**LABCYCLE 5**

**EXPERIMENT 1**

**AIM:**

Write a program to determine whether given year is a leap year(use calender module).

SOURCE CODE:

import calendar

year=int(input("enter a year :"))

if calendar.isleap(year):

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

else:

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

OUTPUT:





EXPERIMENT 2

AIM:

Write a python script to display

a.Current data and time

b.Current year

c.Month of the year

d.Week number of the year

e.Week day of the week

f.Day of year

g.Day of the month

h.Day of week(use time/datetime module).

SOURCE CODE:

from datetime import datetime,date

import calendar

today=datetime.now()

print("Current date and time :",today)

yr=date.today()

print("Current year: ",yr.year)

print("Month of the year: ")

print(calendar.month(yr.year,yr.month))

print("Week number of the year: ",yr.strftime("%W"))

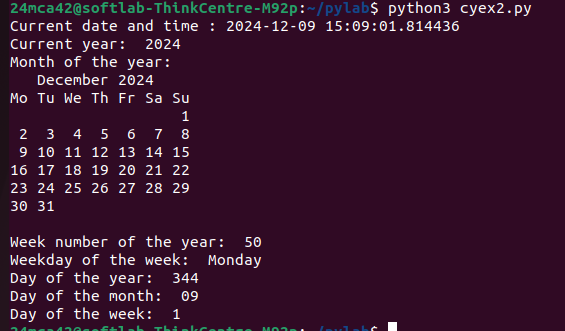
print("Weekday of the week: ",yr.strftime("%A"))

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

print("Day of the month: ",yr.strftime("%d"))

print("Day of the week: ",yr.strftime("%w"))

OUTPUT:



EXPERIMENT 3

AIM:

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

SOURCE CODE:

from datetime import datetime, timedelta

today = datetime.now()

yesterday = today - timedelta(days=1)

tomorrow = today + timedelta(days=1)

print("Yesterday:", yesterday.strftime('%Y-%m-%d'))

print("Today:", today.strftime('%Y-%m-%d'))

print("Tomorrow:", tomorrow.strftime('%Y-%m-%d'))

OUTPUT:



EXPERIMENT 4

AIM:

Wriote a function in the file palindrome.py to check whether a string is palindrome or not.Import the module to find the longest palindromic substring in agiven string by checking every possible substring and verify it is a palindrome.

SOURCE CODE:

def is\_palindrome(s):

return s==s[::-1]

from palindrome import is\_palindrome

def longest(s):

n=len(s)

longest=" "

for i in range(n):

for j in range(i,n):

substring=s[i:j+1]

if is\_palindrome(substring) and len(substring)>len(longest):

longest=substring

return longest

input\_string=input("Enter a string : ")

print("longest palindromic substring : ",longest(input\_string))

OUTPUT:



EXPERIMENT 5

AIM:

Create a package graphics with modules rectangle,circle and sub-package 3D graphics with modules cuboid & 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.

SOURCE CODE:

from graphics.rectangle import area as rect\_area

from graphics.rectangle import perimeter as rect\_perimeter

from graphics.circle import area as circle\_area

from graphics.circle import circumference as circle\_circumference

from graphics.graphics\_3d.cuboid import volume as cuboid\_volume

from graphics.graphics\_3d.cuboid import surface\_area as cuboid\_surface\_area

from graphics.graphics\_3d.sphere import volume as sphere\_volume

from graphics.graphics\_3d.sphere import surface\_area as sphere\_surface\_area

rectangle\_length=float(input("\nEnter the length of the rectangle : "))

rectangle\_breadth=float(input("Enter the breadth of the rectangle : "))

print("Area of the Rectangle : ",rect\_area(rectangle\_length,rectangle\_breadth))

print("Perimeter of the Rectangle : ",rect\_perimeter(rectangle\_length,rectangle\_breadth))

circle\_radius=float(input("\nEnter the radius of the circle : "))

print("Area of the Circle : ",circle\_area(circle\_radius))

print("Circumference of the Circle : ",circle\_circumference(circle\_radius))

cuboid\_length=float(input("\nEnter the length of the cuboid : "))

cuboid\_breadth=float(input("Enter the breadth of the cuboid : "))

cuboid\_height=float(input("Enter the height of the cuboid : "))

print("Volume of the Cuboid : ",cuboid\_volume(cuboid\_length,cuboid\_breadth,cuboid\_height))

print("Surface area of the cuboid : ",cuboid\_surface\_area(cuboid\_length,cuboid\_breadth,cuboid\_height))

sphere\_radius=float(input("\nEnter the radius of the sphere : "))

print("Volume of the Sphere : ",sphere\_volume(sphere\_radius))

print("Surface area of the Sphere : ",sphere\_surface\_area(sphere\_radius))

OUTPUT:

