1.ADDITION OF TWO NUMBERS

ALGORITHM:

1.Start
2.Read a and b values
3.Add a and b values and assign it to c, c=a+b
4.Print the result
5.Stop

PROGRAMS:

```
a=int(input("enter first no"))
b=int(input("enter second no"))
c=a+b
print("the sum is",c)
```

OUTPUT:

```
enter first no20
enter second no10
the sum is 30
```

2.AREA OF CIRCLE

ALGORITHM:

1.Start
2.Read radius value
3.Calculate area of circle, area=3.14*radius*radius
4.Print the result
5.Stop

PROGRAM:

```
radius=int(input("enter the radius"))
area=3.14*radius*radius
print("Area of the Circle is ",area)
```

OUTPUT:

```
enter the radius 5
Area of the Circle is 78.5
```

3.SIMPLE INTEREST

ALGORITHM:

1.Start
2.Read p, n and r values
3.Calculate simple interest, si=p*n*r/100
4.Print the result
5.Stop

PROGRAM:

```
p=int(input("enter principle amount"))
n=int(input("enter no of years"))
r=int(input("enter rate of interest"))
si=p*n*r/100
print("simple interest is ",si)
```

OUTPUT:

```
enter principle amount25000
enter no of years12
enter rate of interest5
simple interest is 15000.0
```

4.CELSIUS TO FAHRENHEIT

ALGORITHM:

1.Start
2.Read celsius value
3.Calculate fahrenheit, f=(c*1.8)+32
4.Print the result
5.Stop

PROGRAM:

```
c=int(input("Enter Celsius"))
f=(c*1.8)+32
print("Temperature in Fahrenheit",f)
```

OUTPUT:

```
Enter Celsius 36
Temperature in Fahrenheit 96.8
```

5.SWAP TWO VALUES OF VARIABLES

ALGORITHM:

1.Start
2.Read two values
3.Assign temp=a, then a=b and b=temp
4.Print the result
5.Stop

PROGRAM:

```
a=int(input("enter a value"))
b=int(input("enter b value"))
temp=a
a=b
b=temp
print("a=",a,"b=",b)
```

OUTPUT:

```
enter a value89
enter b value78
a= 78 b= 89
```

6.ODD OR EVEN

ALGORITHM:

1.Start
2.Read a value
3.Check the condition if (number%2==0), print even
4.Else,print odd
5.Stop

PROGRAM:

```
number=int(input("Enter a number:"))
if(number%2==0):
      print("IT IS AN EVEN NUMBER")
else:
      print("IT IS AN ODD NUMBER")
```

OUTPUT:

```
Enter a number:24
IT IS AN EVEN NUMBER
```

7.SUM OF N NUMBERS

ALGORITHM:

```
1.Start
2.Read a value and initialize i=1 and sum=0
3.Check while condition 'i' less than or equal to 'n'
3.a)add sum with 'i' and assign it to sum
3.b)increment 'i' value
4.Print the result
5.Stop
```

PROGRAM:

```
n=int(input("Enter a number:"))
sum=0
i=1
while (i<=n):
    sum=sum+i
    i=i+1
print("sum is",sum)
```

OUTPUT:

```
Enter a number:10
sum is 55
```

8.SUM OF N EVEN NUMBERS

ALGORITHM:

```
1.Start
2.Read a value and initialize sum=0
3.Perform for loop in range (2,n+1,2)
3.a)add sum with 'i' and assign it to sum
4.Print the result
5.Stop
```

PROGRAM:

```
n=int(input("Enter a number:"))
sum=0
for i in range(2,n+1,2):
    sum+=i
print("sum is",sum)
```

OUTPUT:

```
Enter a number:10
sum is 30
```

9.GREATEST OF THREE NUMBERS

ALGORITHM:

1.Start
2.Read a, b and c value
3.Check if condition 'a' is greater than 'b' and 'c', print a is the greatest value
4.Else check the condition 'b' is greater than 'c' and 'a', print b is the greatest value
5.Otherwise, print c is the greatest value
6.Stop

PROGRAM:

```
a=int(input("enter the value of a"))
b=int(input("enter the value of b"))
c=int(input("enter the value of c"))
if(a>b and a>c):
    print("the greatest no is",a)
elif(b>c and b>a):
    print("the greatest no is",b)
else:
    print("the greatest no is",c)
```

