**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)TM**

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



**FOCUS ON EXCELLENCE**

**20MCA131 PROGRAMMING LAB LABORATORY RECORD**

**Name: DEVADARSU JEEVANKUMAR**

**Branch: MASTER OF COMPUTER APPLICATIONS**

**Semester: 1 Batch: A Roll No:51**

**MARCH 2022**

**FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)TM**

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



**FOCUS ON EXCELLENCE**

**CERTIFICATE**

*This is to certify that this is a Bonafede record of the Practical work done by* ***DEVADARSU JEEVANKUMAR*** *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 Signature of H O D

Name: Name:

**Date of University practical examination ………………………**

Signature of Signature of

Internal Examiner External Examiner

**CONTENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl No** | **Date of Experiment** | **Title of the Experiment** | **Page No:** | **Signature of Staff –In – Charge** |
| 1 | 23/10/21 | Display future leap years from current year to a final year entered by user. | 7 |  |
| 2 | 23/10/21 | List comprehensions:   1. Generate positive list of numbers from a given list of integers 2. Square of N numbers 3. Form a list of vowels selected from a given word 4. List ordinal value of each element of a word (Hint: use ord() to get ordinal values) | 8-11 |  |
| 3 | 23/10/21 | Count the occurrences of each word in a line of text. | 12 |  |
| 4 | 23/10/21 | Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead. | 13 |  |
| 5 | 10/11/21 | Store a list of first names. Count the occurrences of ‘a’ within the list | 14 |  |
| 6 | 10/11/21 | 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 | 15-16 |  |
| 7 | 10/11/21 | Get a string from an input string where all occurrences of first character replaced with ‘$’, except first character.  [eg: onion ->oni$n] | 17 |  |
| 8 | 10/11/21 | Create a string from given string where first and last characters exchanged. [eg: python - >nythop] | 18 |  |
| 9 | 10/11/21 | Accept the radius from user and find area of circle. | 19 |  |
| 10 | 11/11/21 | Find biggest of 3 numbers entered. | 20 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl No** | **Date of Experiment** | **Title of the Experiment** | **Page No:** | **Signature of Staff –In – Charge** |
| 11 | 11/11/21 | Accept a file name from user and print extension of that. | 21 |  |
| 12 | 11/11/21 | Create a list of colors from comma-separated color names entered by user. Display first and last colors. | **22** |  |
| 13 | 11/11/21 | Accept an integer n and compute n+nn+nnn. | **23** |  |
| 14 | 11/11/21 | Print out all colors from color-list1 not contained in color-list2. | **24** |  |
| 15 | 17/11/21 | Create a single string separated with space from two strings by swapping the character at position 1. | **25** |  |
| 16 | 17/11/21 | Sort dictionary in ascending and descending order. | **26** |  |
| 17 | 17/11/21 | Merge two dictionaries. | **27** |  |
| 18 | 17/11/21 | Find gcd of 2 numbers. | **28** |  |
| 19 | 17/11/21 | From a list of integers, create a list removing even numbers. | **29** |  |
| 20 | 25/11/21 | Program to find the factorial of a number. | **30** |  |
| 21 | 25/11/21 | Generate Fibonacci series of N terms. | **31-32** |  |
| 22 | 25/11/21 | Find the sum of all items in a list | **33** |  |
| 23 | 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. | **34-35** |  |
| 24 | 2/12/21 | Display the given pyramid with step number accepted from user.  Eg: N=4  1   1. 4 2. 6 9   8 12 16 | **36** |  |
| 25 | 2/12/21 | Count the number of characters (character frequency) in a string. | **37** |  |
| 26 | 2/12/21 | Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add ‘ly’ | **38** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl No** | **Date of Experiment** | **Title of the Experiment** | **Page No:** | **Signature of Staff –In – Charge** |
| 27 | 9/12/21 | Accept a list of words and return length of longest word. | 39 |  |
| 28 | 9/12/21 | Construct following pattern using nested loop  \*   * \* * \* \* * \* \* \* * \* \* \* \* * \* \* \* * \* \* * \* | 40-41 |  |
| 29 | 9/12/21 | Generate all factors of a number. | 42 |  |
| 30 | 29/1/22 | 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) | 43-47 |  |
| 31 | 13/1/22 | Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area. | 48-49 |  |
| 32 | 13/1/22 | 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. | 50-51 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl No | Date of Experiment | Title of the Experiment | Page No: | Signature of Staff –In – Charge |
| 33 | 13/1/22 | Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles. | 52-53 |  |
| 34 | 20/1/22 | Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time. | 54-55 |  |
| 35 | 20/1/22 | 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. | 55-57 |  |
| 36 | 3/2/22 | Write a Python program to read a file line by line and store it into a list. | 58 |  |
| 37 | 3/2/22 | Write a Python program to read each row from a given csv file and print a list of string. | 59 |  |

# Course Outcome 1 Programs

**PROGRAM 1**

**AIM**

Display future leap years from current year to a final year entered by user.

