Program 21

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

Program

```
test-str = "hai ashique how are you"
count = 0
for i in test_str:
if i == 'a':
    count = count + 1
print ("count of a string is:" + str (count))
```

Result

The program has been executed and output has been verified.

Output

count of a string is: 3

Program 22

Aim: Adding at the end of a given string. If it already ends with 'ing' then add 'ly'?

program

```
def add_string(str1):
    length = len(str1)
if length > 2:
 if str1[-3:] == 'ing':
    Str1 += 'ly'
 else
    str1+=st 'ing'
 return str1
 print(add_string('as'))
 print(add_string('asb'))
 print(add_string('string'))
```

Result

The program has been executed and output was verified.

Output

as

ashing

stringly

Program 23

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

Program

```
a=[ ]
n=int(input("Enter the number of element in list:"))
for x in range(0,n):
element = input("Enter element"+ str(x+1)+':')
  a.append(element)
max1 = len(a[0])
temp = a[0]
for i in a:
        if (len(i)>max1):
            max1=len(i)
            temp=i
  print("The word with longest length is:")
  print(temp)
```

Result

The program has been executed and output is verified.

Output

Enter the number of elements in list : 3

Enter element 1: Apple

Enter element 2: Banana

Enter element 3: Strawberry

The word with the longest length is : Strawberry

Program 24

Aim: Construct following pattern using nested loop?

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Program

```
n=5
for i in range (n):
    for j in range (i):
        print ('*', end=' ')
    print ('')
for i in range (n, 0, -1):
    for j in range (i):
    print ('*', end = ' ')
    print ('')
```

Result

The program has been executed and output was verified.

Output

Program 25

Aim: Generate all factors of number of a number?

Program

```
def print_factor (x):
    print ('The factor of ', 'x', are:")
    for in range (1, x+1):
    if x% i== 0:
    print (i)
num = 200
print_factors (num)
```

Result

The program has been executed and output was verified.

Output

The factors of 200 are:

1
2
4
5
8
10
20
25
40
50
100
200

Program 26

Aim: Write lambda functions to find area of square, rectangle & triangle

Program

```
Import math
t_peri = lambda p,q,r : p+q+r
r_area = lambda .len, hi : len * br
C_Perl = lambda rad: 2*math.pi*rad
c_area = lambda rad : math.pi * rad* rad
print ("perimeter of triangle(10,20,15) is: ", t_peri (5, 20, 10))
print ("Area of rectangle (30,20) is : ", r_area (20,10))
```

Result

The program has been executed and output was verified.

Output

Perimeter of triangle (10,20,15) is : 35

Area of rectangle (30,20) is : 200

Program 27

Aim: form a list of integers, create a list removing even numbers

Program

```
list = [11, 22, 33, 44, 55, 66]
print = (" original list ")
print (list)
for i in list:
        if (i%2)==0);
        list.remove(i)
        print (" list after removing an even number:")
        print (list)
```

Result

The program has been executed and output was verified.

Output

Original list

[11, 22, 33, 44, 55, 66]

list after removing an even numbers

[11, 33, 44, 55, 66]

list after removing an even number

[11, 33, 55, 66]

list after removing an even number

[11, 33, 55]

Program 28

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

Program

```
import datetime
a = datetime.datetime.now()
a = int(a.year)
b = int(input("Enter final year:"))
print("in leap years")
for i in range(a, b+1):
        if(i%4==0):
            print(i)
```

Result

The program has been executed and output verified.

Output

Enter final year: 2050

leap years:

2024
2028
2032
2036
2040
2044
2048

Program 29

Aim: Generate positive list of numbers from a given list of integers.

Program

```
list1 = [5, -8, 69, 57, -55, 24, -66, -20, 80, -33, -639, 852]
pos = list ()
for i in list1:
        if i > 0:
        pos.append (i)
print ("original list :", list1)
print ("positive integer list :", pos)
```

Result

The program has been executed and output verified.

Output

Original list : [5, -8, 69, 57, -55, 24, -66, -20, 80, -33, -639, 852]

Positive integer list : [5, 69, 57, 24, 80, 852]

Program 30

Aim: find biggest of 3 numbers entered

program

```
a = int(input ('Enter 1st no: '))
b = int (input ('Enter 2nd no: '))
c = int (input ('Enter 3rd no: '))
if a>b and b>c
      print (a, 'is the biggest number')
elif b>a and b>c
      print (b,' is the biggest number')
else:
      print (c,'is the biggest number)
```

Result

The program has been executed and verified.