OUTPUT:

```
enter the value of a10
enter the value of b15
enter the value of c20
the greatest no is 20
```

10.PRINT THE TABLE OF A NUMBER

ALGORITHM:

1.Start
2.Read a value
3.Perform for loop in range(1,11)
4.Print as (n,"x",i,"=",n*i)
5.Stop

PROGRAM:

```
n=int(input("Enter the number to print the table for:"))
for i in range(1,11):
    print(n,"x",i,"=",n*i)
```

OUTPUT:

```
Enter the number to print the table for:10
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
```

10 x 10 = 100

11.FACTORIAL OF A NUMBER

ALGORITHM:

1.Start
2.Read a value
3.The value is passed as an argument to factorial function
4.The base condition is that number to be lesser than or equal to 1 and return 1 if it is
5.Otherwise the function is called recursively with the number minus 1 multiplied by the number itself
6.Stop

PROGRAM:

```
def factorial(n):
    if(n<=1):
        return 1
    else:
        return(n*factorial(n-1))
n=int(input("Enter number"));
print("Factorial:")
print(factorial(n))
```

OUTPUT:

```
Enter number 5
Factorial:
120
```

12.GCD OF TWO NUMBERS

ALGORITHM:

1.Start
2.Get the input values
3.Check whether b is equivalent to zero
4.If b is zero, then print the 1st number
5.else assign a=b & b=remainder of a/b and continue with steps 3&4
6.Print the result
7.Stop

PROGRAM:

```
def gcd(a,b):
     if b==0:
         print(" The GCD of two num",a)
     else:
         return gcd (b,a%b)
a=int(input("Enter the first no:"))
b=int(input("Enter the second no:"))
c=gcd(a,b)
print(c)
```

OUTPUT:

```
Enter the first no:9
Enter the second no:18
The GCD of two num 9
```

13.CHECK WHETHER THE GIVEN NUMBER IS PALINDROME OR NOT

ALGORITHM:

1.Start
2.Get the input value and initialize temp=0 and rev=0
3.Check whether n is greater than zero
3.a)divide 'n' value by 10 using modulo operator, remainder is assigned to 'dig'
3.b)'rev' is multiplied with 10 and add with 'dig' and assigned to 'rev'
3.c)divide 'n' value by 100 using floor division operator, quotient is assigned to 'n'
4.If temp==rev, then print the number is a palindrome
5.Else, print the number is not a palindrome
6.Stop

PROGRAM:

```
n=int(input("Enter number:"))
temp=n
rev=0
while(n>0):
    dig=n%10
    rev=rev*10+dig
    n=n//10
if (temp==rev):
    print("the number is a palindrome!")
else:
    print("the number isn't a palindrome!")
```

OUTPUT:

```
Enter number:98
the number isn't a palindrome!
```

14.SQUARE ROOT USING NEWTON METHOD

ALGORITHM:

1. Define a function named newtonSqrt().
2. Initialize approx as 0.5*n and accurate as 0.5*(approx+n/approx)
3. Use a while loop with a condition accurate!=approx to perform the following,
i) Set approx=accurate
ii) accurate=0.5*(approx+n/approx)
4. Print the value of approx

PROGRAM:

```
def newtonSqrt(n):
    approx = 0.5 * n
    accurate = 0.5 * (approx + n/approx)
    while accurate!= approx:
        approx = accurate
        accurate = 0.5 * (approx + n/approx)
```

```
    return approx
print('The square root is' ,newtonSqrt(81))
```

OUTPUT:

```
The square root is 9.0
```

15.EXPONENTIATION OF A NUMBER

ALGORITHM:

1.Start
2.Get the base 'b' and power 'e' values
3.Initialize p=1
4.for i in range(e): then perform p=p*b
5.Print the result
6.Stop

PROGRAM:

```
b=int(input("Enter base: "))
e=int(input("Enter exponential value: "))
p=1
for i in range(e):
    p=p*b
print("The exponentiation value of number",b,"is",p)
```

OUTPUT:

```
Enter base:2
Enter exponential value:3
The exponentiation value of number 2 is 8
```

