc
Price : 600
Number of Pages : 900
This Fuction is a member fuction of class Publisher
akhila@akhila-HP-Laptop-15s-fq2xxx:~/Desktop/S1/python/co4$
```

**COURSE OUTCOME 5****Sl no.:-36****Aim:-**Write a program to read a file line by line and store it into a list.**Program code:-**

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

Python is a high-level general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.

**Output:-**

```
stud@debian: ~/AkhilaDison/python$ python3 1.py
['Python is a high-level general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small- and large-scale projects.', 'Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.', 'Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.']
stud@debian: ~/AkhilaDison/python$
```



**Sl no.:-37**

**Aim:-**Write a Python program to read each row from a given csv file and print a list of strings.

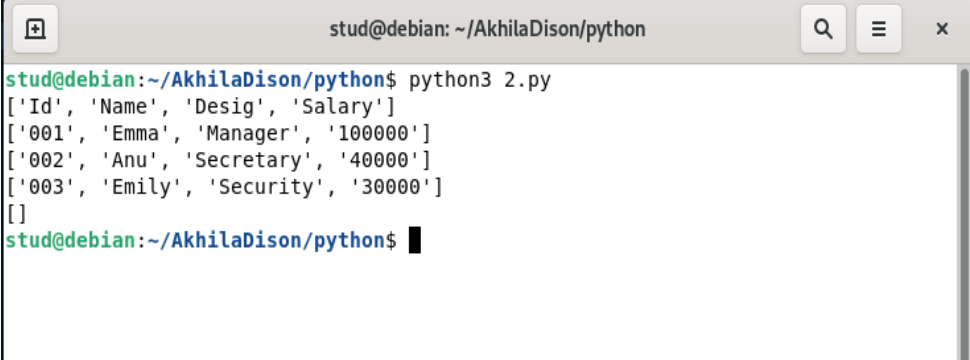
**Program code:-**

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

**test.csv**

```
Id,Name,Desig,Salary
001,Emma,Manager,100000
002,Anu,Secretary,40000
003,Emily,Security,30000
```

**Output:-**



The screenshot shows a terminal window titled 'stud@debian: ~/AkhilaDison/python'. The user has executed the command 'python3 2.py'. The output of the script is displayed as a list of lists, where each inner list represents a row from the CSV file. The first row is the header: ['Id', 'Name', 'Desig', 'Salary']. The subsequent rows are: ['001', 'Emma', 'Manager', '100000'], ['002', 'Anu', 'Secretary', '40000'], and ['003', 'Emily', 'Security', '30000']. The terminal prompt is now 'stud@debian: ~/AkhilaDison/python\$'.

```
stud@debian:~/AkhilaDison/python$ python3 2.py
['Id', 'Name', 'Desig', 'Salary']
['001', 'Emma', 'Manager', '100000']
['002', 'Anu', 'Secretary', '40000']
['003', 'Emily', 'Security', '30000']
[]
stud@debian:~/AkhilaDison/python$
```