MAHA BARATHI ENGINEERING COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GE3171

PROBLEM SOLVING AND PYTHON PROGRAMMING LABORATORY MANUAL

REGULATION – 2021

YEAR/ SEMESTER: I/I

ACADEMIC YEAR: 2022-2023

BATCH: 2022-2026

GE3171 PROBLEM SOLVING AND PYTHON PROGRAMMING LABORATORY L T P C 0 0 4 2

COURSE OBJECTIVES:

- To understand the problem solving approaches.
- To learn the basic programming constructs in Python.
- To practice various computing strategies for Python-based solutions to real world problems.
- To use Python data structures-lists, tuples, dictionaries.
- To do input/output with files in Python.

EXPERIMENTS:

Note: The examples suggested in each experiment are only indicative. The lab instructor isexpected to design other problems on similar lines. The Examination shall not be restrictedtothesampleexperimentslisted here.

1. Identification and solving of simple real life or scientific or technical problems, and developing flow charts for the same. (Electricity Billing, Retail shop billing, Sine series, weight of a motor bike, Weight of a steel bar, compute Electrical Current in Three Phase AC Circuit, etc.)
2. Python programming using simple statements and expressions (exchange the values of two variables, circulate the values of n variables, distance between two points).
3. Scientific problems using Conditionals and Iterative loops. (Number series, Number Patterns, pyramid pattern)
4. Implementing real-time/technical applications using Lists, Tuples. (Items present in a library/Components of a car/ Materials required for construction of a building –operations of list& tuples)
5. Implementing real-time/technical applications using Sets, Dictionaries. (Language, components of an automobile, Elements of a civil structure, etc.-operations of Sets &Dictionaries)
6. Implementing programs using Functions.(Factorial, largest number in a list, area of shape)
7. Implementing programs using Strings. (reverse, palindrome, character count, replacing characters)
8. Implementing programs using written modules and Python Standard Libraries (pandas, numpy, Matplotlib, scipy)

9. Implementing real-time/technical applications using File handling. (copy from one file to another, word count, longest word)
10. Implementing real-time/technical applications using Exception handling. (divide by zero error, voter’s age validity, student mark range validation)
11. Exploring Pygame tool.
12. Developing a game activity using Pygame like bouncing ball, car race etc.

TOTAL:60 PERIODS

CONTENTS OF MANUAL

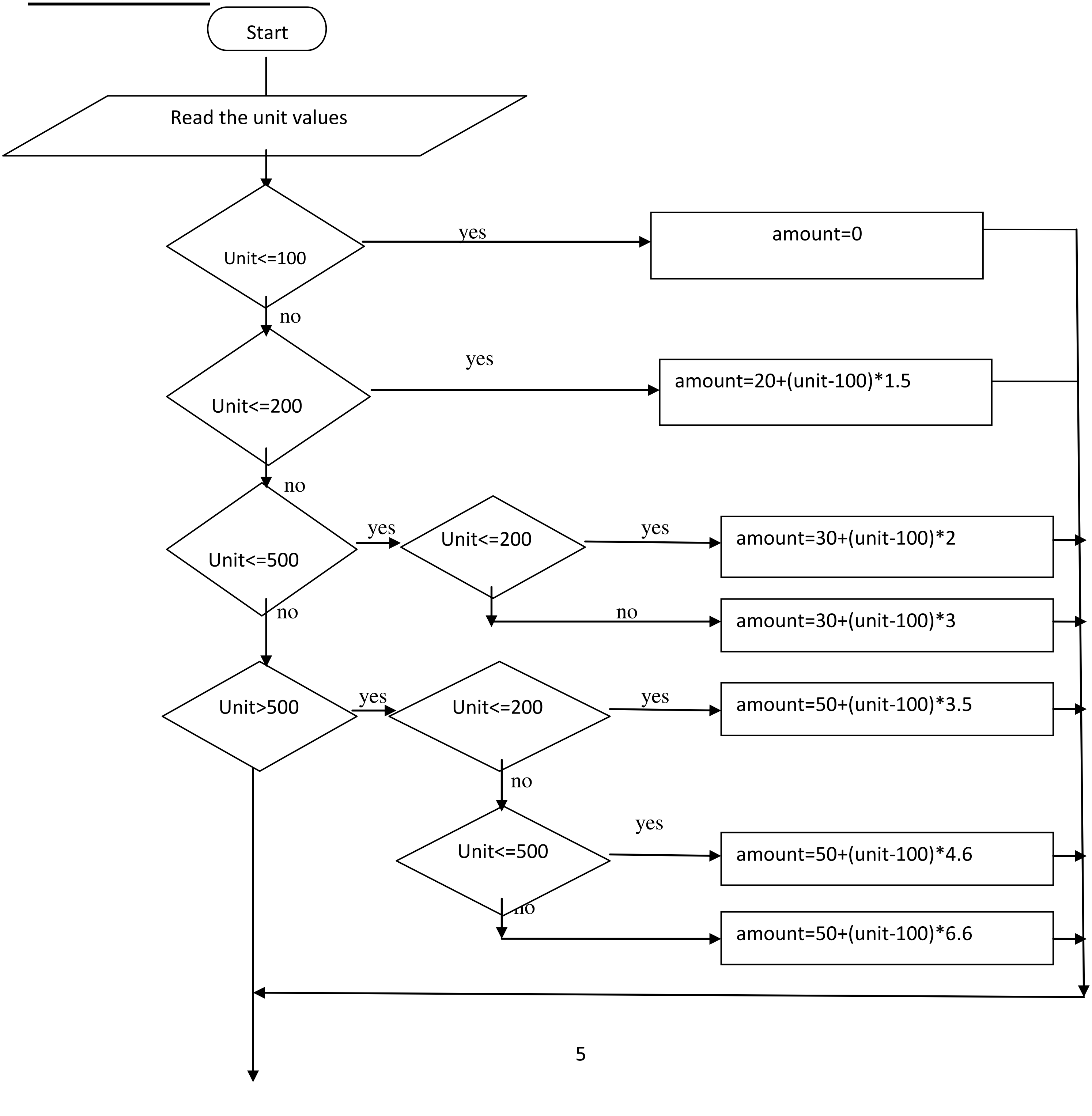
12	Developing a game activity using Pygame like bouncing ball, car race etc.	

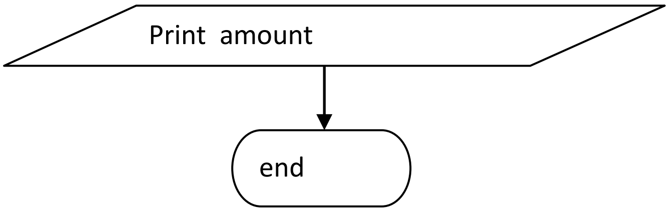
Ex. No : 1(a) **Date :**	**ELECTRICITY BILLING**

PROBLEM STATEMENT:

In real time the electricity billing calculation is of different range with different rates considering the free unit for every range like(100 unit free for every consumer) and within 200 units rate vary accordingly and within 500 units the range is for(1-200 and 201-500) and above 500 the range is for (1-200,201-500 and >500). Fixed rate is different for different range.