Output

Enter 1st no : 562

Enter 2nd no: 960

Enter 3rd no : 750

960 is the biggest number

Program 31

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

Program

```
~~clors~~
colors = (input('Enter colors seperated by commas:')).
                                                split (',')
print ('first color:; colors [0])
print ('last color:; colours [len(colors)-1])
```

Result

The program has been executed and output verified.

Output

Enter colors seperated by commas: blue, green,
red, black, yellow

first color : blue
last color : Yellow

Program 32

Aim: Print out all colors from color_list1 not contained in color_list2.

Program

```
colors1=set((input('colour seperated by commas:')).split(','))
colors2=set((input('Colour seperated by commas:')).split(','))
print('colors in color_list1 not contained in colour_list2
      are:', list(colors1.difference(colors2)))
```

Result

The program has been executed and output is verified.

Output

Color seperated by colons : red, orange, green, black

Color seperated by colons : blue, yellow, gold, brown

Color in color_list1 not contained in color_list2 are :

['red', 'orange', 'green', 'black']

Program 33

Aim: Create a single string seperated with spaie from two strings by swapping the character at position 1.

Program

```
Str 1 = Input ('Enter a string :')
Str 2 = input ('Enter another string :')
Str 3 = Str2 [0] + Str 1 [1] + '' + Str 1 [0] + Str 2 [1]
print (Str 3)
```

Result

The program has been executed and output verified.

Output

Enter a string : ashique

Enter another string : kozhikode

kshique aozhikode

Program 34

Aim: Create a package graphics with modulus rectangle, circle and sub package 3D_graphics with modulus cuboid and sphere include method to find area and perimeter of respective figure in each module. Write programs that finds area and perimeter of figure by different importing statements.
(include selective import of modules and import * statements)

Program

→ circle.py

findarea.py

```
import circle
from rectangle import *
from Graphics_3D_graphics import cuboid, sphere
a = float(input ('Enter length of the rectangle:'))
b=float (input ('Enter ~~length of the~~ breadth of the
                                    rectangle : '))
area (a,b)
r = float (input ('Enter the radius of circle:'))
circle.area (r)
l= float (input ('Enter length of the cuboid :'))
b = float (Input ('Enter breadth of the cuboid :'))
h=float (input ('Enter height of the cuboid:'))
cuboid area (l,b,h)
r= float (Input ('Enter the radius of the sphere:'))
sphere.area (r)
```

→ findperimeter.py

```
import circle
from rectangle import *
from graphics_3D_graphics import cuboid, sphere
a = float (input ('Enter length of the rectangle:'))
b = float (input ('Enter breadth of the rectangle:'))
perimeter (a,b)
r = float (input ('Enter the radius of the circle:'))
circle.circumference (r)
l = float (input ('Enter length of the cuboid:'))
b = float (input ('Enter breadth of the cuboid:'))
h = float (input ('Enter height of the cuboid:'))
cuboid.perimeter (l,b,h)
r = float (input ('Enter the radius of the sphere:'))
sphere.perimeter (r)
```

→ rectangle.py

```
def area (a,b):
    print ('Area of rectangle with sides','a','and','b','is:',
                                    %.2f'%(a*b),'Sq.units')
def perimeter (a,b)
    print ('perimeter of rectangle with sides','a:and','b:' is:',
                    '%':2f'%(2*(a+b)),'units')
```

→ 3D_graphics

```
cubad.py
def area (l,b,h):
print ('Total surface area of cuboid with dimensions
        'l','b','h',is ',':%'.2F'%'(2*((l*b)+(b*b)+
        (l*h))),'sq units')
```

```
def perimeter (l,b,h):
    print ('perimeter of cuboid with dimensions ,'l','b','h'
        is ',' %0', %F'%' (4*(l+b+h)), 'unit')
```

→ sphere.py

```
def area (r):
    print ('Area of sphere with radius,'r', 'is:',
            %0 %2F %0 (3.14 * r* r)), 'sq.units')
def perimeter (r):
    print ('perimeter of (great circle of) sphere with
    radius 'r' is:' %0',%F %0 (2 * 3.14*r),' units)
```

Result

The program has been executed and output is verified.