## PROGRAM

print ("Enter last year")

endYear = int(input())

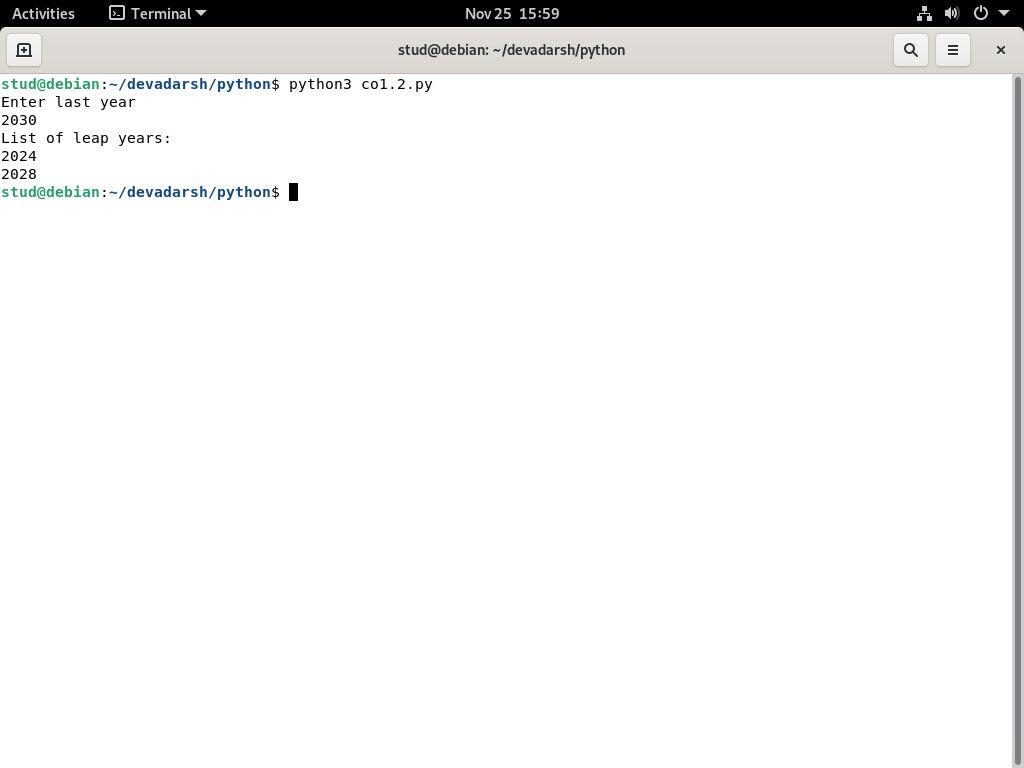
print ("List of leap years:")

for year in range(2021, endYear+1):

if (year % 4 == 0 ) and (year % 100 != 0) or (year % 400 == 0):

print (year)

**OUTPUT**



## PROGRAM 2(a)

## AIM

List comprehensions:

Generate positive list of numbers from a given list of integers

## PROGRAM

list1 = []

n = int(input("enter number of elements:"))

print("enter numbers:")

for i in range(0, n):

element = int(input())

list1.append(element)

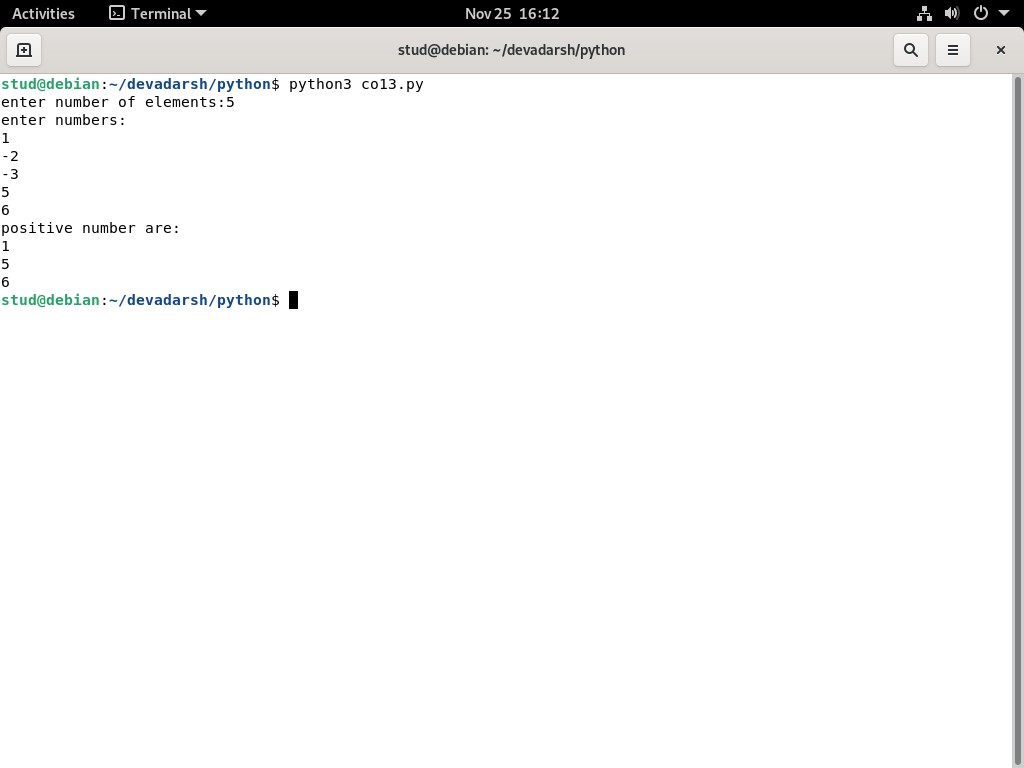
print("positive number are:")

for num in list1:

if num >= 0:

print(num)

## OUTPUT



**PROGRAM 2(b)**

**AIM**

Square of N numbers

## PROGRAM

list1 = []

n = int(input("enter number of elements:"))

print("enter the number to be squared") for i in range(0, n):

element = int(input())