FLOWCHART


Start
Read the unit values
Unit<=100
yes
amount=0
no
Unit<=200
yes
amount=20+(unit-100)*1.5
no
Unit<=500
yes
Unit<=200
yes
amount=30+(unit-100)*2
no
no
amount=30+(unit-100)*3
Unit>500
yes
Unit<=200
yes
amount=50+(unit-100)*3.5
no
Unit<=500
yes
amount=50+(unit-100)*4.6
no
amount=50+(unit-100)*6.6



RESULT:

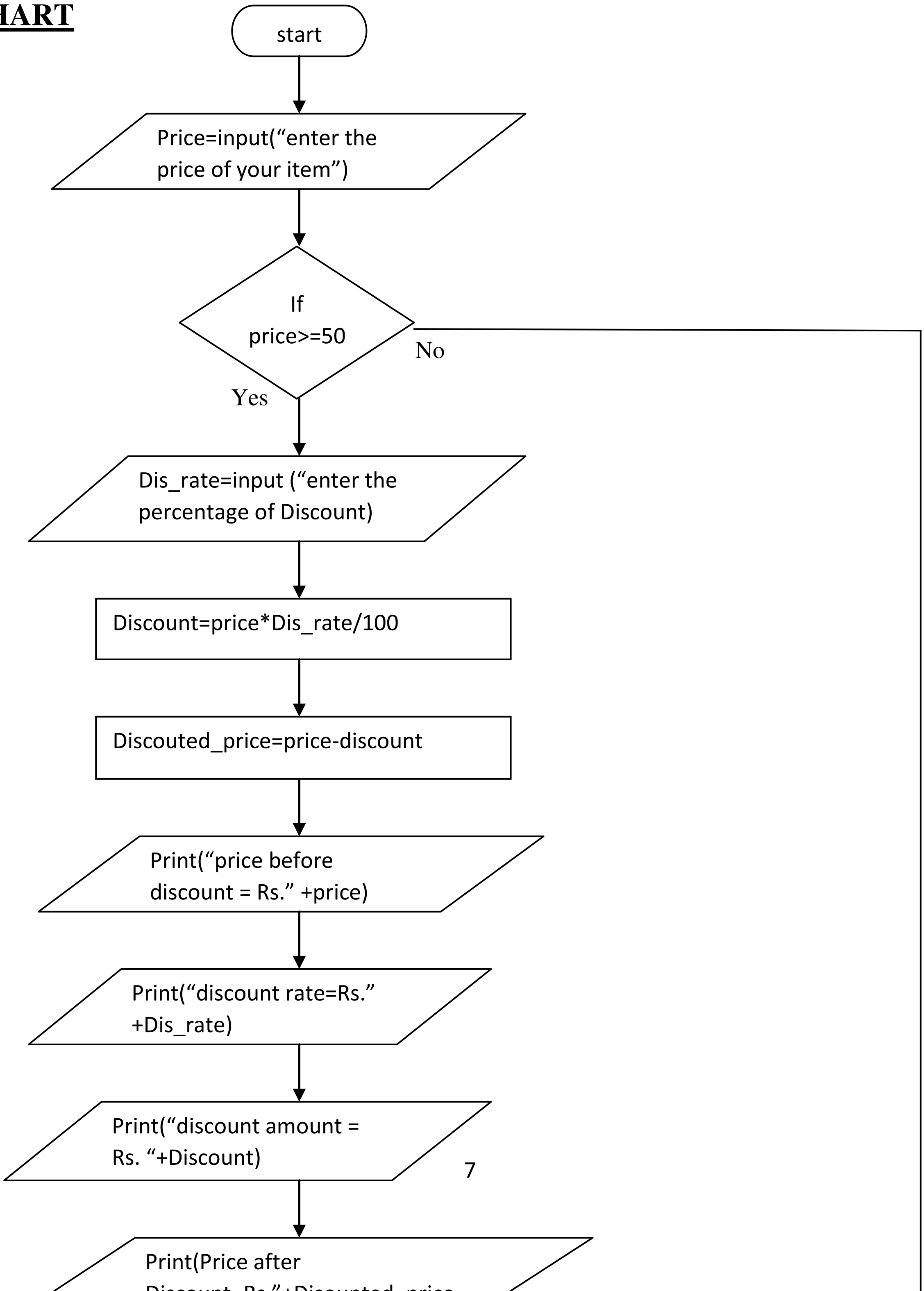
Thus the electricity billing calculation in real time were identified and solved.

Ex. No : 1(b) Date :	RETAILSHOP BILLING

PROBLEM STATEMENT:

In real time the retail shop offers to the customer if customer buys an item above Rs.50.the shop discount amount as % of the item price. first calculate discount rate, discounted amount and discounted price. Finally display the actual price of the item, discount rate andprice after discounted price of the item.

FLOWCHART


start
Price=input(“enter the price of your item”)
If price>=50
No
Yes
Dis_rate=input (“enter the percentage of Discount)
Discount=price*Dis_rate/100
Discouted_price=price-discount
Print(“price before discount = Rs.” +price)
Print(“discount rate=Rs.” +Dis_rate)
Print(“discount amount = Rs. “+Discount)
Print(Price after Discount=Rs.”+Discounted_price

RESULT:

