

10/02/2021

Program no: 29

20

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

Program:

class Rectangle :

def __init__(self, len, bth):

self.len = len

self.bth = bth

self.len2 = len

self.bth2 = bth

def myfun(hell):

hell.len = hell.len * hell.bth

print ("Area of rectangle = ", + hell.len)

return hell.len

def perimeter(self):

self.len2 = 2 * (self.len2 + self.bth2)

print ("Perimeter of rectangle = ", + self.len2)

return self.len2

```
print ("Enter length and breadth of the first rectangle : ")
```

```
l1 = int (input ())
```

```
b1 = int (input ())
```

```
c1 = Rectangle (l1, b1)
```

```
c1 · perimeter
```

```
a = c1 · myfun ()
```

```
print ("Enter length and breadth of second rectangle")
```

```
l2 = int (input ())
```

```
b2 = int (input ())
```

```
c2 = Rectangle (l2, b2)
```

```
c2 · perimeter
```

```
b = c2 · myfun ()
```

```
if (a == b):
```

```
    print ("Area of both rectangles are equal")
```

```
if (a > b):
```

```
    print ("Area of Rectangle 1 is greater")
```

```
if (a < b):
```

```
    print ("Area of Rectangle 2 is greater")
```

Output :

) Enter length and breadth of first rectangle.

5

6

Perimeter of rectangle = 22

Area of Rectangle = 30

Enter length and breadth of second rectangle:

7

4

Perimeter of rectangle = 22

Area of Rectangle = 28

Area of Rectangle 1 is greater.

10/10/21
2021

23

Aim : Create a Bank account with members account number, name, type of account and balance . write constructor and methods to deposite at the bank and withdraw an amount from the bank .

Program :

class Bank :

```
def __init__(self, accountnumber, number, typeofaccount, balance):
    self.accountnumber = accountnumber
    self.name = name
    self.typeofaccount = typeof account
    self.balance = balance
    print("Account number : ", + self.accountnumber)
    print("Name : ", self.name)
    print("Type of account : ", self.typeofaccount)
    print("Account Balance : ", self.balance)
```

def deposit (deposit, money):

deposit.balance = deposit.balance + money

print (money, "deposited . Account balance : ", + deposit.balance)

```
def withdraw (withdraw, money):
    if (money > withdraw.balance):
        print ("Not enough money to withdraw")
    else:
        withdraw.balance = withdraw.balance - money
        print (money, "withdrawn from the account. Balance:",
               + withdraw.balance)
```

accountnumber = int(input ("Enter account number :"))

name = str(input ("Enter name :"))

a = Bank (accountnumber, name, "Normal", 0)

while (1)

print ("Enter 1 to deposit money. Enter 2 to withdraw
money. 3 to exit")

toggle = int(input ())

if (toggle == 1):
 dep = int(input ("Enter the amount to deposit"))
 a.deposit (dep)

if (toggle == 2):

draw = int(input ("Enter the amount to withdraw"))
 a.withdraw (draw)

```
if (toggle == 3):  
    break;
```

Result :

The program has been executed and the output
was verified.

Output :

1) Enter account number : 1001

Enter name : Aravind

Account number : 1001

Name : Aravind

Type of account : Normal

Account Balance : 0

Enter 1 to deposite money. Enter 2 to withdraw money.

3 . to exit

1

Enter the amount to deposit : 10000

10000 deposited . Account balance : 10000

1

Enter the amount to deposit : 15000

15000 deposited . Account balance : 25000

2

Enter the amount to withdraw : 5000

5000 withdrawn . Account balance : 20000

10/07/2021 Program no: 31

Aim :

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

Program:

class Rectangle :

def __init__(self, length, breadth):

 self.length = length

 self.breadth = breadth

 self.a = self.length * self.breadth

 print("Area of rectangle is : ", + self.a)

def __lt__(self, other):

 if (self.a < other.a):

 return True

print("Enter length and breadth of Rectangle 1 : ")

l1 = int(input())

b1 = int(input())

r1 = Rectangle(l1, b1)

print("Enter the length and breadth of Rectangle 2 : ")

l2 = int(input())

b2 = int(input())

$\text{r1} = \text{Rectangle}(12, 62)$

$\text{if } (\text{r1} < \text{r2}):$

$\text{print } (" \text{Area of Rectangle 2 is less than Rectangle 1}")$

else :

$\text{if } (\text{r1} < \text{r2}):$

$\text{print } (" \text{Area of Rectangle 1 is less than Rectangle 2}")$

Result:

The program has been executed and the output was verified.