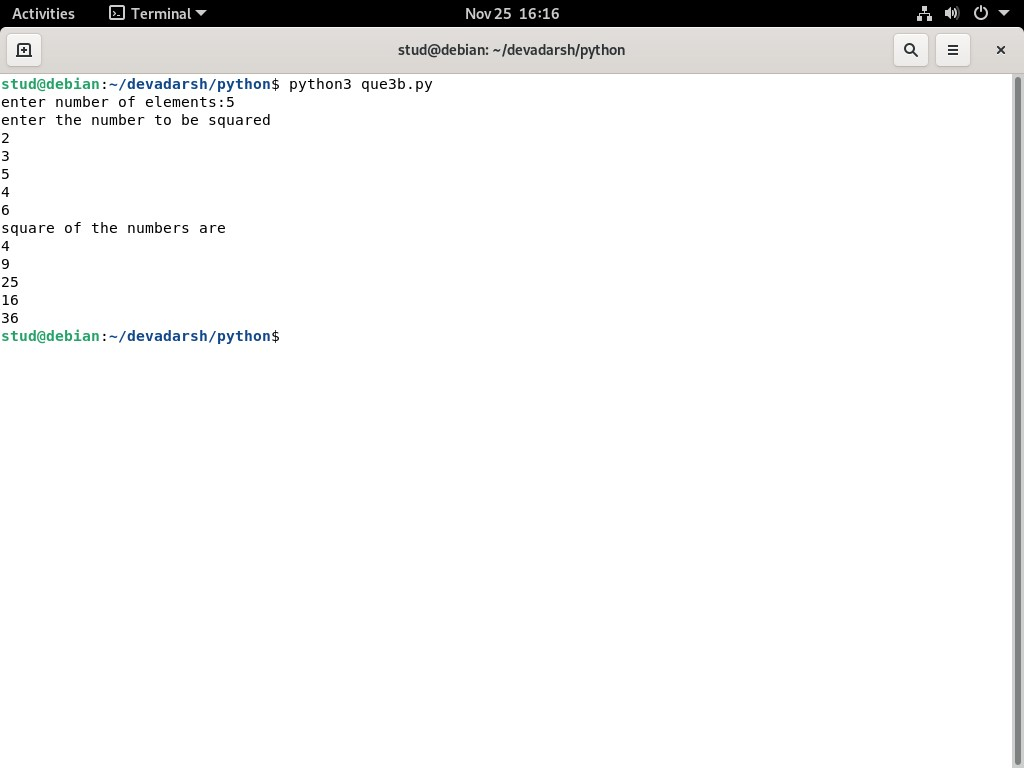
list1.append(element)

print("square of the numbers are")

for num in list1:

print(num\*num)

## OUTPUT



**PROGRAM 2(c)**

**AIM**

Form a list of vowels selected from a given word

## PROGRAM

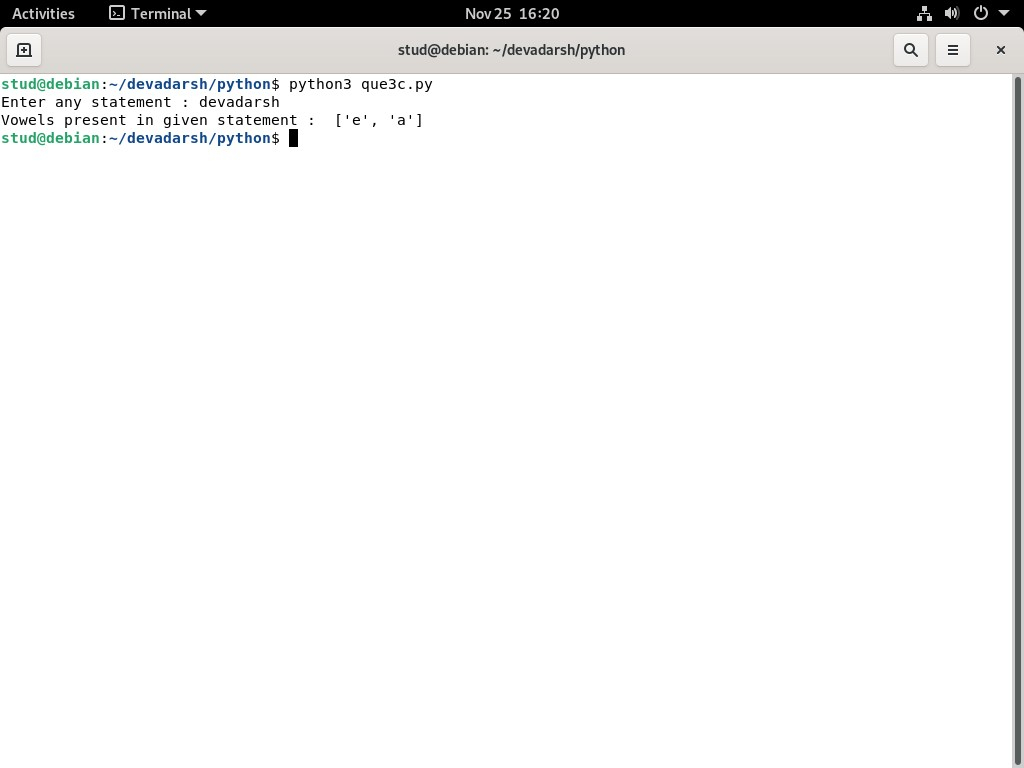
word=input("Enter any statement:")

vo=[]

vowels=['a','e','i','o','u','A','E','I','O','U']

vo=[i for i in word if i in vowels and i not in vo] print(“vowels present in given statement : ”,vo)

## OUTPUT



## PROGRAM 2(d)

## AIM

List ordinal value of each element of a word (Hint: use ord() to get ordinal values)

## PROGRAM

word=input('Enter the string:')

print([ord(x) for x in word])

## OUTPUT



**PROGRAM 3**

**AIM**

Count the occurrences of each word in a line of text.