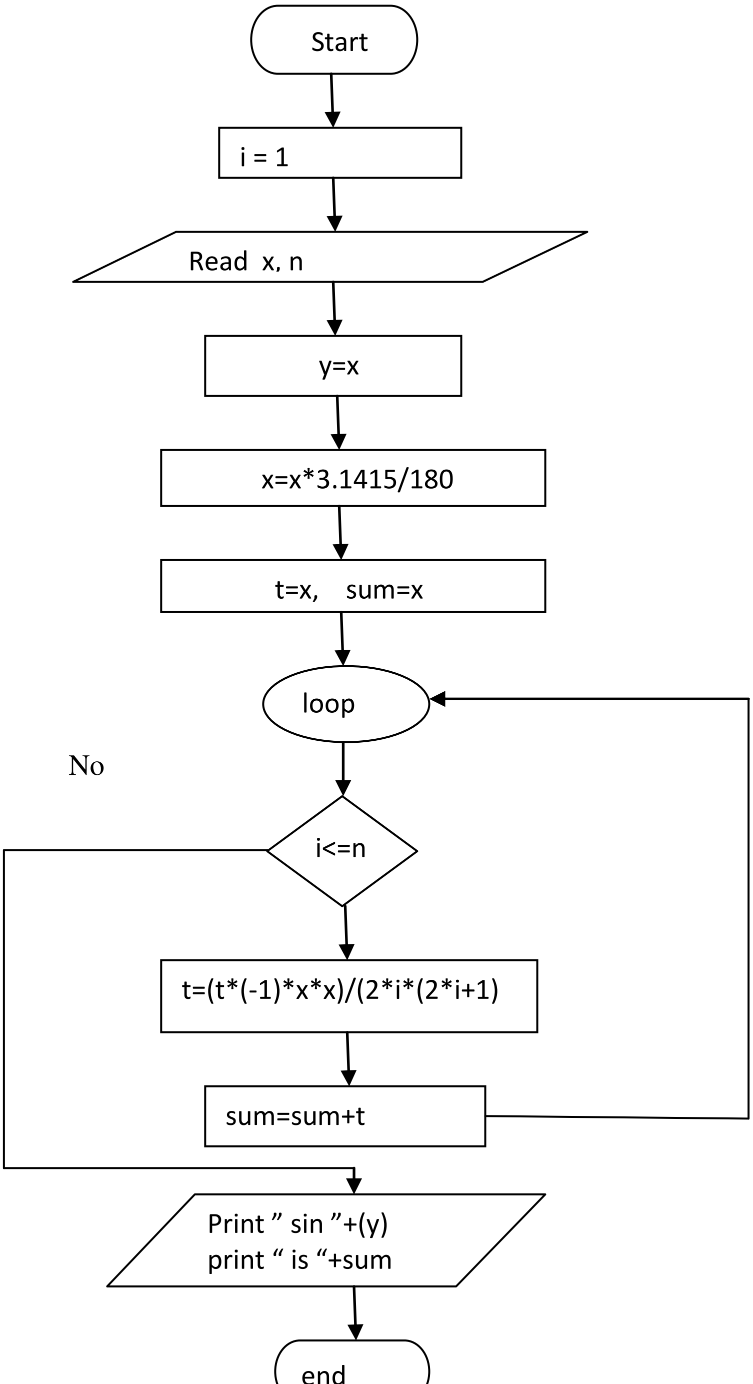
Thus the Retail shop billing calculation in real time were identified and solved.

Ex. No : 1(c) **Date :**	**SIN SERIES**

PROBLEM STATEMENT:

1. In mathematical series, the sin series used to identify and solve scientific problems. In this initialize i=1 and read the inputs as x and n. X is the angle in degree which is converted into radian. Then initialize y =x for finally printing the sine value and Process x=x*3.1415/180 which is Converting ‘x’ to radian value Then It assigns t=x and sum=x, Check the condition if i<=n, then the loop continues till the condition of the loop is true. Calculate the sine series using t=(t*(-1)*x*x)/(2*i*(2*i+1)), calculate the sum, sum= sum + t. Finally print the sine value and sine series value (sum value)

FLOWCHART


Start
i = 1
Read x, n
y=x
x=x*3.1415/180
t=x, sum=x
loop
No
i<=n
t=(t*(-1)*x*x)/(2*i*(2*i+1)
sum=sum+t
Print ” sin ”+(y)
print “ is “+sum
end

RESULT:

Thus the sin series calculation in real time were identified and solved.

Ex. No : 1(d) **Date :**	**WEIGHT OF A STEEL BAR**

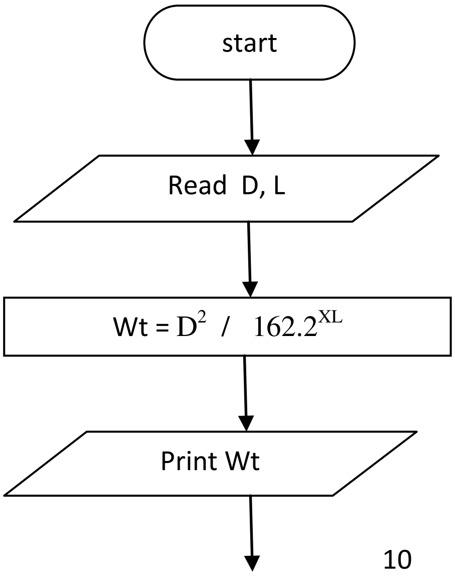
PROBLEM STATEMENT:

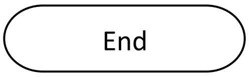
In mathematical theory, the weight of a steel bar calculated using formula

$$W= \frac{D2/}{162.2} *XL$$

Read the diameter (D) and length(L) then applying the above formula to calculate the weight and print it.

FLOWCHART





RESULT:

Thus the weight of a steel bar calculation in real time were identified and solved.

Ex. No : 2(a) **Date :**	**EXCHANGE THE VALUES OF TWO VARIABLES**

AIM:

Towriteapython programfor exchanging the values of two variables by using simple statement.

ALGORITHM:

```
Step 1: start the process.
Step 2: get the input values as a, b.
Step 3: swapping the values as a,b=b,a
Step 4: print the swapped values
Step 5: stop the process
```

PROGRAM

```
#Exchange the values of two variables
a=int(input('Enter the first value'))
b=int(input('Enter the second value'))
print('Before Swapping',a,b)
a,b=b,a
print('After Swapping',a,b)
```

OUTPUT

```
Enter the first value 10
Enter the second value 45
```

```
Before Swapping 10 45

After Swapping 45 10
```

RESULT:

Thus the program for exchanging the values of two variables has been implemented and the output verified successfully.

Ex. No : 2(b) **Date :**	**CIRCULATE THE VALUES OF N VARIABLES**

AIM:

To write a python program for circulate the values of N variables.

ALGORITHM:

Step 1: start the process.

Step 2: define notation list and n.

Step 3: assign new for storing circulate value of the list.

Step 4: give the example list and print it.

Step 5: rotate the example list by clockwise as 1,2,-2 and print it.

Step 5: stop the process.

PROGRAM

```
#Circulate the values of n variables
def rotate(list,n):
new=list[n:]+list[:n]
return new
example=[1,2,3,4,5]
print('Original list:',example)
a=rotate(example,1)
print('List rotated clockwise by 1:',a)
a=rotate(example,2)
print('List rotated clockwise by 1:',a)
a=rotate(example,-2)
print('List rotated clockwise by 1:',a)
```

OUTPUT

```
Original list: [1, 2, 3, 4, 5]

List rotated clockwise by 1: [2, 3, 4, 5, 1]
List rotated clockwise by 1: [3, 4, 5, 1, 2]

List rotated clockwise by 1: [4, 5, 1, 2, 3]
```

RESULT:

Thus the program for Circulate the values of n variables has been implemented and the output verified successfully.

Ex. No : 2(c) **Date :**	**DISTANCE BETWEEN TWO POINTS**

AIM:

To write a python program for distance between two points by using simple statement.

ALGORITHM:

Step 1: start the process.