Output :

1) Enter length and breadth of Rectangle 1 :

5, 4

Area of rectangle is : 20

Enter length and breadth of Rectangle 2 :

3, 2

Area of rectangle is : 6

Area of Rectangle 2 is less than Rectangle 1

10/10/21
2021

Program no: 32

Aim:

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

Program:

class Time :

def __init__(self, hour, minute, second):

self.__hour = hour

self.__minute = minute

self.__second = second

def __add__(self, other):

print("Added hours ", +self.__hour+other.__hour)

print("Added minutes ", +self.__minute+other.__minute)

print("Added seconds ", +self.__second+other.__second)

print("Enter first hour, minute and second :")

h1 = int(input())

m1 = int(input())

s1 = int(input())

t1 = Time(h1, m1, s1)

```
print ("Enter second hour, minute and second :")
```

```
h2 = int (input ())
```

```
m2 = int (input ())
```

```
s2 = int (input ())
```

```
t2 = Time (h2, m2, s2)
```

```
print ("Sum of two times :")
```

```
print (t1 + t2)
```

Result :

The program has been executed and the output was verified.

Output

1) Enter first hour, minute and second :

2

55

23

Enter second hour, minute and second :

13

43

10

Sum of two time :

Added hour 15

Added minutes 98

Added seconds 33

10/02/
2021

Program no. 33

Aim: Create a class Publisher(name). Derive class Book from Publisher with attribute 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:

class Publisher:

```
def __init__(self, title, author):  
    self.title = title  
    self.author = author
```

class Book(Publisher):

```
def __init__(self, title, author, price, page):  
    Publisher.__init__(self, title, author)  
    self.price = price  
    self.page = page
```

```
class Python(Book):  
    def __init__(self, title, author, price, page):  
        Book.__init__(self, title, author, price, page)  
        print("Book Price : ", self.price)  
        print("Book No of Pages : ", self.page)  
        print("Book Title : ", self.title)  
        print("Book Author : ", self.author)  
  
obj = Python ("Python Programming", "Krushal", 250, 400)
```

Result :
The program has been executed and the output
was verified.

Output

i) Book Price : 250

Book No. of pages : 400

Book Title : Python Programming

Book Author : Kruskal

21/02/
2021

Program no. 34

Aim :

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

Program :

```
list = []
a = open ('file1.txt', 'r')
c = a.readlines()
print ("Contents of this file")
print (c)
print ("File stored in a list")
list.append (c)
print (list)
```

Output :

i) contents of this file

['Hello python

python is a good programming language

Python is dynamic ']

List stored :

[['Hello python

Python is a good programming language

Python is dynamic']]

Program no: 35

Aim: Python program to copy odd lines of one file to other.

Program:

```
a = open ('text.txt', 'r')  
b = open ('text2.txt', 'w')  
c = a.readlines()  
  
for i in range (0, len(c)):  
    if (i%2 == 0):  
        b.write(c[i])  
  
    else:  
        pass  
  
b.close()  
b = open ('text2.txt', 'r')  
d = b.read()  
print ("Copied File :")  
print (d)  
a.close()  
b.close()  
a = open ('text.txt', 'r')  
d = a.read()  
print ("Original File :")
```

```
print(a)  
a.close()
```

Result

The program has been executed and result was verified.

Output

original file :

Hello python

Python is a good programming language

Python is dynamic

copied file :

Hello python

Python is dynamic

21/02/2021

Program no: 36

Aim :

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

Program :

```
import csv  
with open ('Sample100.csv', newline = '') as csvfile:  
    d = csv.reader(csvfile, delimiter = ',', quotechar = "'")  
    for r in d:  
        print(r)
```

Result :

The program has been executed and the output was verified.