16.BASIC OF LIST

AIM:

To write a python program to
1)add new elements to end of the list
2)reverse the elements in list
3)display a list multiple times
4)concatenate two list
5)sort the list

ALGORITHM:

1.Start
2.Using append function, add new elements to end of the list
3.Using reverse function, reverse the elements in list
4.Display the list multiple times by '*' operator
5.Concatenate two list by '+' operator
6.Using sort function, rearrange the list elements in ascending order
7.Stop

PROGRAM:

```
a=[3,2,8]
b=[1,5]
a.append(4)
print("After adding element in list:",a)
a.reverse()
print("Reverse of list:",a)
```

```
c=a*3
print("print list element Multiple times:",c)
d=a+b
print("Concatenation of two list:",d)
a.sort()
print("Sorted list:",a)
```

OUTPUT:

```
After adding element in list: [3, 2, 8, 4]
Reverse of list: [4, 8, 2, 3]
print list element Multiple times: [4, 8, 2, 3, 4, 8, 2, 3, 4, 8, 2,
3]
Concatenation of two list: [4, 8, 2, 3, 1, 5]
Sorted list: [2, 3, 4, 8]
```

17.AVERAGE OF ELEMENTS IN A LIST

ALGORITHM:

```
1.Start
2.Get the number of elements to be stored in the list
3.Use for loop to get the elements
4.Calculate the sum of elements in the list
5.Calculate the average value
6.Print the result
7.Stop
```

PROGRAM:

```
n=int(input("Enter the number of elements to be inserted :"))
a=[]
for i in range(0,n):
    elem=int(input("Enter element:"))
    a.append(elem)
avg=sum(a)/n
print("Average of elements in the list",round(avg,2))
```

OUTPUT:

```
Enter the number of elements to be inserted :3
Enter element:1
Enter element:3
Enter element:5
Average of elements in the list 3.0
```

18.MAXIMUM OF A LIST

ALGORITHM:

```
1.Start
2.Give the list with values
3.Assign list[0] to max
4.for i in range of length of list
4.a) Check the condition if list[i]>max and assign list[i] to max
5.Print the maximum value
6.Stop
```

PROGRAM:

```
list=[1,2,3]
```

```
max=list[0]
for i in range(len(list)):
    if list[i]>max:
        max=list[i]
print(max)
```

OUTPUT:

```
3
```

19.FIRST N PRIME NUMBERS

ALGORITHM:

```
1.Start
2.Get the input value
3.Using for a in range(2,r+1), perform the following
4.Assign k=0
5.Using for i in range(2,a//2+1)
6.Check if(a%i==0), increment k by 1
7.If k is less than or equal to zero, print the result
8.Stop
```

PROGRAM:

```
r=int(input("Enter a number:"))
for a in range(2,r+1):
    k=0
    for i in range(2,a//2+1):
        if(a%i==0):
            k=k+1
    if(k<=0):
        print(a)
```

OUTPUT:

```
Enter a number:10
2
3
5
7
```

20.PRIME NUMBER OR NOT

ALGORITHM:

```
1.Start
2.Get the input value and initialize k=0
3.Using for i in range(2,a//2+1)
3.a)Check if(a%i==0),increment k by 1
4.If k is less than or equal to 0, print it is a prime number
5.Otherwise, print it is not a prime number
6.Stop
```

PROGRAM:

```
a=int(input(Enter a number:"))
k=0
for i in range(2,a//2+1):
    if(a%i==0):
        k=k+1
```

```
if(k<=0):
    print("Prime number")
else:
    print("Not a Prime number")
```

OUTPUT:

```
Enter a number:5
Prime number
```

21.MATRIX ADDITION:

ALGORITHM:

1. Define two matrices X and Y
2. Create a resultant matrix named 'result'
3. for i in range(len(X)):
i) for j in range(len(X[0])):
a) result[i][j] = X[i][j] + Y[i][j]
4. for r in result, print the value of r

PROGRAM:

```
X = [[1,2,3],
[4 ,5,6],
[7 ,8,9]]
Y = [[9,8,7],
[6,5,4],
[3,2,1]]
result = [[0,0,0],
[0,0,0],
[0,0,0]]
for i in range(len(X)):
     for j in range(len(X[0])):
          result[i][j] = X[i][j] + Y[i][j]
for r in result:
     print(r)
```