Step 2: import the math function.

Step 3: assign p1 and p2 value.

Step 4: using math.sqrt function to identify the distance between p1 and p2.

Step 5: print the distance.

Step 5: stop the process.

PROGRAM

```
#Distance between two points

import math

p1=[2,4]

p2=[3,6]

distance=math. sqrt(((p2[0]-p1[0])**2)+((p2[1]-p1[1])**2))

print('Distance between two points:',distance)
```

OUTPUT

```
Distance between two points: 2.23606797749979
```

RESULT:

Thus the program for distance between two points has been implemented and the output verified successfully.

Ex. No : 3(a) **Date :**	**NUMBER SERIES**

AIM:

To write a python program for number series by using iterative loops.

ALGORITHM:

```
Step 1: start the process.
Step 2:get the input as n(number if u want print the series.
Step 3: initialize sum=0
Step 4: using for loop until the given number.
Step 5: add sum value + i.
Step 6:print the sum value
Step 7: stop the process.
```

PROGRAM

```
#Number Series
n=int(input('Enter the number'))
sum=0
fori in range(1,n+1):
sum=sum+i
print(i,'+',sep=",end=")
print('=',sum)
```

OUTPUT

```
Enter the number 6
```

```
1+2+3+4+5+6+= 21
```

RESULT:

Thus the program for number series has been implemented and the output verified successfully.

Ex. No : 3(b) **Date :**	**NUMBER PATTERN**

AIM:

To write a python program for number pattern by using iterative loops.

ALGORITHM:

Step 1: start the process.

Step 2:get the input as n(number of rows if u want print the pattern).

Step 3: initialize num=1

Step 4: using for loop to check range of the i.

Step 5: again using for loop to check range of the j.

Step 6: print the number pattern (num=num+1)

Step 7: stop the process.

PROGRAM

```
#Number Pattern
n=int(input('Enter the number of rows'))
num=1
fori in range(1,n+1):
for j in range(1,i+1):
print(num,end=' ')
num=num+1
print()
```

OUTPUT

```
Enter the number of rows 5

1

2 3
```

```
4 5 6
7 8 9 10
11 12 13 14 15
```

RESULT:

Thus the program for number pattern has been implemented and the output verified successfully.

Ex. No : 3(c) Date :	PYRAMID PATTERN

AIM:

To write a python program for pyramid pattern by using iterative loops.

ALGORITHM:

Step 1: start the process.

Step 2:get the input as rows=number of rows .

Step 3: initialize k=0

Step 4: using for loop to check range of the i.

Step 5: again using for loop to check range of the space.

Step 6:using while loop to identify the factorial value of k.

Step 7:print the pyramid pattern of the given value.

Step 7: stop the process.

PROGRAM

```
#PYRAMID PATTERN
rows = int(input("Enter number of rows: "))
k = 0
fori in range(1, rows+1):
for space in range(1, (rows-i)+1):
print(end="  ")
while k!=(2*i-1):
print("* ", end="")
```

```
        k += 1
    k = 0
print()
```

OUTPUT

```
Enter number of rows: 5
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

RESULT:

Thus the program for pyramid pattern has been implemented and the output verified successfully.

Ex. No : 4(a) **Date :**	**ITEMS PRESENT IN A LIBRARY**

AIM:

To write a python program for items present in a library using tuple operations.

ALGORITHM:

Step 1: start the process

Step 2: initialize library details as tuple values.

Step 3: perform tuple operations like index, length, concatenation, repetition, membership.

Step 4: slice the tuple value.

Step 5: print all the tuple operation results.

Step 6: stop the process.

PROGRAM

```
t_library1= (“books”.”guides”,”journals”)

t_library2= (“questionpapers”.”megazine”,”ieeepapers”)

print (“index position of library1:”,t_library1[0])

print (“number of items in library1:”,len(t_library1))

print (“number of items in library2:”,len(t_library2))

print (“concatenation of items in library1and library2:”,t_library1+t_library2)

print (“repetition of items in library1:”,t_library1*2)

print (“membership operation of library1:”,”megazine” in t_library1)

print (“membership operation of library2:”,”megazine” in t_library2)

print (“slicing of library1:”,t_library1[0:2])
```

OUTPUT

```
index position of library1:books
number of items in library1:3
number of items in library2:3
concatenation of items in library1and
library2:(“books”.”guides”,”journals”,“questionpapers”.”megazine”,”ieeepapers”)
repetition of items in library1:(“books”.”guides”,”journals”,“books”.”guides”,”journals”)
membership operation of library1: False
membership operation of library2: True
slicing of library1:(“books”,”guides”)
```

RESULT:

Thus the program for items present in a library using tuple operations has been implemented and the output verified successfully.

Ex. No : 4(b) **Date :**	**CONSTRUCTION OF A BUILDING**

AIM:

To write a python program for materials required for construction of a building using tuple operations.

ALGORITHM:

Step 1: start the process

Step 2: initialize materials required for construction of a building as tuple values.

Step 3: perform tuple operations like index, length, concatenation, repetition, membership.

Step 4: slice the tuple value.

Step 5: print all the tuple operation results.

Step 6: stop the process.

PROGRAM

```
t_building1= (“bricks”.”sand”,”cement”)

t_building 2= (“tiles”.”paint”,”wood”)

print (“index position of building1:”,t_building1[1])

print (“number of items in building1:”,len(t_building1))

print (“number of items in building2:”,len(t_building2))

print (“concatenation of items in building1and building2:”,t_building1+t_building2)

print (“repetition of items in building1:”,t_building2*2)

print (“membership operation of building1:”,”wood” in t_building1)

print (“membership operation of building2:”,”wood” in t_building2)

print (“slicing of building1:”,t_building1[0:2])
```

OUTPUT

```
index position of  building1:sand
number of items in building1:3
number of items in building2:3
concatenation of items in building1and
building2:(“bricks”.”sand”,”cement”,“tiles”.”paint”,”wood”)
repetition of items in building2:(“tiles”.”paint”,”wood”,“tiles”.”paint”,”wood”)
membership operation of building1: False
membership operation of  building2: True
slicing of building1:(“bricks”,”sand”)
```

RESULT:

Thus the program for materials required for construction of a building using tuple operations has been implemented and the output verified successfully.

Ex. No : 4(b) **Date :**	**COMPONENTS OF CAR**

AIM:

To write a python program for components of a car using list operations.

ALGORITHM:

Step 1: start the process

Step 2: initialize components of a caras list values.

Step 3: perform list operations like length, concatenation, repetition, membership.

Step 4: slice the list value.

Step 5: print all the list operation results.

Step 6: stop the process.

PROGRAM