Output

['Serial', 'Company']

['1', 'Companyname']

['2', 'Companyname']

['3', 'Companyname']

['4', 'Companyname']

etc:

Program no: 37

11/02/2021

Aim:

Write a Python program to read specific columns of a given CSV file and print the content of the columns.

Program :

```
import csv
with open ('Sample100.csv', newline='') as csvfile:
    d = csv.DictReader (csvfile)
    print ("Serial no      Company Name")
    for r in d:
        print (r['Serial'], r['Company Name'])
```

Result :

The program has been executed and the output verified.

Output

SERIAL NO

COMPANY NAME

9.78871

TALES OF SHNA

9.8701

GOD OF SMALL THINGS

9.78936

59 Flags

etc.

11/09/2021

Program no: 38

Aim:

Write a Python program to write a Python dictionary to a csv file. After writing csv file read the csv file and display the content.

Program:

```
import csv
CSV-columns = ['No', 'Name', 'Country']
dict-data = [
    {'No': 1, 'Name': 'Agrivind', 'Country': 'India'},
    {'No': 2, 'Name': 'Anand', 'Country': 'France'},
    {'No': 3, 'Name': 'Ebin', 'Country': 'Italy'},
    {'No': 4, 'Name': 'Me', 'Country': 'Russia'},
    {'No': 5, 'Name': 'Amal', 'Country': 'USA'}]
csv-file = "New.csv"
with open(csv-file, 'w') as csvfile:
    w = csv.DictWriter(csvfile, fieldnames=CSV-columns)
    w.writeheader()
    for data in dict-data:
        w.writerow(data)
```

```
with open ('New.csv', newline= '') as csvfile :
```

```
    d = csv.reader(csvfile, delimiter = ',', quotechar = '')
```

```
    for r1 in d:  
        print(r1)
```

Result:

The program has been executed and the output
was verified.

Output

['No, Name, Country']

['1, Amal, India']

['2, Anand, France']

['3, Ebin, Italy']

['4, Me, Russia']

Anavind . v.v

MCA SI a

Roll no: 24

Programming Lab

Record

1/01/1
2021

Program no: 1

COI. 11

Aim: Find the greatest of the 3 given integers.

Program:

```
print ("Enter three numbers one by one")
```

```
x = int (input ())
```

```
y = int (input ())
```

```
z = int (input ())
```

```
if x = y = z :
```

```
    print ("Elements are equal")
```

```
else :
```

```
    if x > y and x != z :
```

```
        if x > z :
```

```
            print (x, "is greater")
```

```
        else :
```

```
            print (z, "is greater")
```

if $y > x$ and $y = z$:

if $y > z$:

print (y , " is greater")

else: print (z , " is greater")

if $x = y$:

if $x > z$:

print ("First and second elements
are equal and greatest")

else: print (z , " is greater")

if $x = z$:

if $x > y$:

print ("First and Third numbers
are equal and greatest")

else: print (y " is greater")

if $y = z$:

if $y > x$:

print ("Second and Third numbers
are equal and greatest")

else: print (x " is greater number")

Result:

The program has been executed and the output was verified.

Output

-) Find the biggest of 3 numbers entered by the user

Enter the three numbers one by one

4, 5, 6

6 is greater

-) Enter three numbers one by one

2, 33, 33

Second and Third numbers are both equal and great

-) Enter three numbers one by one :

7, 6, 5

7 is greater :

17/01/

2021

Program no: 2

col 36

Aim:

Program to find the square of an integer

Program:

```
print ("Enter a number")
```

```
x = int (input ())
```

```
p = int (x * x)
```

```
print ("The square of the number is ")
```

```
print (p)
```

Result :

The program was executed and output was verified.

Output

.) Square of the given number

Enter the number : 7

Square of the number is : 49

.) Enter the number : 76

Square of the number is : 5776

17/01/

2021

Program no: 3

COI - 10

Aim :

Program to find the area of circle

Program :

```
print ("Enter the radius of circle")
```

```
def area(x):
```

```
    print ("Area of circle is ")
```

```
a = float (3.14 * (x * x))
```

```
return a
```

```
print (area (int (input ())))
```

Result:

The program has been executed and the output was verified.