OUTPUT:

```
[10,10,10],
[10,10,10],
[10,10,10]
```

22.MATRIX MULTIPLICATION:

ALGORITHM:

1. Define two matrices X and Y
2. Create a resultant matrix named 'result'
3. for i in range(len(X)):
i) for j in range(len(Y[0])):
a) for k in range(len(Y)):
b) result[i][j] += X[i][k] * Y[k][j]
4. for r in result, print the value of r

PROGRAM:

```
X = [[12,7,3],
[4 ,5,6],
[7 ,8,9]]
Y = [[5,8,1,2],
```

```
[6,7,3,0],
[4,5,9,1]]
result = [[0,0,0,0],
[0,0,0,0],
[0,0,0,0]]
for i in range(len(X)):
     for j in range(len(Y[0])):
          for k in range(len(Y)):
               result[i][j] += X[i][k] * Y[k][j]
for r in result:
     print(r)
```

OUTPUT:

```
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
```

23.LINEAR SEARCH

ALGORITHM:

1.Create a list
2.Read the element to be searched
3.Check while i<len(items) then perform step4 until conditions satisfied
4.If items[i]is equal to x, then assign flag=1 and break
5.Check if flag is equal to 1, then print the position of element
6.Otherwise print element is not found

PROGRAM:

```
items=[12,24,36,5,60]
print("The list of numbers:",items)
x=int(input("Enter the item to be searched:"))
i=flag=0
while(i<len(items)):
    if(items[i]==x):
        flag=1
        break
    i=i+1
if(flag==1):
    print("item is found at:",i+1)
else:
    print("item is not found")
```

OUTPUT:

```
The list of numbers: [12, 24, 36, 5, 60]
Enter the item to be searched:36
item is found at: 3
```

24.BINARY SEARCH

ALGORITHM:

1. Read the search element
2. Find the middle element in the sorted list
3. Compare the search element with the middle element
i) if both are matching, print element found

ii) else then check if the search element is smaller or larger than the middle element
4. If the search element is smaller than the middle element, then repeat steps 2 and 3 for the
left sublist of the middle element
5. If the search element is larger than the middle element, then repeat steps 2 and 3 for the
right sublist of the middle element
6. Repeat the process until the search element if found in the list
7. If element is not found, loop terminates

PROGRAM:

```
def bsearch(alist,item):
    first=0
    last=len(alist)-1
    found=False
    while first<=last and not found:
        mid=(first+last)//2
        if alist[mid]==item:
            found=True
            print("element found in position",mid)
        else:
            if item<alist[mid]:
                last=mid-1
            else:
                first=mid+mid-1
    return found
a=[6,2,3,5,7,14,25]
x=int(input("enter element to search"))
bsearch(a,x)
```

OUTPUT:

```
enter element to search 5
element found in position 3
```

25.SELECTION SORT:

ALGORITHM:

1. Create a function named sel_sort
2. Initialize pos=i
3. If min>a[j] then perform the following till length of list
4. Set pos=j
5. Swap a[i] and a[pos]
6. Print the sorted list

PROGRAM:

```
def sel_sort(a):
    for i in range(len(a)):
        min=a[i]
        pos=i
        for j in range(i+1,len(a)):
            if min>a[j]:
                min=a[j]
                pos=j
        a[pos],a[i]=a[i],a[pos]
```

```
a=[7,8,9,3,2,1]
sel_sort(a)
print('\n the elements in the list are',a)
```

OUTPUT:

```
the elements in the list are', [1, 2, 3, 7, 8, 9]
```

26.INSERTION SORT

ALGORITHM:

```
1. Create a function named isort
2. Initialize currentvalue=a[index] and position=index
3. while position>0 and a[position-1]>currentvalue, perform the
following till length of list
3.a) a[position]=a[position-1]
3.b) position = position-1
4. a[position]=currentvalue
5. Print the sorted list
```

PROGRAM:

```
def isort(a):
      for index in range(1,len(a)):
            current_value=a[index]
            position=index
      while position>0 and a[position-1]>current_value:
            a[position]=a[position-1]
            position=position-1
      a[position]=current_value
      print('list after insertion:',a)
a=[5,3,4,9,2]
isort(a)
```