```
l_car1= (“tyre”.”mirror”,”seat”,engine)
l_car 2= (“steering”.”wheel”,”light”)
print (“index position of car1:”,l_car1[2])
print (“number of items in car1:”,len(l_car1))
print (“number of items in car2:”,len(l_car2))
print (“concatenation of items in car1and car2:”,l_car1+l_car2)
print (“repetition of items in car1:”,l_car1*2)
print (“membership operation of car1:”,”steering” in l_car1)
print (“membership operation of car2:”,”steering” in l_car2)
print (“slicing of car1:”,l_car1[0:2])
```

OUTPUT

```
index position of  car1:seat
```

```
number of items in car1:4

number of items in car2:3

concatenation of items in car1and
car2:[“tyre”.”mirror”,”seat”,“engine”.”steering”,”wheel”,”light”]

repetition of items in car1:[“tyre”.”mirror”,”seat”,“engine”,“tyre”.”mirror”,”seat”,“engine”]

membership operation of  car1: False

membership operation of  car2: True

slicing of car1:(“tyre”,”mirror”)
```

<u>RESULT:</u>

Thus the program for components of a car using list operations has been implemented and the output verified successfully.

Ex. No : 5(a) **Date :**	**LANGUAGE DETAILS**

AIM:

To write a python program for programming language details using set operations.

ALGORITHM:

```
Step 1: start the process
Step 2: initialize programming language details of set values.
Step 3: perform set operations like union, intersection and set difference.
Step 4: print all the set operation results.
Step 5: stop the process.
```

PROGRAM

```
set1_lang={“c”,”c++”,”java”,”.Net”}
set2_lang={“python”,”java”,”c#”,”.Net”}
print(“the set1 languages are”,set1_lang)
print(“the set2 languages are”,set2_lang)
print(“union of two sets are”,set1_lang.union(set2_lang))
print(“intersection of two sets are”,set1_lang.intersection(set2_lang))
print(“difference of two sets are”,set1_lang.difference(set2_lang))
```

OUTPUT

```
the set1 languages are{‘c’,’c++’,’java’,’.Net’}
the set2 languages are{‘python’,’java’,’c#’,’.Net’}
union of two sets are{‘c’,’c++’,’java’,’.Net’,’python’,’c#’}
intersection of  two sets are {’java’,’.Net’}
difference of  two sets are{‘c’,’c++’}
```

RESULT:

Thus the program for programming languages using set operations has been implemented and the output verified successfully.

Ex. No : 5(b) **Date :**	**COMPONENTS OF AN AUTOMOBILE**

AIM:

To write a python program for implementing components of an automobile using dictionaries.

ALGORITHM:

```
Step 1: start the process
Step 2: initialize components of  an automobile using Dictionaries.
Step 3: perform dictionaries operations like add items, accessing the items and membership
operator .
Step 4: print all the dictionary operation results.
Step 5: stop the process.
```

PROGRAM

```
automobile={“transmission”:”clutch”,”body”:”steeringsystem”,”auxiliary”:”seats”}
print(“items in the dictionaries:”,automobile)
automobile[“body2”]=”wheels”
print(“add element in the dictionaries:”,automobile)
print(“accessing single element in the dictionaries:”,automobile[“auxiliary”])
print(“item in the dictionaries or not:”transmission” in automobile)
print(“item in the dictionaries or not:”component” in automobile)
```

OUTPUT

```
items in the dictionaries:{‘transmission’:’clutch’,’body’:’steeringsystem’,’auxiliary’:’seats’}
add element in the dictionaries:
{‘transmission’:’clutch’,’body’:’steeringsystem’,’auxiliary’:’seats’,’body2’:’wheels’}
accessing single element in the dictionaries: seats
item in the dictionaries or not: True
item in the dictionaries or not: False
```

RESULT:

Thus the program for components of an automobile using dictionary operations has been implemented and the output verified successfully.

Ex. No : 6(a) Date :	FACTORIAL

AIM:

To write a python program for factorial using user defined functions.

ALGORITHM:

Step 1: start the process

Step 2: initialize num variable to get the input.

Step 3: user defined function is defined for factorial of the given value .

Step 4: check the values using conditionals.

Step 5: calculate the factorial by using while loop.

Step 6: print the factorial result.

Step 5: stop the process.

PROGRAM

```
def factorial(n):
        if n<0:
                return 0
        elif n==0 or n==1:
                return 1
        else:
                fact = 1
        while(n>1):
                fact*=n
                n-=1
        return fact
num=int(input(“enter the number:”))
print(“Factorial of”,num,”is”factorial(num))
```

OUTPUT

```
enter the number: 5
factorial of 5 is 120
```

RESULT:

Thus the program for factorial of given number using user defined functions has been implemented and the output verified successfully.

Ex. No : 6(a) **Date :**	**LARGEST NUMBER IN THE LIST**

AIM:

To write a python program for largest number in the list using user defined functions.

ALGORITHM:

```
Step 1: start the process.
Step 2: initialize list values.
Step 3: user defined function is defined for largest number of the given list.
Step 4: define the root element as max.
Step 5: check the each element is large compared to max of the array using for loop.
Step 6: find the largest element in the list using if conditions.
Step 7: return the largest element.
Step 8: stop the process.
```

PROGRAM

```
def large(arr)

max=arr[0]

for element in arr:

    if(element>max)

        max=element

    return max

list=[3,8,7,4,9]

result=large(list)

print("the largest number in the given list:"result)
```

OUTPUT

```
the largest number in the given list: 9
```

RESULT:

Thus the program for largest number of the given list using user defined functions has been implemented and the output verified successfully.

Ex. No : 6(b)

Date :

AREA OF RECTANGLE,SQUARE,TRIANGLE,CIRCLE, PARALLELOGRAM

AIM:

To write a python program for find the area of different shapes.

ALGORITHM:

```
Step 1: start the process.
Step 2: get the shape from the user to calculate area.
Step 3: call the function for the user defined shape.
Step 4: calculate the area of different shapes like rectangle, square, triangle, circle, parallelogram.
Step 5: print the area of shape for user specified.
Step 6: stop the process.
```

PROGRAM

```
defcalculate_area(name):
name=name.lower()
if(name==”rectangle”)
l=int(input(“enter the rectangle’s length:”))
b=int(input(“enter the rectangle’s breadth:”))
rect_area=l*b
print(“the area of rectangle is:”,rect_area)
elif(name==”square”)
s=int(input(“enter square’s side length:”))
square_area=s*s
print(“the area of square is:”,square_area)
elif(name==”triangle”)
h=int(input(“enter triangle’s height:”))
b=int(input(“enter triangle’s breadth:”))
tri_angle=0.5*b*h
```

```
print(“the area of triangle is:”,tri_angle)
elif(name==”circle”)
s=int(input(“enter circle’s radius length:”))
pi=3.14
circle_area=pi*r*r
print(“the area of circle is:”,circle_area)
elif(name==”parallelogram”)
h=int(input(“enter parallelogram’s height:”))
b=int(input(“enter parallelogram’s breadth:”))
para_area=b*h
print(“the area of parallelogram is:”,para_area)
else
print(“sorry! This shape is not available”)
if_name_==”_main_”:
print(“calculate shape area”)
shape_name=input(“enter the shape name whose area you want to find:”)
calculate_area(shape_name)
```

OUTPUT