Output

Perimeter of circle with radius 10 is 62.831853071795865

Area of a circle with radius 10 is 314.1592653589793

Area of triangle with length and width 10 is 100

Perimeter of a rectangle with length and width 10 is 40

Area of cuboid with length, width, height 10 is 600

Perimeter of cuboid with length, width, height 10 is 120

Area of sphere with radius 10 is 1256637061435917.3

Perimeter of sphere with radius 10 is 62.83185307179586

Program 35

Aim: Create Rectangle class with attributes length, breadth and methods to find area and perimeter. Compare two rectangle objects by their area.

Program

```
class Rectangle:
    def __init__(self,l,b):
        self.length = l
        self.breadth = b
    def area(self):
        return self.length * self.breadth
    def perimeter(self):
        return 2*(self.length + self.breadth)
    def cmp(self,obj):
        if self.area() > obj.area():
            print('Rectangle with length = ' self.length,
                  and breadth' has the greater area')
        elif self.area() > obj.area():
            print('Rectangle with length = ', obj.length,
                  and breadth = ', object.breadth, 'has
                                  the greater area')
        else:
            print("They have equal area")

r1 = Rectangle(6,5)
r2 = Rectangle(8,4)
r1.cmp(r2)
```

Result

The program has been executed and output is verified.

Output

Rectangle with length = 8 and breadth = 4 has the greater area.

Program 36

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 BankAccount:
    def __init__(self, an, t, b):
        self.accno = a
        self.name = n
        self.type = t
        self.bal = b
    def deposit(self,a):
        self.bal += a
        print("Rs.", 'a', "deposited! current balance is:
                                    Rs", self.bal)
    def withdraw(self, a):
        if self.bal >= a
            self.bal = a
        print (Rs, 'a', "withdrawn; current
            balance is: Rs", self.bal)
    else:
        print ("Insufficient balance to make this
                                    transaction!")
a = int (input ("Enter Account number: "))
```

```
n = input ("Enter name of the account holder :")
t = input ("Enter Account type : ")
b = float (Input ("Enter your balance:"))
acl = Bank Account (a,n,t,b)
acl. deposit (float (Input ("Enter amount to deposit :")))
acl. withdraw (float (Input ("Enter amount to withdraw")))
```

Result

The program has been executed and output is verified.

Output

Enter account number : 11011012135
Enter name of the account holder : Ashique
Enter account type : Zero balance account
Enter your balance : 10000
Enter amount to deposit : 1000
Rs 1000.0 deposited ! Current balance is Rs. 11000.0
Enter amount to withdraw : 500
Rs 200.0 is withdrawn ! current balance is Rs 10800.0

Program 37

Aim: Create a class rectangle with private attributes length & width. Overload '<' operator to compare the area of 2 rectangle.

Program

```
class Rectangle:

    def __init__(self, l, w):
        self.__length = l
        self.__width = w
        self.area = self.__width * self.__length

    def __lt__(self, other):
        if self.area < other.area:
            print("Rectangle with length = " self.__length
                  ; and width = " self.__width " has
                    the lesser area !")

    elif
        other.area < self.area:
        print ("Rectangle with length = ", other.__length, " and
               width = " + Other.__width, " has the lesser area")

    else
        print ("They have equal area !)

l = float (Input ("Enter length of 1st rectangle: "))
w = float (Input ("Enter width of 1st rectangle: "))
R1 = Rectangle (l, w)
```

```
l = float (input ("Enter length of 2nd rectangle :"))
w = float (input ("Enter ~~length~~width of 2nd rectangle :"))
R2 = Rectangle (l, w)
R1 < R2
```

Result

The program has been executed and output is verified.

Output

Enter length of 1st rectangle : 5

Enter width of 1st rectangle : 3

Enter length of 2nd rectangle : 9

Enter length of 2nd rectangle : 6

Rectangle with length = 5 and width = 3 has the lesser area.

Program 38

Aim: Create a class Time with private attribute hour, minutes and seconds. Overload '+' operator to find Sum of 2 time.

Program

```
class Time:
    def __init__(self, hh=0, mm=0, ss=0):
        self.__hour = hh
        self.__second = ss

    def __add__(self, other):
        second = int((self.second + other.second) % 60)
        minute = int((self.__minute + other.__minute) % 60 +
                     (self.__second + other.second) % 60))
        hour = int((self.__hour + other.hour) % 24 + (self.__
                   minute + other.__minute) % 60)

    print(time[hh:mm:ss]; hour, ":" minute, ":" second)
T1 = Time(2, 25, 45)
T2 = Time(18, 50, 45)
T1 + T2
```

Result

The program has been executed and output is verified.

Output

Time [hh:mm:ss] 21:16:30