Output

) Area of Circle

Enter the radius of circle : 7

Area of circle is : 153.86 cm^2

) Area of Circle

Enter the radius of circle : 89

Area of circle is : 24871.94

27/11/
2021

Program no: 4

col. 2

Aim:

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

Program:

```
a = int(input("Enter a future year:"))

if (a > 2021):
    if (a > 2023):
        print ("Leap years between 2021 and ", +a, "are:")
        for i in range(2021, a+1):
            if (i%4) == 0:
                if (i%100) == 0:
                    if (i%400) == 0:
                        print(i)
                    else:
                        print(i)
                else:
                    print("No leap year between
                          2021 and ", +a)
            else:
                print ("Year entered is not a future year")
```

Result:

The program has been executed and output was verified.

Output:

) Enter a future year: 2046

Leap years between 2021 and 2046 are:

2024

2028

2032

2036

2040

2044

) Enter a future year: 2023

No leap years between 2021 and 2023

) Enter a future year: 1998

Year entered is not a future year.

27/01/

2021

Program no: 5

COI 3C

Aim : Form a list of vowels selected from a given word .

Program :

```
print ("Enter a word :")
```

```
word = str (input ())
```

```
list = []
```

```
for i == 'a' or i == 'e' or i == 'i' or i == 'o' or  
      (i == 'u') :
```

```
    list.append (i)
```

```
print ("The vowels on the word , " word , " are :")
```

```
print (list)
```

Result :

The program has been executed and output was verified .

Output:

.) Enter a word : elephant

The vowels on the word elephant are :

['e', 'e', 'e']

.) Enter a word : umbrella

The vowels on the word umbrella are :

['u', 'e', 'a']

.) Enter a word : how are you

The vowels on the word how are you are :

['o', 'a', 'e', 'o', 'u']

Aim: Count the occurrence of each word in a line of text.

Program:

```
text = str(input("Enter a string : "))

text = text.lower()

words = text.split(" ")

print(words)

d = dict()

print("Occurrence of each words are : ")

for word in words:
    if word in d:
        d[word] = d[word] + 1
    else:
        d[word] = 1

for key in list(d.keys()):
    print(key, ":", d[key])
```

Result:

The program has been executed and output verified.

Output:

.) Enter a string: how are you are you fine

Occurrence of each words are :

how : 1

are : 2

you : 2

fine : 1

.) Enter a string : python is simple python is
dynamic Python is easy to learn

Occurrence of each words are :

python : 3

is : 3

simple : 1

dynamic : 1

easy : 1

to : 1

learn : 1

27/01/2021

Program no. 7

co1.6

Aim: Store a list of first names. Count the occurrence of 'a' with the list.

Program:

```
print ("List of first names are : ")
```

```
list = ["aravind", "anand", "amal", "alent", "bibin",
        "bello", "ghiron", "hani", "arun"]
```

```
print (list)
```

```
count = 0
```

```
for a in range (0, len (list)):
```

```
    for m in l[a]:
```

```
        if m == 'a':
```

```
            count = count + 1
```

```
print ("The number of times 'a' occurs in list is : ")
```

```
print (count)
```

Result:

The program has been executed and result verified.

Output

The List of first names are :

[" anovind ", " anand ", " amal ", " alent ", " bibin ",
" bello ", " shivan ", " hari ", " arun "]

The number of times 'a' occur in list is :

10

7/01/
2021

Program no: 8

col. 7

Aim: Enter 2 lists of integers. Check (a)

wheather list are of some length (b) wheather
list sums to some value (c) wheather any
value occurs in both.

Program:

list 1 = []

list 2 = []

n = int (input ("Enter the size of list 1: "))

m = int (input ("Enter the size of list 2: "))

print ("Enter integer elements of list 1")

for i in range (0, n):

 value = int (input ())

 list 1.append (value)

print ("Enter integer elements of list 2")

for i in range (0, m):

 value = int (input ())

 list 2.append (value)

```
a = len(list 1)  
b = len(list 2)  
c = 0  
d = 0
```

if $a == b$:

 print ("Lists are equal in length ", +a)

else:

 print ("Lists are not equal in length : ", +a, ",
 and ", +b, ",
 are the lengths")

for i in range (0, len(list 1)):

 c = c + list1[i]

for i in range (0, len(list 2)):

 d = d + list2[i]

if $c == d$:

 print ("Sum of integers are equal in both list : ", +c)

else:

 print ("Sum of integers in list 1 : ", +c, " and sum
 of integers in list 2 : ", +d, " are not equal")

```
flag = 0
print (" Values occurring in both lists are : ")
for i in list 1 :
    for j in list 2 :
        if i == j :
            print (i)
            flag = 1
if (flag == 0) :
    print (" No some element occur in both list ")
```

Result :

The result program has been executed and
output was verified .

Output

.) Enter the size of list 1 : 4

Enter the size of list 2 : 3

Enter integer element of list 1 : 66, 5, 3, 90

Enter integer elements of list 2 : 9, 4, 151

Lists are not equal : 4 and 3 are the lengths

sum of integers are equal in both list : 164

values occurring in both lists are :

No esome element occur in both the list.

) Enter the size of list 1 : 5

Enter the size of list 2 : 5

Enter integer elements of list 1 : 3, 6, 5, 8, 71

Enter integer elements of list 2 : 9, 23, 6, 5, 21

Lists are equal in length : 5

Sum of integers in list 1 : 99 and sum of
integers in list 2 : 64 are not equal.

Values occurring in both lists are :

6

5

27/01/

2021

Program no: 9

co1.8

Aim: Get a string from an input string

where all occurrence of first character replaced
with '\$' except first character.

[eg: onion \rightarrow on\$in]

Program:

```
a = str(input("Enter a string :"))
```

```
c = str()
```

```
d = len(a)
```

```
flag = 0
```

```
for b in range(2, d):
```

```
    if a[b] == a[0]:
```

```
        c = a.replace(a[b], '$')
```

```
    flag = 1
```

```
if (flag == 1):
```

```
    e = a[0] + c[1:]
```

```
print ("Replaced string is :")
```

```
print (e)
```

```
else :
```

```
    print ("Replaced string is :")
```

```
print (a)
```

Result :

The program has been executed and
the output was verified.

Output :

i) Enter a string : antennq

String replaced with \$ is :

antenn\$

ii) Enter a string : apple orange almond avacando

String replaced with \$ is :

apple orang\$ almond \$v\$cndo

27/01/1
2021

Program no: 10

co1 q

Aim: Create a string from given string

where first and last characters exchanged.

[eg : python \rightarrow nythop]

Program:

```
a = str(input("Enter the string:"))
```

```
print ("String after exchanging first and last characters  
are")
```

```
b = a[-1:] + a[1:-1] + a[:1]
```

```
print (b)
```

Result:

The program has been executed and the output was verified.

Output

-) Enter the string

umbrella

String after first and last characters exchanged

are :

ambrellu

.-) Enter a string : java is robust

String after first and last characters exchanged

are :

tava is robusj

27/01/
2021

Program no: 11

01-13

Aim :

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

Program:

```
sl = []
n = int(input("Enter the list size : "))
print("Enter", n, "color names : ")
for i in range(0, n):
    item = input()
    sl.append(item)
print("The list of colors is : ", sl)
print("First and last colors in list are : ")
print(sl[0], "and", sl[n-1])
```

Result:

The program has been executed and the result was verified.

Output

) Enter the size of list : 6

Enter 6 color names :

black

white

red

blue

yellow

green

The list of colors is : ['black', 'white', 'red', 'blue',
'yellow', 'green']

First and last colors in list are :

black and green

27/01/

2021

Program no: 12

co1.14

Aim:

Accept an integer n and compute

$$n + nn + nnn .$$

Program:

```
n = int(input("Enter a number n: "))
```

```
m = str(n)
```

```
n1 = m + m
```

```
n2 = m + m + m
```

```
res = n + int(n1) + int(n2)
```

```
print("The value is : ", res)
```

Result:

The program is executed and the output was verified.

Output

.) Enter a number n : 7

The value is : 861

.) Enter a number n : 27

The value is : 275481

Aim: Print out all colors from color-list 1 not contained in color-list 2.

Program:

```
color-list1 = ["black", "white", "yellow", "blue"]
```

```
color-list2 = ["black", "white", "yellow"]
```

```
color-list3 = list(set(color-list1) - set(color-list2))
```

```
print(color-list3)
```

```
print("has all elements of color-list 1 not contained in  
colorlist2")
```

Result:

The program has been executed and the output has been verified -

Output

.) color-list 1 = ["black", "white", "yellow", "blue"]

color-list 2 = ["black", "white", "yellow",]

color-list 3 = ["blue"]

∴ all elements of colorlist-1 not contained
in color-list 2.

10/2
2021

Program no: 14

co1.17

Aim : Sort dictionary in ascending
and descending order.

Program :

```
d2 = {'2': 7, '1': 9, '3': 11, 'a': 8}

print ("Original dictionary : ", d2)

print ("Sorted in ascending : ")

for i in sorted(d2):
    print ("", i, ":", d2[i], "", "", end="")

print ("Sorted in descending : ")

for i in sorted(d2, reverse=True):
    print ("", i, ":", d2[i], "", "", end="")
```

Result :

The program was executed and the output
was verified.

Output

.) Original dictionary :

$$\{ '2': 7, '1': 9, '3': 11, 'a': 8 \}$$

Sorted in ascending :

$$1: 9, 2: 7, 3: 11, a: 8$$

Sorted in descending :

$$a: 8, 3: 11, 2: 7, 1: 9$$

Program no : 15

co1-18

10/21
2021

Aim : Merge two dictionaries

Program :

```
print ("Program to merge two dictionaries")
```

```
print ("Dictionary 1")
```

```
d1 = { 'a': "Hari", 'b': "Shivon", 'c': "Hello"}
```

```
print (d1)
```

```
print ("Dictionary 2")
```

```
d2 = { 'd': 7, 'e': 9, 'f': 11}
```

```
print (d2)
```

```
print ("Merged dictionary : ")
```

```
d1.update(d2)
```

```
print (d1)
```

Result :

The program was executed and the

output was verified .

Output :

.) Program to merge two dictionaries

Diction ary 1

{'a': 'Hari', 'b': 'Shivon', 'c': 'Hello'}

Dictionary 2

{ 'd': 7, 'e': 9, 'f': 11 }

Merged dictionary :

```
{'a': 'Hari', 'b': 'Shivan', 'c': 'Hello', 'd': 7, 'e': 9  
     'f': "}"}
```

Program no. 16

col-19

21/11/
2021

Aim: Find gcd of 2 numbers.

Program :

import math

print ("Enter 2 numbers to find the gcd")

a = int(input ()))

b = int(input ()))

print ("GCD of given numbers is : ")

print (math.gcd (a, b))

Result :

The program was executed and the output verified.

Output :

) Enter 2 numbers to find the gcd.

42

12

GCD of given numbers are :

6

) Enter 2 numbers to find the gcd .

4510

20

GCD of given numbers are :

10

27/01/
2021

Program no: 17

col. 20

Aim :

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

Program:

```
list = []

print ("Enter the length of the list")

n = int(input())

print ("Enter", + n, "numbers one by one:")

for i in range (0, n):
    a = int(input())
    list.append(a)

print (list)

b = len(list)

c = []
for a in range (0, b):
    if list [a] % 2 != 0:
        c.append(list [a])
print (c) "is the list without even numbers"
```

Result: print(c)

The program has been executed and output was verified

Output:

.) Enter length of list : 3

Enter numbers one by one :

33

44

55

[33, 44, 55]

List without even numbers.

[33, 55]

.) Enter length of list : 5

Enter numbers one by one :

2 3 7 8 4

[2, 3, 7, 8, 4]

List without even numbers

[3, 7,]

3/2/
2021

Program no : C02 1 , 18

AIM :

Program to find the factorial of a number.

PROGRAM:

```
def fact(n):  
    if (n == 1) or (n == 0):  
        return n  
    else:  
        return n * fact(n - 1)
```

```
printf("Enter a positive number : ")  
number = int(input())  
print("Factorial of ", +number, " is : ")  
print(fact(number))
```

RESULT :

The program has been executed and the output was verified.

output : c02 1

) Enter a positive number : 7

Factorial of 7 is : 5040

) Enter a positive number : 10

Factorial of 10 is : 3628800

) Enter a positive number : 0

Factorial of 0 is : 0

312 /

Program No : 19 CO2 - 2

202 /

Aim : Generate Fibonacci series of N terms

Program :

```
print ("Fibonacci series up to entered number")
```

```
n = int (input ("Enter how many terms: "))
```

```
print ("First ", +n, " Fibonacci numbers are : ")
```

```
a = 0
```

```
b = 1
```

```
count = 0
```

```
while count < n:
```

```
    print (a)
```

```
    c = a + b
```

```
    a = b
```

```
    b = c
```

```
    count += 1
```

Result :

The program has been executed and the output was verified.

Output :

-) Fibonacci series up to entered number

Enter how many terms : 7

First 7 fibonacci numbers are :

0, 1, 1, 2, 3, 5, 8

-) Fibonacci series up to entered number

Enter how many terms : 12

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89

Program no: 20 CO2.3

3/21
2021

Aim: Find the sum of all items in a list.

Program:

```
print ("Find the sum of all items in a list")
```

```
list = []
```

```
print ("Enter the size of list : ")
```

```
size = int (input ())
```

```
print ("Enter", + size, "elements one by one : ")
```

```
for i in range (0, size):
```

```
    a = int (input ())
```

```
    list.append (a)
```

```
total = 0
```

```
for i in list:
```

```
    total = total + i
```

```
print ("sum of numbers in list is : ", + total)
```

Result :

The program has been executed and output verified.

Output

) Find the sum of all items in a list

Enter the size of list : 5

Enter 5 numbers one by one :

6

88

3

4

23

Sum of numbers in list is : 124

) Find the sum of all items in a list

Enter the size of list : 7

Enter 7 numbers one by one :

88 9 4 5 6 2 8

Sum of numbers in list is : 122

Program no: 21 CO2.4

31/21
2021

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

Program :

print (List of four digit number with all numbers even and the number is a perfect square)

list = [] . list2 = []

print ("Enter the size of list :")

size = int (input ())

print ("Enter", +size, "4 digit numbers one by one")

for i in range (0, size):

a = int (input ())

if (a >= 1000 && a <= 9999) :

list.append (a)

else :

print (a, "is not a 4 digit number")

```
for i in list:  
    if i % 2 == 0:  
        j = 1  
        while (j * j <= i):  
            if ((i % j == 0) and (i / j == j)):  
                print(i), list2.append(i)  
            j = j + 1  
print(list2)
```

Result:

The program has been executed and the output was verified.

Output

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

.) Enter the size of list : 4

Enter 4 4digit numbers one by one :

1331 , 1225, 1156 , 1296

List with even numbers and perfect square 4 digit numbers:

[1156 , 1296]

.) Enter the size of list : 3

Enter 3 4digit numbers one by one :

1476 , 3025, 2116

List with even and perfect square 4 digit numbers are :

[2116]

3/2/
2021

Program no: 22

co2-S

Aim:

Display the given pyramid with step numbers accepted from user.

Eg: $N = 4$

```
1  
2 4  
3 6 9  
4 8 12 16
```

Program:

```
step = int(input('Enter the number : '))  
  
num = 1  
for i in range(1, b+1):  
    for j in range(1, i+1):  
        print(num * j, ', ', end="")  
  
    print("\n")  
    num = num + 1
```

Result:

The program has been executed and the output has been verified.

Output

Enter the step number for the pyramid : 4

1
2 4
3 6 9
4 8 12 16

Enter the step number for the pyramid : 7

1
2 4
3 6 9
4 8 12 16
5 10 15 20 25
6 12 18 24 30 36
7 14 21 28 35 42 49

3/2/
2021

Program no: 23

CO2.6

Aim:

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

Program:

```
printf ("Character frequency in given string!\n")
string = str (input ("Enter a string: "))

d = {}

for i in string:
    if i in d:
        d[i] += 1
    else:
        d[i] = 1

print ("Count of all characters in the entered string: \n")
print (d)
```

Result:

The program has been executed and the output has been verified.