```
calculate shape area
enter the shape name whose area you want to find : rectangle
enter the rectangle’s length:4
enter the rectangle’s breadth:5
the area of rectangle is:20
>>>
calculate shape area
enter the shape name whose area you want to find : square
enter square’s side length:6
the area of square is:36
```

```
>>>
calculate shape area
enter the shape name whose area you want to find : triangle
enter triangle’s height:7
enter triangle’s breadth:5
the area of triangle is:17.5
>>>
calculate shape area
enter the shape name whose area you want to find : circle
enter circle’s radius length:4
the area of circle is:50.24
>>>
calculate shape area
enter the shape name whose area you want to find : parallelogram
enter parallelogram’s height:4
enter parallelogram’s breadth:3
the area of parallelogram is:12
>>>
calculate shape area
enter the shape name whose area you want to find : pallelogram
sorry! This shape is not available
```

RESULT:

Thus the program for area of different shapes using user defined functions has been implemented and the output verified successfully.

Ex. No : 7 **Date :**	**STRING METHODS(REVERSE,PALINDROME,CHARACTER COUNT,REPLACING CHARACTER)**

AIM:

To write a python program for string methods forreverse,palindrome,character count,replacing character and check whether the given string is palindrome or not.

ALGORITHM:

```
Step 1: start the process.
Step 2: get the string from the user to manipulate string methods.
Step 3: use the string method[::-1],for displaying reverse of the string.
Step 4: check whether the string is palindrome or not using palindrome function.
Step 5: use len(str),for finding character count
Step 6: use replace(),for finding replacing character
Step 7: stop the process.
```

PROGRAM

```
defisPalindrome(s):
return s=s[::-1]
str=input(“enter the string:”)
print(“the reverse of the string is:”,str[::-1])
check_palin=isPalindrome(str)
ifcheck_palin:
print(“the given string(“,str,”) is a palindrome”)
else:
print(“the given string(“,str,”) is not a palindrome”)
print(“the length of the given string is:”,len(str))
print(“replace a character in the string:”,str.replace(“r”,”l”,1))
```

OUTPUT

```
enter the string: refer
the reverse of the string is:refer
the given string(refer)is a palindrome
the length of the given string is:5
replace a character in the string:lefer
>>>
enter the string: read
the reverse of the string is: daer
the given string(refer)is not a palindrome
the length of the given string is:4
replace a character in the string: lead
```

RESULT:

Thus the program for string methods forreverse,palindrome,character count,replacing character and check whether the given string is palindrome or nothas been implemented and the output verified successfully.

Ex. No : 8 **Date :**	**Pandas, NumPy, Matplotlib&SciPy**

AIM:

To implement a python standard library functions using written modules like pandas,numpy,matplotlib and scipy.

Pandas:

Pandas is a python library used for working with datasets. It has functions for analyzing,cleaning, exploring and manipulating data.

Pandas allows us to analyze big data and make conclusions based on statistical theories.

Pandas has two important data structures: *series *DataFrame

Series - one-dimensional labelled array like object.

DataFrame – contains an ordered collection of columns.

Installation of pandas: python and PIP installed on a system, then installation of pandas is very easy.

```
C:\users\your name>pip install pandas
```

Once pandas is installed, import it in your applications by adding the “import” keyword:

```
import pandas
```

PROGRAM

```
import pandas
mydataset={‘cars’:[“BMW”,”Volvo”,”ford”],’passings’:[3,7,2]}
myvar=pandas.DataFrame(mydataset)
print(myvar)
```

OUTPUT

```
        cars        passings
```

```
0       BMW       3
1       Volvo     7
2       ford      2
```

NumPy:

NumPy means Numeric python or Numerical python. It is a fundamental package for scientific computing in python. It is a python library that provides a multidimensional array object.

PROGRAM

```
importnumpy as np
arr=np.array([[-1,2,0,4],[4,-0.5,6,0],[2.6,0,7,8],[3,-7,4,2.0]])
print(“initial array:”)
print(arr)
sliced_arr=arr[:2,::2]
print(“array with first two rows andalternate columns(0 and 2):\n”,sliced_arr)
index_arr=arr[[1,1,0,3],[3,2,1,0]]
print(“\n elements at indices(1,3),(1,2),(0,1),(3,0):\n”,index_arr)
```

OUTPUT

```
initial array:
[-1,2,0,4]
[4,-0.5,6,0]
[2.6,0,7,8]
[3,-7,4,2.0]
array with first two rows and alternate columns(0 and 2):
[[-1,0][4,6]]
elements at indices(1,3),(1,2),(0,1),(3,0):
[0.60,2.3]
```

Matplotlib:

Matplotlib is a python 2D plotting library which produces quality pictures. It is an amazing visualization library in python for 2D plots of arrays. Matplotlib is multi-platform data visualization library built on Numpy arrays and designed to work with the broader SciPy stack.

Importing matplotlib:

```
frommatplotlib import pyplot as plt
```

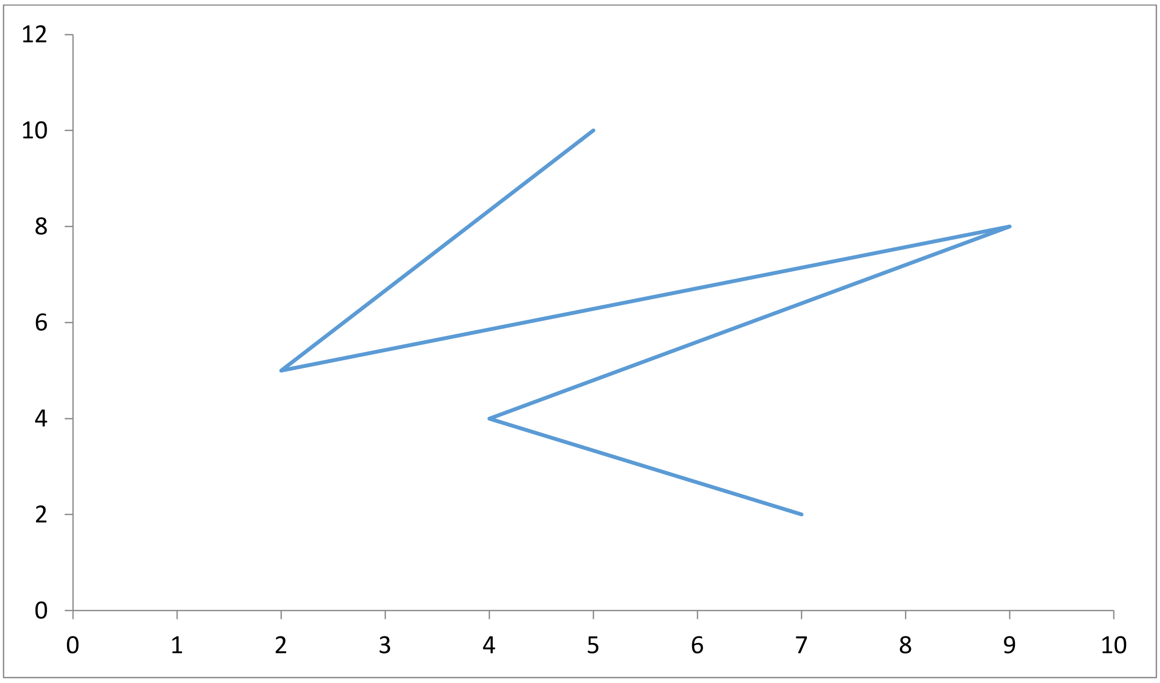
or

```
importmatplotlib.pyplot as plt
```

PROGRAM for Line Plot

```
frommatplotlib import pyplot as plt
x=[5,2,9,4,7]
y=[10,5,8,4,2]
plt.plot(x,y)
plt.show()
```

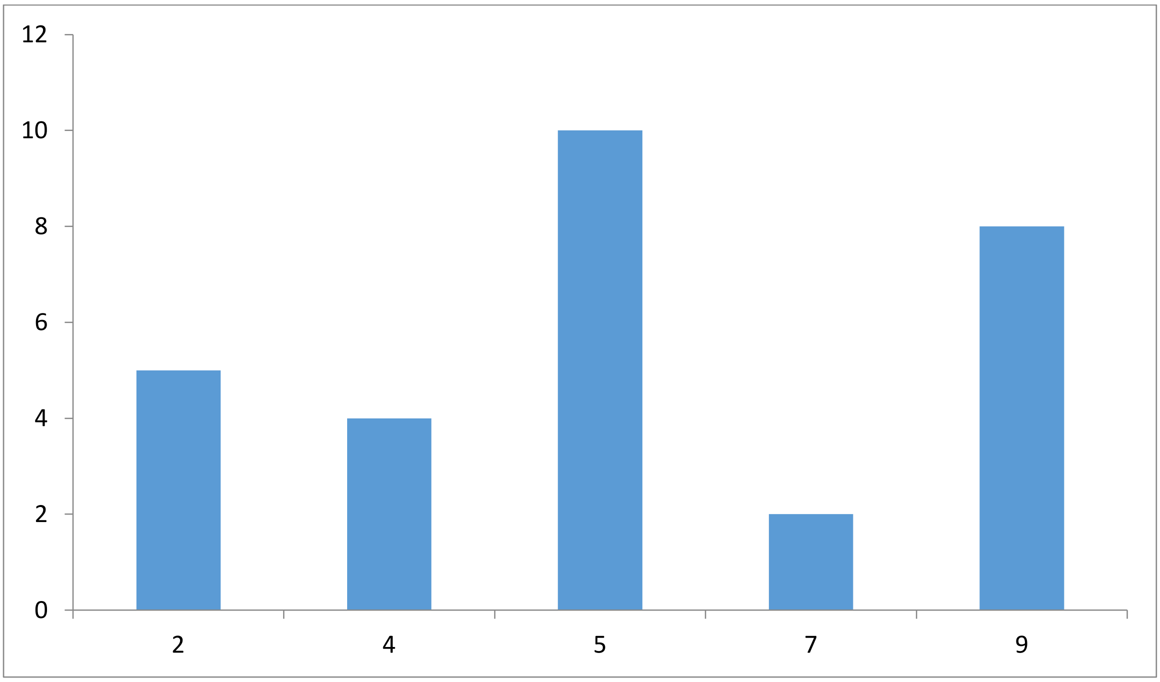
OUTPUT



PROGRAM for Bar Plot

```
frommatplotlib import pyplot as plt
x=[5,2,9,4,7]
y=[10,5,8,4,2]
plt.bar(x,y)
plt.show()
```

OUTPUT



SciPy:

SciPy stands for scientific python. SciPy is a free and open-source python library used for scientific computing and technical computing. SciPy builds on NumPy so import SciPy ,no need to import NumPy.

Import SciPy

```
        fromscipy import module
```

```
fromscipy import constants
```

we have imported the constants module from SciPy and the application ready to use it.

PROGRAM

```
#PI is an example of scientific constant:
fromscipy import constants
print(constants.pi)
```

OUTPUT

```
3.141592653589793
```

RESULT:

Thus the program for python standard library functions using written modules like pandas,numpy,matplotlib and scipy has implemented and output verified successfully.

Ex. No : 9 **Date :**	**FILE HANDLING**

AIM:

To write a python program for file handling operations for word count, longest word and copy of the content from one file to another file.

ALGORITHM:

```
Step 1: start the process.
Step 2: create a text file as text1.txt
Step 3: open the file f1 (“text1.txt”) in the read mode.
Step 4: open the file f2 (“text2.txt”) in the write mode.
Step 5: read the content from the file1 and write into file2.
Step 6: close the files f1 & f2.
Step 7: open the file f3(“text1.txt”) in the read mode.
Step 8: read the contents from the file3 using the method read().split() and stored it in variables.
Step 9: print the word count.
Step 10: print the longest word in the file using conditionals.
Step 11: stop the process.
```

Create a file

text1.txt

```
hello
this is my python programming
```

PROGRAM:

```
f1=open(“text1.txt”,”r”)
```

```
f2=open(“text2.txt”,”w”)
str=f1.read()
f2.write(str)
print(“content of file1(f1) is read and write to file2(f2)”)
f1.close()
f2.close()
f3=open(“text1.txt”,”r”)
words=f3.read().split()
print(“the word count of file(f3) is:”,len(words))
max_len=len(max(words,key=len))
for word in words:
        iflen(word)==max_len:
                print(“the longest word in the file(f3) is:”,word)
f3.close()
```

OUTPUT

```
content of file1(f1) is read and write to file2(f2)

the word count of file(f3) is: 6

the longest word in the file(f3) is: programming
```

RESULT:

Thus the program for file handling operations for word count, longest word and copy of the content from one file to another file has implemented and output verified successfully.

Ex. No : 10(a) **Date :**	**EXCEPTION HANDLING**

AIM:

To write a python program for implement divide by zero error using exception handling.

ALGORITHM:

Step 1: start the process.

Step 2: get the dividend, divisor values from the user.

Step 3: calculate the quotient value using try block.

Step 4: check the divide by zero error in catch block.

Step 5: print the exception handling output.

Step 6: stop the process.

PROGRAM:

```
n=int(input("enter the values of dividend(n):"))
d=int(input("enter the values of divisor(d):"))
c=int(input("enter the values of divisor(c):"))
try:
        q=n/(d-c)
        print("quotient:",q)
        exceptZeroDivisionError:
        print("division by zero")
```

OUTPUT

```
enter the values of dividend(n): 8
enter the values of divisor(d):4
enter the values of divisor(c):4
division by zero
```

RESULT:

Thus the program for implement divide by zero error using exception handling has implemented and output verified successfully.

Ex. No : 10(b) **Date :**	**CHECK THE VOTER'S AGE VALIDITY**

AIM:

To write a python program to check the voter's age validity using exception handling.

ALGORITHM:

Step 1: start the process.

Step 2: get the voter's name and voter's age from the user.

Step 3: calculate the voter's age eligibility in try block.

Step 4: check the value error in catch block.

Step 5: print the eligibility status of voter's age using exception handling.

Step 6: stop the process.

PROGRAM:

```
name=input("enter the voter name:")
try:
    age=int(input("enter your age:"))
    if age>18:
        print("eligible to vote")
    else:
        print("not eligible to vote")
exceptValueError as err:
print(err)
else:
```

```
print(“thank you, you have successfully checked the voting eligibility”)
```

OUTPUT

```
enter the voter name: rajesh

enter your age:21

eligible to vote

thank you, you have successfully checked the voting eligibility

>>>

enter the voter name:mukesh

enter your age: twentty

invalid literal for int() with base 10:’twentty’
```

RESULT:

Thus the program tocheck the voter’s age validity using exception handling has implemented and output verified successfully.

Ex. No : 11 **Date :**	**EXPLORING PYGAME TOOL**

AIM:

To write a python program for exploring pygame tool.

ALGORITHM:

Step 1: start the process

Step 2: import pygame module and initialize pygame.

Step 3: set the screen dimensions.

Step 4: implement pygame tools like pygame .display.setmode(),pygame.event.get(), pygame.QUIT,pygame.display.flip().

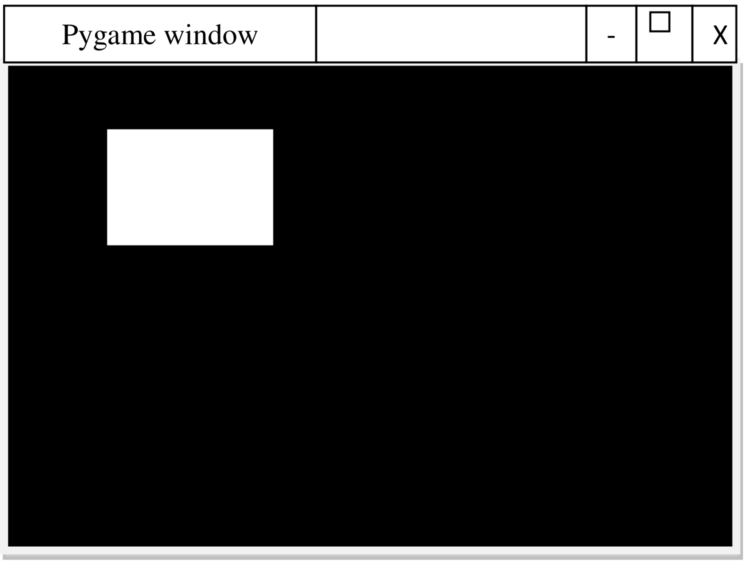
step 5: stop the process.

PROGRAM:

```
importpygame
pygame.init()
screen=pygame.display.set_mode((400,300))
done= False
while not done:
    for event in pygame.event.get():
        ifevent.type==pygame.QUIT:
            done= True
    pygame.draw.rect(screen,(0,125,255),pygame.Rect(30,30,60,60))
```

```
        pygame.display.flip()
```

OUTPUT



RESULT:

Thus the program for exploring pygame tool has implemented and output verified successfully.

Ex. No : 12 **Date :**	**GAME ACTIVITY USING PYGAME**

AIM:

To write a python program for bouncing ball game activity using pygame tool.

ALGORITHM:

Step 1: start the process.

Step 2: download the ball image and save it.

Step 3: import necessary pygame module.

Step 4: initialize the values like speed, color, width and height.

Step 5: read and load the image using pygame procedure.

Step 6: give the bounce speed of the ball using while loop.

Step 7: display the bouncing ball output.

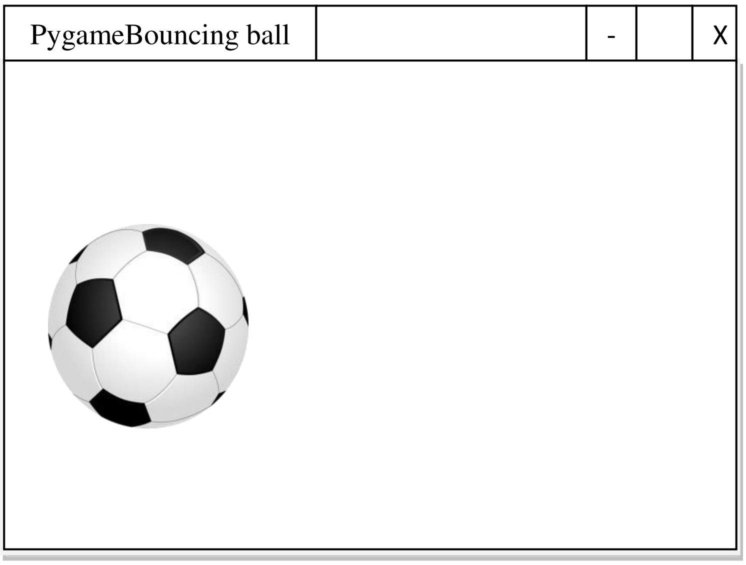
Step 8: stop the process.

PROGRAM:

```
importos
os.environ[“PYGAME_HIDE_SUPPORT_PROMPT”]=”hide”
importsys.pygame
frompygame.locals import*
pygame.init()
speed=[1,1]
color=(255,250,250)
width=550
```

```
height=300
screen=pygame.display.set_mode((width,height))
pygame.display.set_caption(“pygame bouncing ball”)
ball=pygame.image.load(“ball.png”)
rect_boundry=ball.get_rect()
while 1:
    for event in pygame.event.get():
    rect_boundry=rect_boundry.move(speed)
    ifrect_boundry.left<0 or rect_boundry.right>width:
        speed[0]=-speed[0]
ifrect_boundry.top<0 or rect_boundry.bottom>height:
        speed[1]=-speed[1]
    screen.fill(color)
    screen.blit(ball,rect_boundry)
    pygame.display.flip()
    ifevent.type==QUIT:
        pygame.quit()
        sys.exit()
```

OUTPUT



RESULT:

Thus the program for bouncing ball game activity using pygame tool has implemented and output verified successfully.