## PROGRAM

s=input("Enter a string:")

count=dict()

w=s.split()

for i in w:

if i not in count:

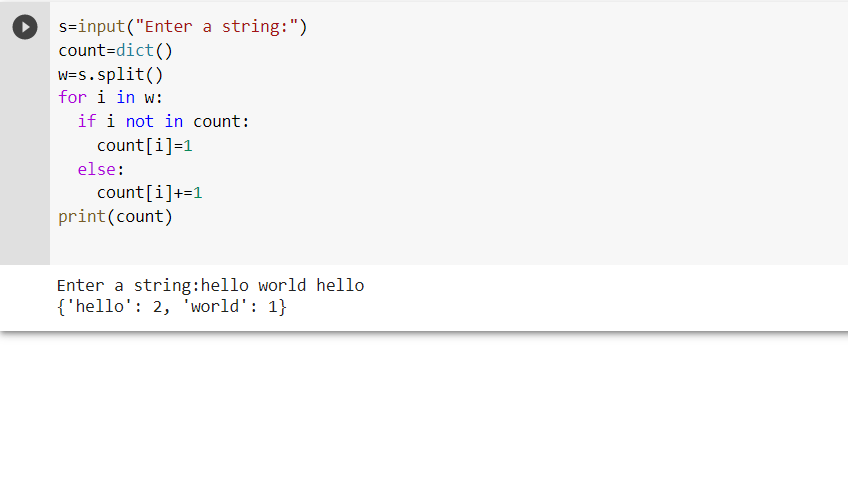
count[i]=1

else:

count[i]+=1

print(count)

## OUTPUT



**PROGRAM 4**

## AIM

Prompt the user for a list of integers. For all values greater than 100, store ‘over’ instead.

## PROGRAM

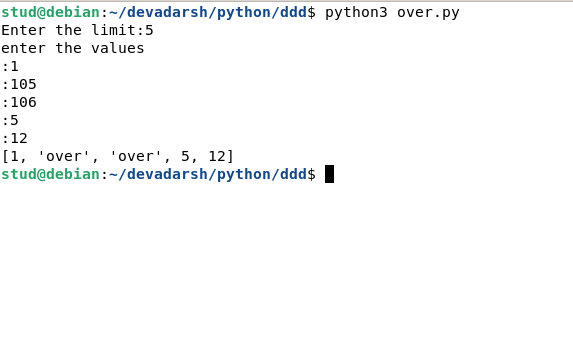
n=int(input("Enter the limit:")) print("Enter the elements:") a=[] for i in range(0,n): x=int(input())

a.append(x)

odd= ['over' if i>100 else i for i in a]

print(odd)

## OUTPUT



**PROGRAM 5**

**AIM**

Store a list of first names. Count the occurrences of ‘a’ within the list

## PROGRAM

## word=['anna','anu']

## r=0

## for d in word:

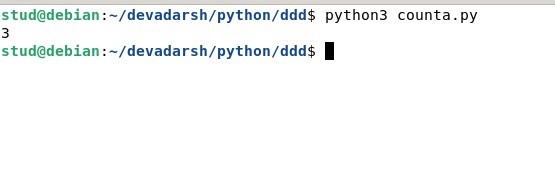
## for c in d:

## if (c=='a'):

## r=r+1

## print(r)

## OUTPUT



**PROGRAM 6**

**AIM**

Enter 2 lists of integers. Check

(a) Whether list is of same length

(b) Whether list sums to same value

(c) Whether any value occur in both

## PROGRAM

l1=[1,2,3,4,5]

l2=[6,3,21,6,5]

p=len(l1)

q=len(l2)

l3=[]

if(p==q):

print("The length of two lists are same")

else:

print("The length of lists are not same")

s=0

p=0

for i in l1:

s=s+i

print("Sum of list1 is",s)

for r in l2:

p=p+r

print("Sum of list2 is",p)

if(s==p):

print("Sum of elements in two lists are same")

else:

print("Sum of elements in two lists are not same")

f=0

for i in l1:

if i in l2:

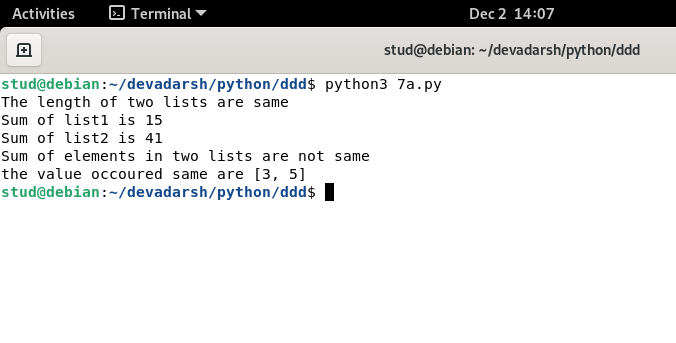
l3.append(i)

f=f+1

print('the value occoured same are',l3)

if(f==0):

print("no element is same")

**OUTPUT**

**PROGRAM 7**

**AIM**

Get a string from an input string where all occurrences of first character replaced with

‘$’, except first character.

[eg: onion -> oni$n]

## PROGRAM

s=input("enter a string:")

print("original string",s)

char=s[0]

s=s.replace(char,'$')

s=char+s[1: ]

print('relpaced string:',s)

## OUTPUT



**PROGRAM 8**

**AIM**

Create a string from given string where first and last characters exchanged.

[eg: python -> nythop]

## PROGRAM

s=input("enter the string:")

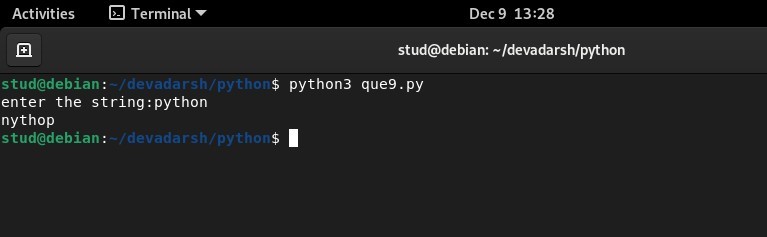
t=s[0] t1=s[-1]

n=len(s)

ns=t1+s[1: n-1]+t

print(ns)

## OUTPUT



**PROGRAM 9**

**AIM**

Accept the radius from user and find area of circle.

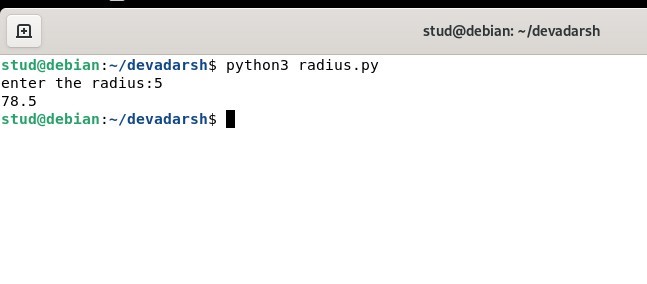
## PROGRAM

r=int(input('enter the radius:'))

A=3.14\*r\*r

print(A)

## OUTPUT



**PROGRAM 10**

**AIM**

Find biggest of 3 numbers entered.

## PROGRAM

a=int(input('enter the first number:'))

b=int(input('enter the second number:'))

c=int(input('enter the first number:'))

if(a>b):

if(a>c):

print(a)

else:

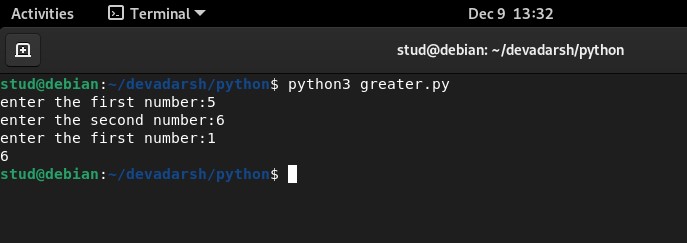
if(b>c):

print(b)

else:

print(c)

## OUTPUT



**PROGRAM 11**

**AIM**

Accept a file name from user and print extension of that.

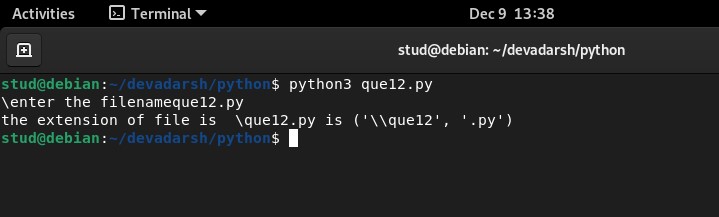
## PROGRAM

import os

a=input("enter the filename")

print("the extension of file is ",a,"is",os.path.splitext(a))

## OUTPUT



**PROGRAM 12**

**AIM**

Create a list of colors from comma-separated color names entered by user. Display first and last colors.

## PROGRAM

color=”red,green,blue,black”

seperate=color.split(",")

print("First color:",seperate[0])

print("Last color:",seperate[-1])

## OUTPUT



**PROGRAM 13**

**AIM**

Accept an integer n and compute n+nn+nnn.

## PROGRAM

x=int(input("enter the numbers"))

a=str(x)

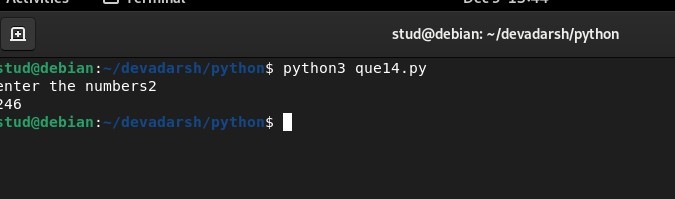
b=a+a

c=a+a+a

d=x+int(b)+int(c)

print(d)

## OUTPUT



**PROGRAM 14**

**AIM**

Print out all colors from color-list1 not contained in color-list2.

## PROGRAM

a=['red','blue','green','grey','black']

b=['red','blue','green']

flag=0

print("colours not contained in b are:\t")

for i in a:

i not in b:

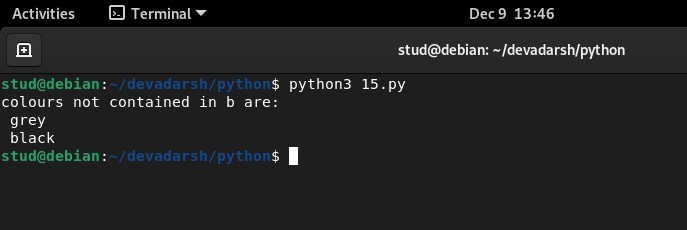
print("",i)

flag=flag+1

if(flag==0):

print("there is no common colours ")

## OUTPUT



**PROGRAM 15**

**AIM**

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

## PROGRAM

str1=input("enter 1st string:")

str2=input("enter 2nd string:")

str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]

print(str3)

## OUTPUT



**PROGRAM 16**

**AIM**

Sort dictionary in ascending and descending order.

## PROGRAM

d1={"dev":1,"kunjuz":3,"ammu":2,"rishy":4}

l=list(d1.items())

print("orginal list is",l)

l.sort()

print("Ascending order is\n",l)

l=list(d1.items())

l.sort(reverse=True)

print("Desencding order is\n",l)

## OUTPUT



**PROGRAM 17**

**AIM**

Merge two dictionaries.

## PROGRAM

d1={'name': "dev" , 'age': 22}

d2={'qlfn':"pg"}

d1.update(d2)

print(d1)

## OUTPUT



**PROGRAM 18**

**AIM**

Find GCD of 2 numbers.

## PROGRAM

a=int(input("enter the number"))

b=int(input("enter the number"))

d=min(a,b)

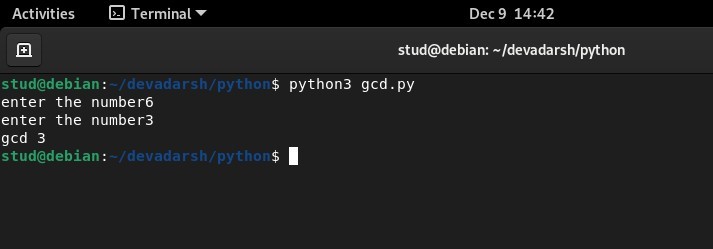
for i in range(1,d+1):

if(a%i==0)and(b%i==0):

hcf=i

print("gcd",hcf)

## OUTPUT



**PROGRAM 19**

**AIM**

From a list of integers, create a list removing even numbers.

## PROGRAM

l1=[1,2,3,4,5] a=[] b=[]

for i in l1:

if(i%2==0):

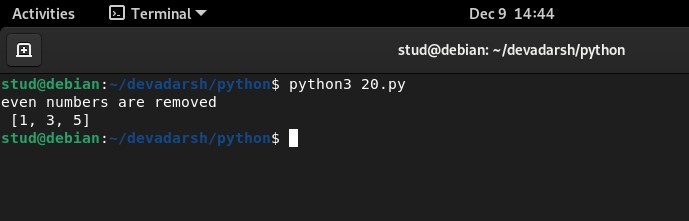
a.append(i)

else:

b.append(i)

print('even numbers are removed\n',b)

## OUTPUT



**Course Outcome 2 Programs**

**PROGRAM 20**

**AIM**

Program to find the factorial of a number.

## PROGRAM

n=int(input("enter the number"))

f=1

for i in range(1,n+1):

f=f\*i

print(f)

## OUTPUT

**PROGRAM 21**

**AIM**

Generate Fibonacci series of N terms.

## PROGRAM

n = int(input("How many terms? "))

a, b = 0, 1

count = 0

if n <= 0:

print("Please enter a positive integer")

elif n == 1:

print("Fibonacci sequence upto",n,":")

print(a)

else:

print("Fibonacci sequence:")

while count < n:

print(a)

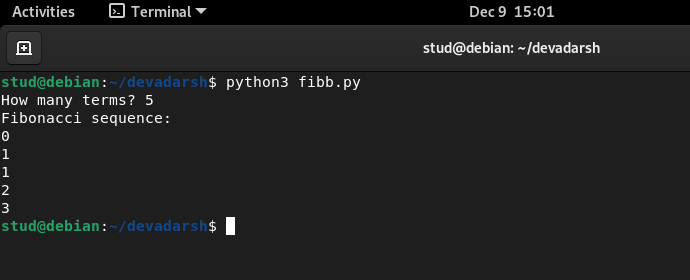
nth = a + b

a = b

b = nth

count += 1

## OUTPUT



**PROGRAM 22**

**AIM**

Find the sum of all items in a list.

## PROGRAM

l1=[1,2,3,4,5]

l3=[]

s=0

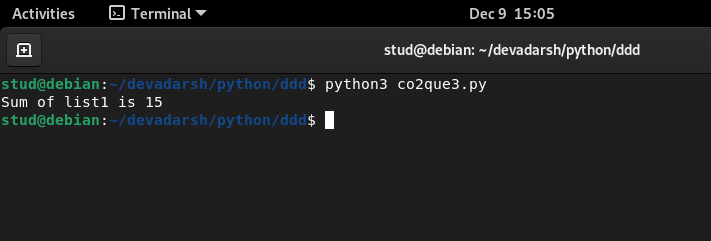
p=0

for i in l1:

s=s+i

print("Sum of list1 is",s)

## OUTPUT



**PROGRAM 23**

## AIM

Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

## PROGRAM

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):

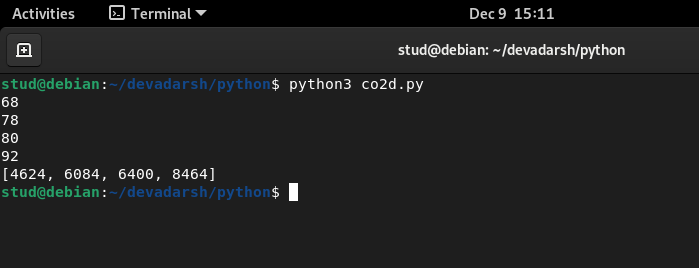
if((k\*\*2)==j):

list1.append(j)

print(k)

print(list1)

## OUTPUT



**PROGRAM 24**

**AIM**

Display the given pyramid with step number accepted from user.

Eg: N=4

1

1. 4
2. 6 9
3. 8 12 16

## PROGRAM

n=int(input('enter the number of steps'))

for i in range(1,n+1):

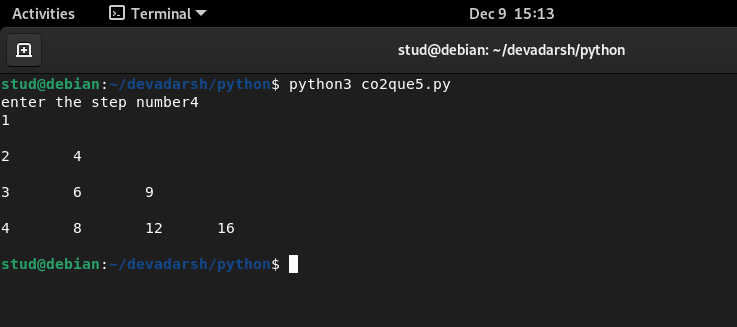
for j in range(1,i+1):

s=i\*j

print(s,'\t',end="")

print("\n")

## OUTPUT



**PROGRAM 25**

**AIM**

Count the number of characters (character frequency) in a string.