OUTPUT:

```
('list after insertion:', [2, 3, 4, 5, 9])
```

27.MERGE SORT

ALGORITHM:

```
1. Create a function named mergesort
2. Find the mid of the list
3. Assign lefthalf = alist[:mid] and righthalf = alist[mid:]
4. Initialise i=j=k=0
5. while i < len(lefthalf) and j < len(righthalf), perform the
following
5.a)if lefthalf[i] < righthalf[j]:assign alist[k]=lefthalf[i] and
Increment i
5.b)else: assign alist[k]=righthalf[j] and Increment j & k
6. while i < len(lefthalf), assign alist[k]=lefthalf[i] and
Increment i & k
7. while j < len(righthalf), assign alist[k]=righthalf[j] and
Increment j & k
8. Print the sorted list
```

PROGRAM:

```
def mergesort(alist):
```

```
    if len(alist)>1:
        mid=len(alist)//2
        lefthalf=alist[:mid]
        righthalf=alist[mid:]
        mergesort(lefthalf)
        mergesort(righthalf)
        i=j=k=0
        while i<len(lefthalf)and j<len(righthalf):
            if lefthalf[i]<righthalf[j]:
                alist[k]=lefthalf[i]
                i=i+1
            else:
                alist[k]=righthalf[j]
                j=j+1
            k=k+1
        while i<len(lefthalf):
            alist[k]=lefthalf[i]
            i=i+1
            k=k+1
        while j<len(righthalf):
            alist[k]=righthalf[j]
            j=j+1
            k=k+1
alist=[54,26,93,17,77,31,44,55,20]
mergesort(alist)
print(alist)
```

OUTPUT:

```
[17, 20, 26, 31, 44, 54, 55, 77, 93]
```

28.COMMAND LINE ARGUMENT(WORD COUNT)

ALGORITHM:

1.Import required package
2.Read the filename
3.Open the file in read mode
4.Read each line from the file to count the words
5.Split each line into words and count them
6.Print the words and its count

PROGRAM:

```
def w_c(rw):
    wc=0
    for line in rw:
        word=line.split()
        wc+=len(word)
    return wc
def rw(filename):
    return open(filename,'r')
def main(filename):
    wc=w_c(rw(filename))
    print (wc)
```

```
if __name__=="__main__":
    import sys
    main(sys.argv[1])
```

OUTPUT:

```
C:\Python\Python37-32>python newcl.py a.txt
10
```

29.BASIC OF STRING

AIM:

To write a python program
1)to calculate length of string
2)reverse the string
3)display the string multiple times
4)concatenate two strings
5)strl="south India" USING string slicing to display "India"

ALGORITHM:

1.Start
2.Get the string values
3.Using length function, the length of the string is calculated
4.Reversing the string can be performed using string slicing
5.Concatenate two strings using '+' operator
6.Slicing can be performed using [start:stop]
7.Stop

PROGRAM:

```
a="South"
b=" India"
print("length of string=",len(a))
print("reverse of string:",a[::-1])
print("multiple times:",a*3)
strl=a+b
print("concatenation:",strl)
print("Slicing:",strl[6:11])
```

OUTPUT:

```
length of string= 5
reverse of string: htuoS
multiple times: SouthSouthSouth
concatenation: South India
Slicing: India
```

30.BASIC OF DICTIONARY

AIM:

To write a Python program

1.To create a dictionary
2.To adding an element to dictionary
3.To display length of the dictionary.
4.To updating element in dictionary.
5.To remove all elements from the dictionary.

ALGORITHM:

1.Create a dictionary 'dic'
2.Add an element to dictionary
3.Display the length of the dictionary using length function
4.Update the element in dictionary using update function
5.Remove all elements from the dictionary using clear function

PROGRAM:

```
dic={'name':'Nithya','age':30}
print("Dictionary:",dic)
dic["occupation"]="Asst.prof"
print("After adding an element:",dic)
length=len(dic)
print("length of dictionary:",length)
dic1={'age':31}
dic.update(dic1)
print("Updated Dictionary:",dic)
dic.clear()
print("After removing all elements from dictionary:",dic)
```

