PGMM1

AIM:

Write a program to determine whether a given year is a leap year [ Use Calendar Module]

SOURCECODE:

import calendar

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

if calendar.isleap(n):

print(f"{n} is leap year")

else:

print(f"{n} is not a leap year")

OUPUT





PGMM2

AIM:

Write a python script to display

a) Current date and time

b) Current Year

c) Month of the year

d) Week number of the year

e) Weekday of the week

f) Day of year

g) Day of the month

h) Day of week [ Use time and datetime Module]

SOURCECODE:

from datetime import datetime

cur=datetime.now()

print("current date and time:",cur)

print("current year:",cur.year)

print("month of the year:",cur.strftime("%B"))

print("Week no of the year:",cur.strftime("%U"))

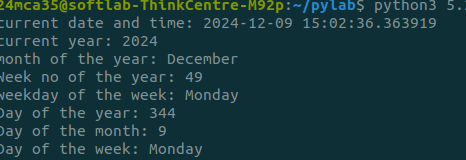
print("weekday of the week:",cur.strftime("%A"))

print("Day of the year:",cur.strftime("%j"))

print("Day of the month:",cur.day)

print("Day of the week:",cur.strftime("%A"))

OUTPUT



PGMM3

AIM:

Write a python program to print yesterday, today and tomorrow.

SOURCECODE

import datetime as m

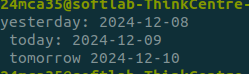
today=m.date.today()

yesterday=today-m.timedelta(days=1)

tomorrow=today+m.timedelta(days=1)

print("yesterday:",yesterday,"\n today:",today,"\n tomorrow",tomorrow)

OUTPUT



PGMM 4

AIM:

Write a function in file palindrome.py to check whether a string is Palindrome or not. Import the module to find the longest palindromic substring in a given string by checking every possible substring and verifying if it is a palindrome

SOURCECODE

rom sub import palindrome

def long\_palindrome(s):

longest=""

for i in range(len(s)):

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

substring=s[i:j]

if palindrome(substring) and len(substring)>len(longe>

longest=substring

return longest

str1=input("Enter string:")

result=long\_palindrome(str1)

print("longest palindrome substring:",result)

OUTPUT



PGMM5

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 find the area and perimeter of figures by different importing statements. (Include selective import of modules and import \* statements)

SOURCECODE

pgmm.py

import graphics.circle as c

import graphics.tgraphics.cuboid as cu

import graphics.tgraphics.sphere as s

len=int(input("Enter length of rectangle:"))

bread=int(input("Enter breadth of rectangle:"))

print("Area of rectangle",r.area(len,bread))

print("Perimeter of rectangle",r.perimeter(len,bread))

r1=int(input("Enter area of the circle:"))

print("Area of circle",c.area(r1))

print("Perimeter of circle",c.circumference(r1))

r2=int(input("Enter radius of the sphere:"))

print("Area of sphere:",s.area(r2))

print("Volume of sphere:",s.volume(r2))len1=int(input("\nEnter the length of the cuboid:"))

bre1=int(input("Enter the breadth of the cuboid:"))

hei1=int(input("Enter the height of the cuboid:"))

print("Area of cuboid=",cu.area(len1,bre1,hei1))

print("Perimeter of cuboid=",cu.perimeter(len1,bre1,hei1))

rectangle.py

def area(l,b):

return l\*b

def perimeter(l,b):

return 2\*(l+b)

circle.py

import math

def area(r):

return math.pi\*r\*r

def circumference(r):

return 2\*math.pi\*r

cuboid.py

def area(l,w,h):

return 2\*(l\*w+w\*h+l\*w)

def perimeter(l,w,h):

return 4\*(l+w+h)

sphere.py

import math

def volume(r):

return(4/3)\*math.pi\*(r\*\*3)

def area(r):

return 4\*math.pi\*r\*r

OUTPUT