## PROGRAM

str=input("Enter a string:")

fnd=input("Enter character:")

cnt=0

fnd=fnd.lower()

str=str.lower()

for i in str:

if i==fnd:

cnt=cnt+1

print("Freq:->",cnt)

## OUTPUT

**PROGRAM 26**

**AIM**

Add ‘ing’ at the end of a given string. If it already ends with ‘ing’, then add

‘ly’.

## PROGRAM

a=input("Enter a word\n")

l=len(a)

ll=a[l-3:l]

if(ll=="ing"):

s=a+"ly"

else:

s=a+"ing"

print (s)

## OUTPUT

**PROGRAM 27**

**AIM**

Accept a list of words and return length of longest word.

## PROGRAM

list=[]

length=[]

print("enter 5 words")

for i in range (5):

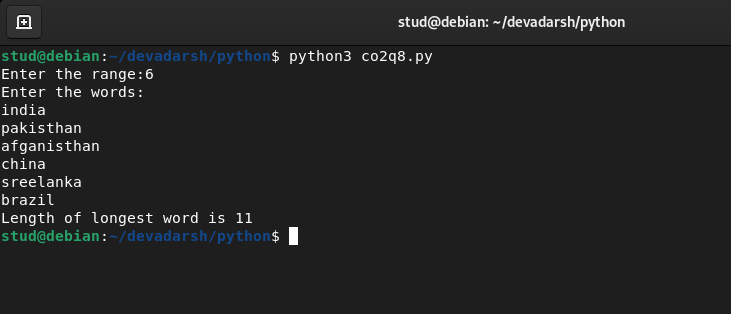
str=input()

list.append(str)

for j in list:

length.append(len(j))

print("length of longest word is:",max(length))

**OUTPUT**

**PROGRAM 28**

**AIM**

Construct following pattern using nested loop

\*

* \*
* \* \*
* \* \* \*
* \* \* \* \*
* \* \* \*
* \* \*
* \*

\*

## PROGRAM

for i in range(1,6):

for j in range(1,i+1):

print("\*",end="")

print("\n")

for i in range(5,0,-1):

for j in range(1,i-1):

print("\*",end="")

print("\n")

## OUTPUT

**PROGRAM 29**

**AIM**

Generate all factors of a number.

## PROGRAM

def print\_factors(x):

print("the factors of",x,"are:\n")

for i in range(2,int(x/2)+1):

if x%i==0:

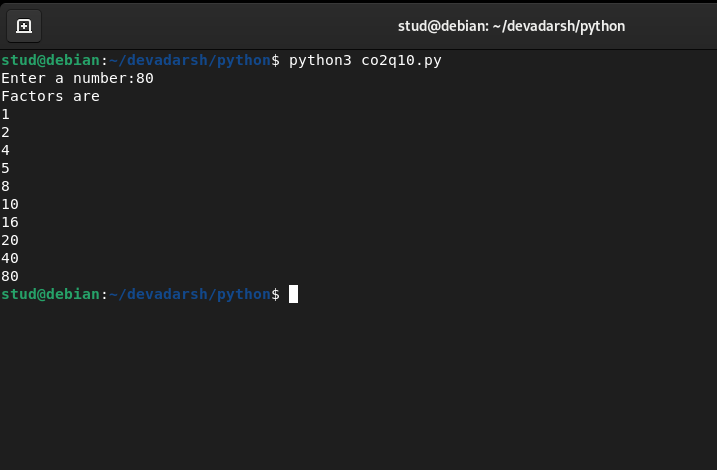
print(i)

num=20

print("number is",num)

print\_factors(num)

## OUTPUT



# Course Outcome 3 Programs

## PROGRAM 30

## AIM

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)

## PROGRAM

# Graphice\circle.py

from math import pi

def area\_circle(radius): return pi\*radius\*radius

def perimeter\_circle(radius): return 2\*pi\*radius

# Graphics\rectangle.py

def area\_rec(length,width): return length\*width

def perimeter\_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 perimeter\_sphere(radius): return 2\*pi\*radius

# Graphics.py (driver code)

import Graphics

from Graphics import circle,rectangle

from Graphics.tdgraphics import cuboid,sphere from Graphics.circle import \*

print("Area of a circle with radius 10 is : ",circle.area\_circle(10)) print("Permeter of a circle with radius 10 is ",circle.perimeter\_circle(10)) print("\n")

print("Area of a Rectangle with length and width 10 is : ",rectangle.area\_rec(10,10)) print("Permeter of a Rectangle with length and width 10 is : ",rectangle.perimeter\_rec(10,10)) print("\n")

print("Area of a cuboid with length,width,height 10 is : ",cuboid.area\_cuboid(10,10,10)) print("Volume of a cuboid with length,width,height 10 is : ",cuboid.volume\_cuboid(10,10,10)) print("\n")

print("Area of a spere with radius 10 is : ",sphere.area\_sphere(10)) print("Permeter of a spere with radius 10 is ",sphere.perimeter\_sphere(10))

## OUTPUT



# Course Outcome 4 Programs

## PROGRAM 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

class Rectangle:

def create (self,l,b):

self.l=l

self.b=b

def area(self):

a=(self.l\*self.b)

print("area=",a)

return a

def perimeter(self):

b= 2\*(self.l+self.b)

print("perimeter=",b)

return b

l=int(input("enter the length"))

b=int(input("enter the breadth"))

r1=Rectangle()

r1.create(l,b)

l=int(input("enter the length"))

b=int(input("enter the breadth"))

r2=Rectangle()

r2.create(l,b)

x=r1.area()

y=r2.area()

z=r1.perimeter()

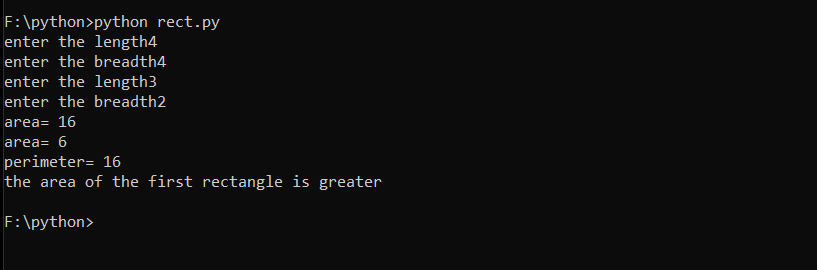
if(x>y):

print("the area of the first rectangle is greater")

else:

print("the area of the second rectangle is greater")

## OUTPUT



## PROGRAM 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

class Bank:

def \_\_init\_\_(self,ano,aname,actype,bal):

self.ano=ano

self.aname=aname

self.actype=actype

self.bal=bal

def withdraw(self,c):

self.bal=(self.bal-c)

print("accound holders name=",self.aname)

print("balance after withdraw=",self.bal)

return self.bal

def deposit(self,a):

self.bal=(self.bal+a)

print("accound holders name=",self.aname)

print("balance after deposit=",self.bal)

return self.bal

b1=Bank(11112,'dev','current',50000)

b1.deposit(100)

b1.withdraw(100)

b2=Bank(11113,'rishy','salary',750)

b2.deposit(150)

b2.withdraw(800)

## OUTPUT



## PROGRAM 33

## AIM

Create a class Rectangle with private attributes length and width. Overload ‘<’ operator to compare the area of 2 rectangles.

## PROGRAM

class Rectangle:

def create (self,l,b):

self.l=l

self.b=b

def area(self):

a=(self.l\*self.b)

print("area=",a)

return a

def perimeter(self):

b= 2\*(self.l+self.b)

print("perimeter=",b)

return b

def \_\_lt\_\_(self,rr):

if(self.b\*self.l>rr.b\*rr.l):

print("the area of the first rectangle is greater")

return True

else:

print("the area of the second rectangle is greater")

return False

l=int(input("enter the length"))

b=int(input("enter the breadth"))

r1=Rectangle()

r1.create(l,b)

l=int(input("enter the length"))

b=int(input("enter the breadth"))

r2=Rectangle()

r2.create(l,b)

x=r1.area()

y=r2.area()

z=r1.perimeter()

r1 > r2

## OUTPUT

## PROGRAM 34

## AIM

Create a class Time with private attributes hour, minute and second. Overload ‘+’ operator to find sum of 2 time.

## PROGRAM

class Time:

def \_\_init\_\_(self,hr,min,sec):

self.\_\_hr=hr

self.\_\_min=min

self.\_\_sec=sec

def \_\_add\_\_(self,t2):

self.\_\_hr=self.\_\_hr+t2.\_\_hr

self.\_\_min=self.\_\_min+t2.\_\_min

self.\_\_sec=self.\_\_sec+t2.\_\_sec

if (self.\_\_sec>60):

self.\_\_sec-=60

self.\_\_min+=1

if(self.\_\_min>60):

self.\_\_min-=60

self.\_\_hr+=1

if(self.\_\_hr>12):

self.\_\_hr-=12

print("Total time is",self.\_\_hr,":",self.\_\_min,":",self.\_\_sec)

t1=Time(3,55,56)

t2=Time(2,24,8)

T1+t2

## OUTPUT



## PROGRAM 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 overriding.

## PROGRAM

class Publisher(object):

def \_\_init\_\_(self,name):

self.name=name

def display1(self):

print(self.title)

print(self.author)

class Book(Publisher):

def \_\_init\_\_(self,name,title,author):

super().\_\_init\_\_(name)

self.title=title

self.author=author

def display2(self):

super().display1()

print(self.title)

print(self.author)

class Python(Book):

def \_\_init\_\_(self,name,title,author,price,no\_of\_pages):

super().\_\_init\_\_(name,title,author)

self.price=price

self.no\_of\_pages=no\_of\_pages

def display3(self):

super().display2()

print(self.price)

print(self.no\_of\_pages)

p=Python("HHH Publications","The Python","Bivina",140,600)

p.display3()

## OUTPUT



# Course Outcome 5 Programs

**PROGRAM 36**

**AIM**

Write a Python program to read a file line by line and store it into a list.

## PROGRAM

fp=open("text\_file.txt",'r')

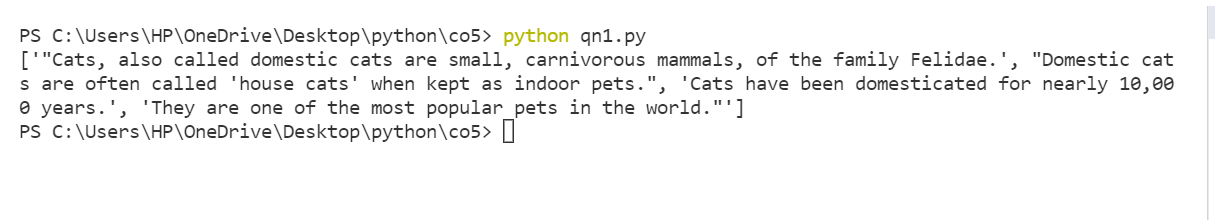
lines=[]

for line in fp:

lines.append(line.strip())

print(lines)

## OUTPUT



## PROGRAM 37

## AIM

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

## PROGRAM

import csv

with open('people.csv', 'r') as file:

reader = csv.reader(file)

for row in reader:

print(row)

## OUTPUT

## 