OUTPUT:

```
Dictionary: {'name': 'Nithya', 'age': 30}
After adding an element: {'name': 'Nithya', 'age': 30, 'occupation':
'Asst.prof'}
length of dictionary: 3
Updated Dictionary: {'name': 'Nithya', 'age': 31, 'occupation':
'Asst.prof'}
After removing all elements from dictionary: {}
```

31.ACCEPT 'N' NAMES & SORT IN ALPHABETICAL ORDER

ALGORITHM:

1.Start
2.Get the string value as input
3.Using split(), the words are separated
4.Using sort(), the words are arranged in sorted order
5.Print the result
6.Stop

PROGRAM:

```
my_str="Hello this Is an Example With cased letter"
words=my_str.split()
words.sort()
```

```
print("The sorted words are:")
for word in words:
    print(word)
```

OUTPUT:

```
The sorted words are:
Example
Hello
Is
With
an
cased
letter
this
```

32.FREQUENT WORDS IN A TEXT FILE

ALGORITHM:

1.Read the filename
2.Open the file in read mode
3.Read each line from the file
4.Split each line into words and then count them
5.Print the words and its count

PROGRAM:

```
file=open("C:/Python/Python37-32/a.txt","r+")
wc={}
for word in file.read().split():
    if word not in wc:
        wc[word]=1
    else:
        wc[word]+=1
for k,v in wc.items():
    print (k,v)
```

OUTPUT:

```
hello 2
world 1
hai 1
```

33.ELLIPTICAL ORBITS:

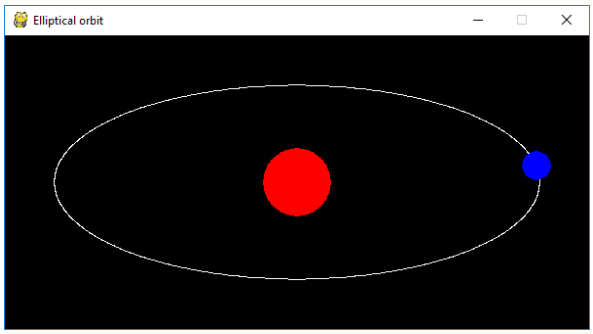
ALGORITHM:

1. Import the required packages
2. Set up the colours for the elliptical orbits
3. Define the parameters to simulate elliptical orbits
4. Display the created orbits

PROGRAM:

```
import pygame
import math
import sys
pygame.init()
screen=pygame.display.set_mode((600,300))
pygame.display.set_caption("Elliptical orbit")
clock= pygame.time.Clock()
while(True):
      for event in pygame.event.get():
            if event.type==pygame.QUIT:
                  sys.exit()
      xRadius=250
      yRadius=100
      for degree in range(0,360,10):
            x1=int(math.cos(degree*2*math.pi/360)*xRadius)+300
            y1=int(math.sin(degree*2*math.pi/360)*yRadius)+150
            screen.fill((0,0,0))
            pygame.draw.circle(screen,(255,0,0),[300,150],35)
            pygame.draw.ellipse(screen,(255,255,255),[50,50,
            500,200],1)
            pygame.draw.circle(screen,(0,0,255),[x1,y1],15)
            pygame.display.flip()
            clock.tick(5)
```

OUTPUT:



34.SIMULATE BOUNCING BALL USING PYGAME

ALGORITHM:

1.Import the required package
2.Define the variable
3.Define some space to display
4.Read the JPEG ball image from corresponding location
5.Bouncing ball in that space is displayed

PROGRAM:

```
import sys
import pygame
pygame.init()
size=width,height=320,240
speed=[2,2]
black=0,0,0
```

```
screen=pygame.display.set_mode(size)
ball= pygame.image.load('C:/Python/Python37-32/ball.jpg')
ballrect=ball.get_rect()
while True:
    for event in pygame.event.get():
         if event.type==pygame.QUIT:sys.exit()
    ballrect=ballrect.move(speed)
    if ballrect.left<0 or ballrect.right>width:
         speed[0]=-speed[0]
    if ballrect.top<0 or ballrect.bottom>height:
         speed[1]=-speed[1]
    screen.fill(black)
    screen.blit(ball,ballrect)
    pygame.display.flip()
```

OUTPUT: