FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

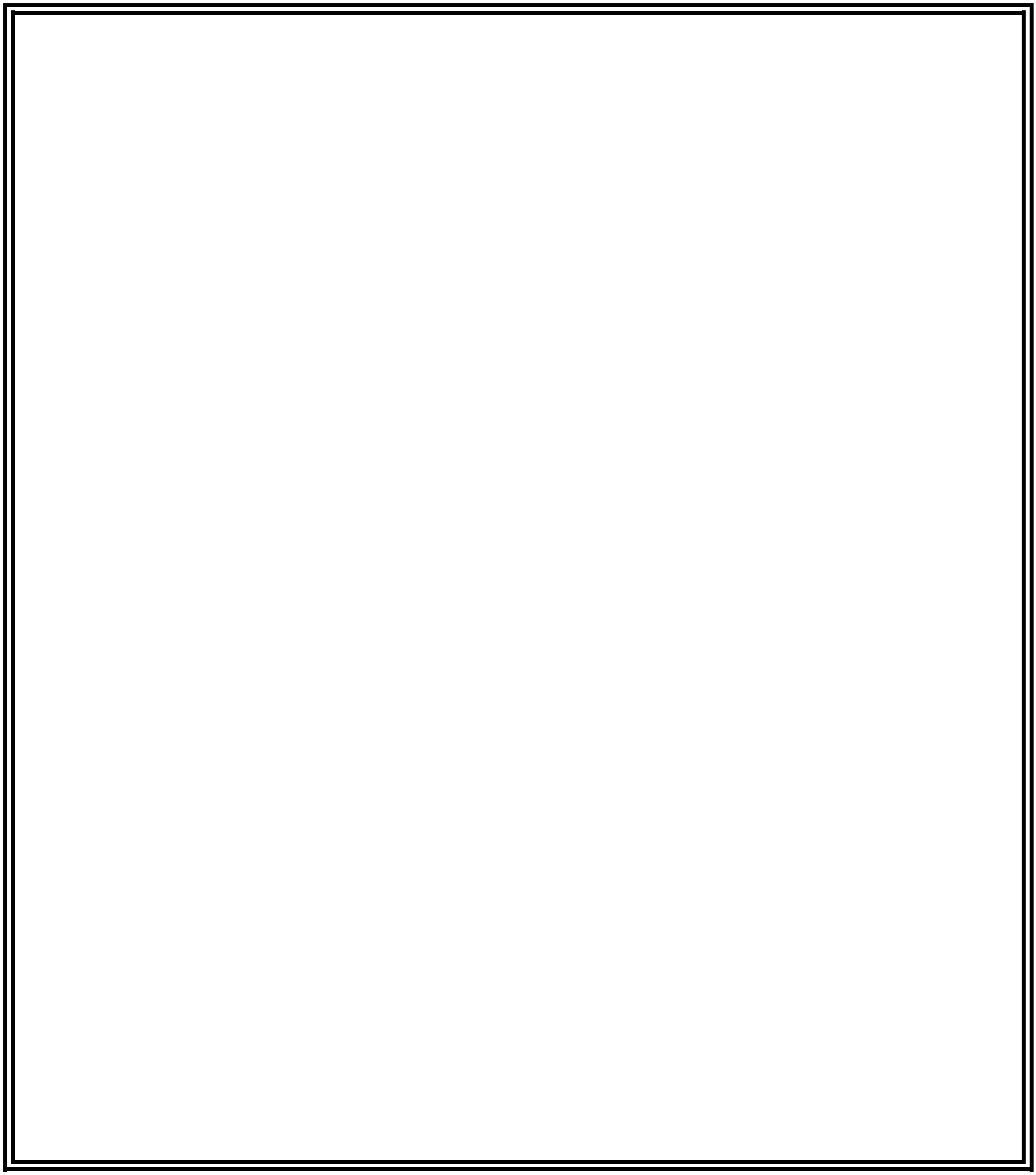
20MCA131 PROGRAMMING LAB LABORATORY RECORD

Name: ANU FRANCIS

Branch: MASTER OF COMPUTER APPLICATIONS

Semester: 1 **Batch: A** **Roll No: 31**

MARCH 2022



FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)™

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577

FOCUS ON EXCELLENCE

CERTIFICATE

*This is to certify that this is a Bonafide record of the Practical work done by **ANU FRANCIS (FIT21MCA-2031)** in the **20MCA131 PROGRAMMING LAB** Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.*

Signature of Staff in Charge

Name:

Signature of H O D

Name:

Date of University practical examination ……

Signature of
Internal Examiner

Signature of
External Examiner

CONTENT

CO1

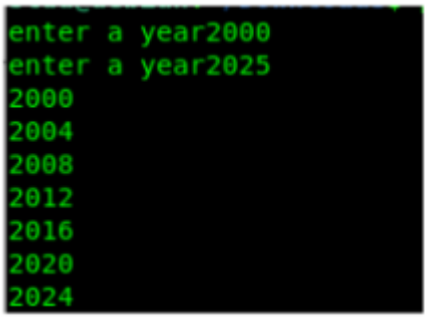
PROGRAM 01:

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

PROGRAM CODE:

```
currentyear=int(input("enter a year"))
year=int(input("enter a year"))
for i in range (currentyear,year,):
if (i%4==0):
print(i)
```

OUTPUT:



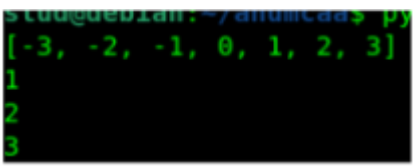
PROGRAM 02:

a) Generate positive list of numbers from given list of numbers:

PROGRAM CODE:

```
n=[-3,-2,-1,0,1,2,3]
for i in n:
if(i>0):
print(i)n=[-3,-2,-1,0,1,2,3]
```

OUTPUT:

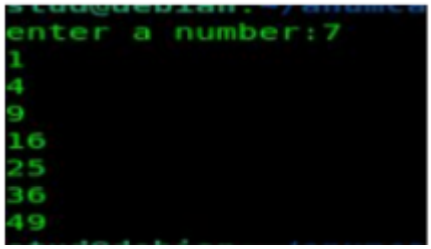


b) square of N numbers:

PROGRAM CODE:

```
n=int(input("enter a number:\n"))
for i in range(1,n+1,1):
y=i*i
print(y)
```

OUTPUT:

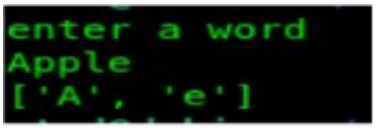


c) list of vowels from a given word:

PROGRAM CODE:

```
L=[]
s=("apple")
for i in s:
if i in "AaEeIiOoUu":
L.append(i)
print(L)
```

OUTPUT:



d) list of ordinal values:

PROGRAM CODE:

```
s=("good")
for i in s:
print(ord(i))
```

OUTPUT:

PROGRAM 03:

count the occurrences of each word in a line of text:

PROGRAM CODE:

```
s=input("Enter a string:")
count=dict()
word=s.split()
for i in word:
if i in count:
count[i]+=1
else:
count[i]=1
print(count)
```

OUTPUT:

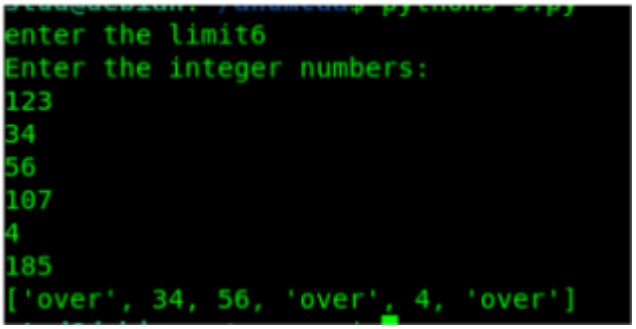
PROGRAM 04:

Prompt the user for list of integers for all values greater than 100 store ‘over instead:

PROGRAM CODE:

```
list=[]
n=int(input(“enter the limit”))
print("Enter the integer numbers:")
for i in range(0,n):
j=int(input())
if j>=100:
list.append('over')
else:
list.append(j)
print(list)
```

OUTPUT:



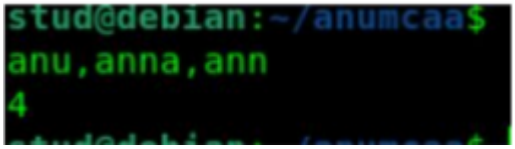
PROGRAM 05:

Store a list of first names count the occurances of 'a' within the list:

PROGRAM CODE:

```
l=[]
r=0
s=("anu,anna,ann")
print(s)
for i in s:
if i in ("aA"):
r=r+1
l.append(i)
print(r)
```

OUTPUT:



PROGRAM 06:

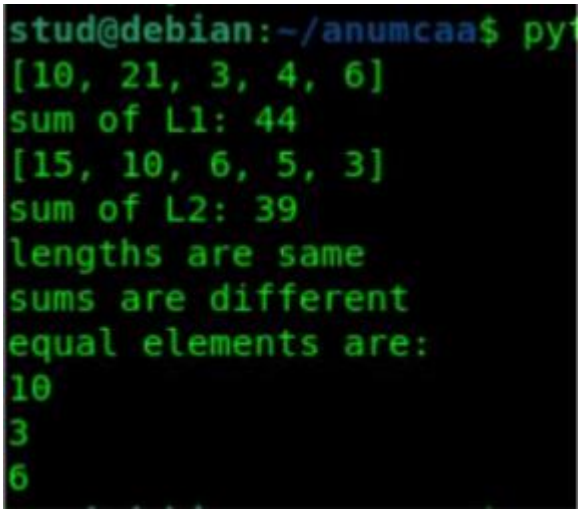
Enter 2 list of integers.check(a) whether list are of same length (b)whether list sums to same value(c)whether any value occur in both:

PROGRAM CODE:

```
L1=[10,21,3,4,6]
L2=[15,10,6,5,3]
print(L1)
sum1=0
sum2=0
for i in range (len(L1)):
sum1=(sum1+L1[i])
print("sum of L1:",sum1)
print(L2)
for j in range(len(L2)):
sum2=sum2+L2[j]
print("sum of L2:",sum2)
if len(L1)==len(L2):
print("lengths are same")
else:

print("lengths are different")
if sum1==sum2:
print("sums are same")
else:
print("sums are different")
print("equal elements are:")
for i in range(len(L1)):
for j in range(len(L2)):
if L1[i]==L2[j]:
print(L1[i])
```

OUTPUT:



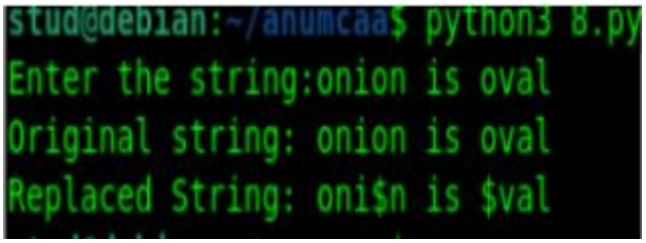
PROGRAM 07:

Get a string from an input string where all occurance of first character replaced with '$' ,except first character:

PROGRAM CODE:

```
str1=input("Enter the string:")
print("Original string:",str1)
char=str1[0]
str1=str1.replace(char,'$')
str1=char+str1[1:]
print("Replaced String:",str1)
```

OUTPUT:



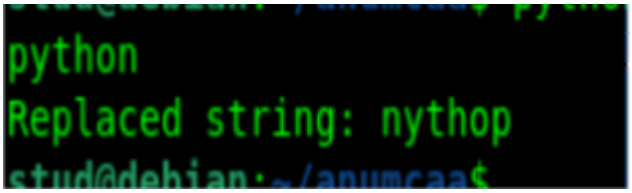
PROGRAM 08:

Create a string from given string where first and last character exchanged:

PROGRAM CODE:

```
S="python"
print(S)
t=S[0]
t1=S[-1]
n=len(S)
nS=t1+S[1:n-1]+t
print("Replaced string:",nS);
```

OUTPUT:



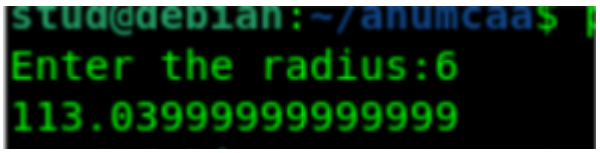
PROGRAM 09:

Accept radius from user and find the area of a circle:

PROGRAM CODE:

```
p=int(input("Enter the radius:"))
a=3.14*p*p
print(a)
```

OUTPUT:



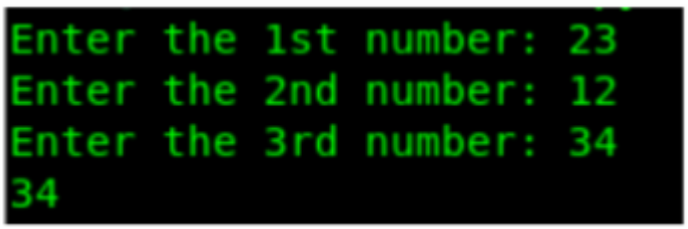
PROGRAM 10:

Find biggest of three number:

PROGRAM CODE:

```
a=int(input("Enter the 1st number: "))
b=int(input("Enter the 2nd number: "))
c=int(input("Enter the 3rd number: "))
if a>b:
if a>c:
print(a)
else:
print(c)
else:
if b>c:
print(b)
else:
print(c)
```

OUTPUT:



PROGRAMM 11:

Accept a file name from user and print extension of that:

PROGRAM CODE:

```
Import os
a=input("enter the file name:")
print("the extension of file",a,'is',os.path.splitext (a))
```

OUTPUT:

```
enter the file name:12.py
the extension of file 12.py is ('12', '.py')
```

PROGRAM 12:

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

PROGRAM CODE:

```
color=[]
color=[i for i in input("Enter the colors separated by coma:\n").split(',')]
print(color)
i=len(color)-1
print("First color:",color[0])
print("Last color:",color[i])
```

OUTPUT:

```
Enter the colors separated by coma:
yellow,red,blue,green,white
['yellow', 'red', 'blue', 'green', 'white']
First color: yellow
Last color: white
stud@debian:~/anumcaa$
```

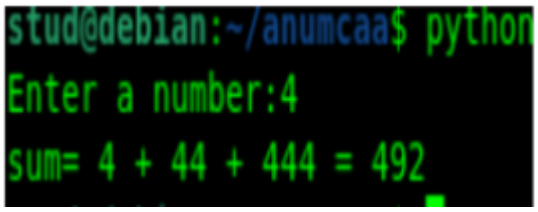
PROGRAM 13:

Accept an integer n and computr n+nn+nnn:

PROGRAM CODE:

```
n=int(input("Enter a number:"))
a=n*1
b=n*11
c=n*111
sum=a+b+c
print("sum=",a,"+",b,"+",c,"=",sum)
```

OUTPUT:



PROGRAMM 14:

Print out all color from color-list1 not contained in color-list2:

PROGRAM CODE:

```
list1=['white','green','blue']
list2=['black','red','blue']
print("list1=",list1)
print("list2=",list2)
list3=[]
for i in list1:
if i not in list2:
list3.append(i)
print("list3=",list3)
```

OUTPUT:

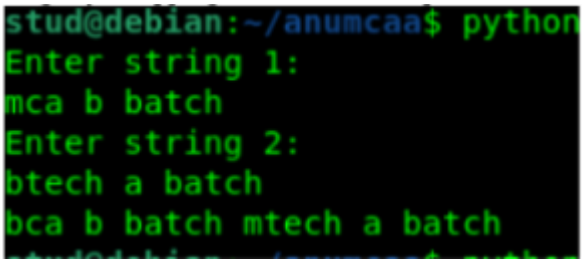
PROGRAM 15:

create a single string separated with space from two strings by swapping the characters at position 1:

PROGRAM CODE:

```
a=input("Enter string 1:\n")
b=input("Enter string 2:\n")
a1=b[0]+a[1:]
b1=a[0]+b[1:]
c=a1+' '+b1
print(c)
```

OUTPUT:



PROGRAM 16:

Sort dictionary in ascending and descending order.

PROGRAM CODE:

```
dict1={"a":1,"c":3,"d":2,"b":4}
l=list(dict1.items())
print(l)
 l.sort()
 print("Ascending Order is \n",l)
 l=list(dict1.items())
 l.sort(reverse=True)
 print("Descending order is \n",l)
```

OUTPUT:

```
[('d', 2), ('c', 3), ('a', 1), ('b', 4)]
Ascending Order is
 [('a', 1), ('b', 4), ('c', 3), ('d', 2)]
Descending order is
 [('d', 2), ('c', 3), ('b', 4), ('a', 1)]
```

PROGRAM 17:

merge 2 dicitionaries:

PROGRAM CODE:

```
d1={"name":"angel","age":"23"}
d2={"sex":"f","qualification":"UG"}
d1.update(d2)
print(d1)
```

OUTPUT:

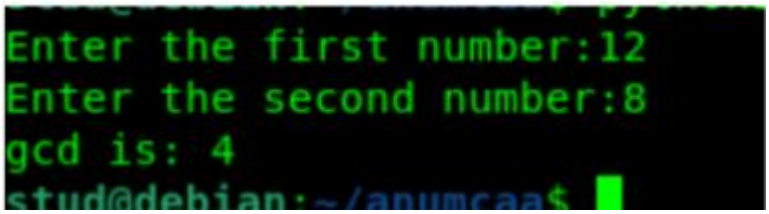
PROGRAM 18:

Find gcd of 2 numbers:

PROGRAM CODE:

```
x=int(input("Enter the first number:"));
y=int(input("Enter the second number:"));
if x>y:
smallest=y
else:
smallest=x
for i in range(1,smallest+1):
if((x%i==0) and (y%i==0)):
gcd=i
print("gcd is:",gcd)
```

OUTPUT:



PROGRAM 19:

From a list of integers,create a list removing even numbers:

PROGRAM CODE:

```
list1=[1,2,3,4,5,6,7,8,9]
num=[]
print(list1)
for i in list1:
if(i%2!=0):
num.append(i)
print(num)
```

OUTPUT:

```
stud@debian:~/anumcaa$ python3 20.py
[1, 2, 3, 4, 5, 6, 7, 8, 9]
[1, 3, 5, 7, 9]
```

CO2

PROGRAM 01:

program to find the factorial of a number:

PROGRAM CODE:

```
n=int(input("enter a number"))
fact=1
for i in range(1,n+1):
fact=fact*i
print(fact)
```

OUTPUT:

```
enter a number:6
720
```

PROGRAM 02:

generate fibonaci series of N terms:

PROGRAM CODE:

```
n=int(input("enter a number"))
f1=0
f2=1
print(f1)
print(f2)
for i in range(2,n):
f3=f1+f2
print(f3)
f1=f2
f2=f3
```

OUTPUT:

```
enter a number7
0
1
1
2
3
5
8
```

PROGRAM 03:

find the sum of all items in a list:

PROGRAM CODE:

```
list=[10,6,2,23]
print("list=",list)
sum=0
for i in list:
sum=sum+i
print("sum=",sum)
```

OUTPUT:

```
list= [10, 6, 2, 23]
sum= 41
```

PROGRAM 04:

generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square:

PROGRAM CODE:

```
limit1=1000
limit2=9999
list1=[]
for i in range(limit1,limit2):
j=i
digit=[]
while(i!=0):
digit.append(i%10)
i=int(i/10)
count=0
for n in digit:
if n%2==0:
count=count+1
if count==4:
for k in range(31,100):
if((k**2)==j):
list1.append(j)
print(k)
print(list1)
```

OUTPUT:

```
68
78
80
92
[4624, 6084, 6400, 8464]
```

PROGRAM 05:

display the given pyramid with step number accepted from user. Eg n=4:

1

2 4

3 6 9

4 8 12 16

PROGRAM CODE:

```
n=int(input("enter a number:"));
for j in range(0,n+1):
for i in range(1,j+1):
i=j*i;
print(i,end=" ");
print("\n")
```

OUTPUT:

```
enter a number:6


1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

6 12 18 24 30 36
```

PROGRAM 06:

count the numbers of characters(character frequency) in a string.

PROGRAM CODE:

```
string=input("Enter a string:")
list1=[]
for i in string:
if i not in list1:
list1.append(i)
for i in list1:
count=0
for j in string:
if(i==j):
count=count+1
print(i,"\t:",count)
```

OUTPUT:

```
Enter a string:mcaAbatch
m       : 1
c       : 2
a       : 2
A       : 1
b       : 1
t       : 1
h       : 1
```

PROGRAM 07:

Add ‘ing’ at the end of a given string.If it already ends with ‘ing’,then add ‘ly’:

PROGRAM CODE:

```
str1=input("enter a string:")
if str1[-3:]=='ing':
str1=str1+'ly'
else:
str1=str1+'ing'
print("changed string:",str1)
```

OUTPUT:

```
enter a string:song
changed string: songing
```

```
enter a string:sing
changed string: singly
```

PROGRAM 08:

accept a list of words and return length of longest word:

PROGRAM CODE:

```
list=[]
length=[]
print("enter 5 words")
for i in range (5):
str=input()
list.append(str)
for j in list:
length.append(len(j))
print("length of longest word is:",max(length))
```

OUTPUT:

```
enter 5 words
class
standard
dance
song
division
length of longest word is: 8
```

PROGRAM 09:

Construct following patterns using nested loop:

```
*

* *

* * *

* * * *

* * * * *

* * * *

* * *

* *

*
```

PROGRAM CODE:

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

print('*',end=" ");
print("\n");
```

OUTPUT:

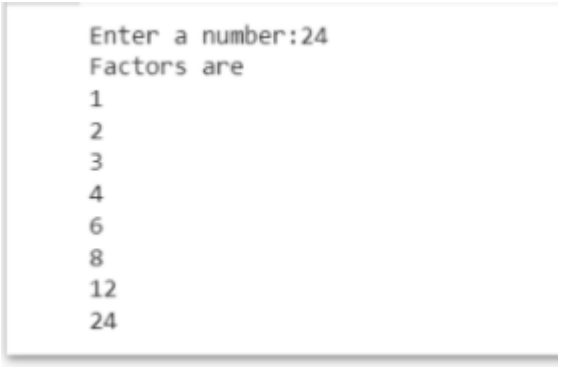
PROGRAM 10:

Generate all factors of a number:

PROGRAM CODE:

```
n=int(input("Enter a number:"))
print("Factors are")
for i in range(1,n+1):
if(n%i==0):
print(i)
```

OUTPUT:



CO3

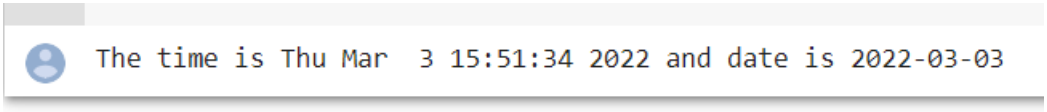
PROGRAM 01:

Work with built-in packages.

PROGRAM CODE:

```
import time
import datetime
today=datetime.date.today()
print(f"The time is {time.ctime()} and date is {today}")
```

OUTPUT:



PROGRAM 02 :

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

PROGRAM CODE:

#graphics.py

```
import graphics
from graphics import circle,rectangle
from graphics.tdgraphics import cuboid,sphere
from graphics.circle import *
print("Area of a circle with radius 12 is:",circle.area_circle(12))
print("Permeter of a circle with radius 12 is:",circle.perimeter_circle(12))
print("\n")
print("Area of a Rectangle with length and width 12 is:",rectangle.area_rec(12,12))
print("Permeter of a Rectangle with length and width 12 is:",rectangle.perimeter_rec(12,12))
print("\n")
print("Area of a cuboid with length,width,height 12 is:",cuboid.area_cuboid(12,12,12))
print("Volume of a cuboid with length,width,height 12 is:",cuboid.volume_cuboid(12,12,12))
print("\n")
print("Area of a spere with radius 12 is:",sphere.area_sphere(12))
print("Permeter of a spere with radius 12 is:",sphere.perimeter_sphere(12))
```

#circle.py

```
def cir_area(r):
 print("area of circle:",3.14*r*r)
def cir_perimeter(r):
 print("perimeter of circle:",2*3.14*r)
```

#rectangle.py

```
def rect_area(l,b):
 print("area of rectangle:",l*b)
def rect_area(l,b):
 print("perimeter of rectangle:",2*(l+b))
```

#cuboid.py

```
def cuboid_area(lb,bh,hl):
 print("area of cuboid is:",2*(lb+bh+hl))
def cuboid_perimeter(l,b,h):
 print("perimeter of the cuboid:",4*(l+b+h))
```

#sphere.py

```
def sphere_circum(r):
 printf("circuference of sphere:",2*3.14*r)
def sphere_area(r):
 print("surface area of sphere:",4*3.14*r*r)
```

```
stud@debian:~/anumcaa/python$ mkdir graphics
stud@debian:~/anumcaa/python$ cd graphics
stud@debian:~/anumcaa/python/graphics$ gedit circle.py
stud@debian:~/anumcaa/python/graphics$ gedit rectangle.py
stud@debian:~/anumcaa/python/graphics$ mkdir tdgraphics
stud@debian:~/anumcaa/python/graphics$ cd tdgraphics
stud@debian:~/anumcaa/python/graphics/tdgraphics$ gedit cuboid.py
stud@debian:~/anumcaa/python/graphics/tdgraphics$ gedit sphere.py
stud@debian:~/anumcaa/python/graphics/tdgraphics$
```

OUTPUT:

```
stud@debian:~/anumcaa/python$ python3 driver.py
Area of a circle with radius 12 is: 452.3893421169302
Permeter of a circle with radius 12 is: 75.39822368615503


Area of a Rectangle with length and width 12 is: 144
Permeter of a Rectangle with length and width 12 is: 48


Area of a cuboid with length,width,height 12 is: 864
Volume of a cuboid with length,width,height 12 is: 1728


Area of a spere with radius 12 is: 1809.5573684677208
Permeter of a spere with radius 12 is: 75.39822368615503
stud@debian:~/anumcaa/python$
```

CO4

PROGRAM 01 :

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

PROGRAM CODE:

```
class rectangle:
  def area(self,l,b):
    a=l*b
    return(a)
  def peri(self,l,b):
    p=2*(l+b)
    print(p)
rect=rectangle()
rect1=rectangle()
a2=rect.area(4,6)
a1=rect1.area(3,5)
rect.peri(2,7)
if a1 > a2:
  print("first rectange has more area")
else:
```

```
print("second rectangle has more area")
```

OUTPUT:

```
18
second rectangle has more area
```

PROGRAM 02 :

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 CODE:

```
class Bank_account:
 def __init__(self,acc_no,name,acc_type,balance):
  self.acc_no=acc_no
  self.name=name
  self.acc_type=acc_type
  self.balance=balance
 def deposit(self,deposit_am):
  print("Initial balance:",self.balance)
  print("Amount to be deposited:",deposit_am)
  self.balance=self.balance+deposit_am
  print("New balance is:",self.balance)
 def withdraw(self,withdrawn_am):
  print("current balance:",self.balance)
  print("amount_withdrawn:",withdrawn_am)
  self.balance=self.balance-withdrawn_am
  print("New Balance is:",self.balance)
P=Bank_account(1200,'Anu','savings',102000)
P.deposit(20000)
P.withdraw(30000)
```

OUTPUT:

```
Initial balance: 102000
Amount to be deposited: 20000
New balance is: 122000
current balance: 122000
amount_withdrawn: 30000
New Balance is: 92000
```

PROGRAM 03 :

Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.

PROGRAM CODE:

```
class rectangle:
 def getdata(self,l,r):
  self.length=l;
  self.breadth=b;
 def area(self,l,b):
  area=l*b;
  return area;
 def overload(self,a1,a2):
 if a1>a2:
  print('1 greater than 2')
 else:
  print('2 greater than 1')
rect1=rectangle()
rect2=rectangle()
a1=rect1.area(3,5)
a2=rect2.area(4,8)
rect=rectangle()
rect.overload(a1,a2)
```

OUTPUT:

```
2 greater than 1
```

PROGRAM 04:

Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time.

PROGRAM CODE:

```
class Time(object):
  def __init__(self, hours, minutes,seconds):
    self.hours = hours
    self.minutes = minutes
    self.seconds=seconds
  def addTime(t1, t2):
    t3 = Time(0, 0,0)
```

```
        t3.hours = t1.hours + t2.hours
    t3.minutes = t1.minutes + t2.minutes
    t3.seconds = t1.seconds + t2.seconds
    while t3.minutes >= 60:
      t3.hours += 1
      t3.minutes -= 60
    return t3
  def displayTime(self):
    print("Time is %d hours and %d minutes %d seconds" %(self.hours, self.minutes,self.seconds))
  def displayMinutes(self):
    print((self.hours * 60) + self.minutes, "minutes")

a = Time(4, 40, 30)
b = Time(1, 50, 10)
c = Time.addTime(a,b)
c.displayTime()
```

OUTPUT:

```
Time is 6 hours and 30 minutes 40 seconds
```

PROGRAM 05 :

Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.

PROGRAM CODE:

```
class Publisher:
  def __init__(self,n):
    self.name=n
class Book(Publisher):
  def __init__(self,n,a,t):
    super().__init__(n)
    self.title=t
    self.author=a
class Python(Book):
  def __init__(self,n,a,t,p,pg):
    super().__init__(n,a,t)
    self.price=p
    self.pages=pg
  def Print(self):
```

```
    print(P.name)
    print(P.title)
    print(P.author)
    print(P.price)
    print(P.pages)
P=Python('DC_Books','Programming_Lab','Luke John',550,150)
P.Print()
```

OUTPUT:

```
DC_Books
Luke John
Programming_Lab
550
150
```

CO5

PROGRAM 01:

Write a Python program to read a file line by line and store it into a list.

PROGRAM CODE:

text_file.txt

Federal Institute of Science And Technology (FISAT) is a private, self-financing engineering college, established and run by the Federal Bank Officers' Association Educational Society (FBOAES). It is one of the top private engineering colleges in Kerala. It is an initiative of the Federal Bank Officers' Association (FBOA), the sole representative body of the entire officers of the Federal Bank. FISAT is accredited by NBA and NAAC.FISAT is set up at Mookannoor, near Angamaly in Ernakulam District, Kerala, the birthplace of the founder of The Federal Bank Ltd, Late K.P Hormis.

5c1.py

```
fp=open("text_file.txt",'r')
lines=[]
for line in fp:
 lines.append(line.strip())
print(lines)
```

OUTPUT:

```
["Federal Institute of Science And Technology (FISAT) is a private, self-financi
ng engineering college, established and run by the Federal Bank Officers' Associ
ation Educational Society (FBOAES). It is one of the top private engineering col
leges in Kerala. It is an initiative of the Federal Bank Officers' Association (
FBOA), the sole representative body of the entire officers of the Federal Bank.
FISAT is accredited by NBA and NAAC.", '', 'FISAT is set up at Mookannoor, near
Angamaly in Ernakulam District, Kerala, the birthplace of the founder of The Fed
eral Bank Ltd, Late K.P Hormis.']
```

PROGRAM 02:

Write a Python program to read each row from a given csv file and print a list of string.

PROGRAM CODE:

```
import csv
with open('co5.csv','r') as file:
 reader = csv.reader(file)
 for row in reader:
  print(row)
```

OUTPUT:

```
stud@debian:~/anumcaa/python$ python3 pgrm40.py
['name place age']
['anu mattom 22']
['alan alkode 21']
['ansa kochi 22']
['ammu kechery 22']
stud@debian:~/anumcaa/python$
```