Output :

) Character Frequency in given string :

Enter a string : hello how are you

Count of all characters in the entered string :

{ 'h': 2, 'e': 2, 'i': 2, 'o': 3, 'w': 1,
'a': 1, 'n': 1, 'y': 1, 'u': 1 }

) Character Frequency in given string :

Enter a string : umbrella

Count of all characters in the entered string:

{ 'u': 1, 'm': 1, 'b': 1, 'r': 1, 'e': 1,
'l': 2, 'a': 1 }

Program no: 24

CO2.7

3/2/

Aim:

21

Add 'ing' at the end of a given string -
if already ends with 'ing', then add 'ly'

Program:

```
a = str(input("Enter a string: "))

c = str("ing")
d = str("ly")

list = []

for i in a:
    list.append(i)

length = len(list)

if (list[length-1] == 'g') and (list[length-2] == 'n') and
   (list[length-3] == 'i'):

    b = a[0:] + d
    print(b)
else:
    b = a[0:] + c
    print(b)
```

Output

) Enter a string: play

ploying

) Enter a string : joking

jokingly

3/21
21

Aim :

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

Program:

list = []

size = int(input("Enter the number of words to be listed"))

print("Enter", size, "words one by one :")

for k in range(0, size):

string = str(input())

list.append(string)

print("List of words are : ", list)

list2 = []

for i in range(0, len(list)):

a = len(list[i])

list2.append(a)

c = max(list2)

```
for i in range (0, len (list 2)):
```

```
    if (list 2 [i] == c):
```

```
        break;
```

```
print ("Length of the longest word in list is:"  
      + c)
```

```
print ("The longest word in list is:", list[i])
```

Result :

Program has to been executed and the output was verified

Output

Program to accept a list of words and return length of the longest word

Enter the number of words in the list : 4

Enter 4 words one by one :

umbrella

purpose

glad

hello

List of words are : ['umbrella', 'purpose', 'glad',
'hello',]

Length of the longest word in list is : 8

Longest word in the list : umbrella

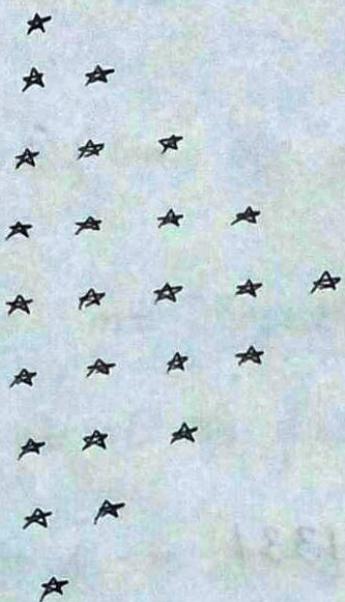
Program no: 26

CO2.9

3/21
21

Aim :

Construct following pattern using nested loop.



Program :

```
a = int(input("Enter the number : "))

for i in range(0, a):
    for j in range(0, i+1):
        print("*", end="")
    print("\n")

for i in range(a, 1, -1):
    for j in range(1, i):
        print("*", end=" ")
    print("\n")
```

Output

Enter the number : 5

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

Enter a number : 3

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

Program no: 27

CO2-10

3/2/
21

Aim:

Generate all factors of a number.

Program:

print (Generate all factors of a number)

n = int (input ("Enter a number : "))

print ("The factors of ", +n, "are: ")

for i in range (1, n+1):

if n % i == 0:

print (i)

Result:

The output was verified and program was

executed.

Output

Generate all factors of a given number :

Enter a number : 10

Factors of 10 are :

1
2
5
10

Enter a number : 1331

Factors of 1331 are :

1
11
121
1331

Program no: 28

CO2.11

3/2/
21

Aim :

write lambda function to find area of square, rectangle and triangle.

Program:

```
print ("Lambda function to find area of  
square, rectangle and triangle")
```

```
print (" Area of rectangle ")
```

```
print (" Enter the length and width of the  
rectangle : ")
```

```
l = int (input ())
```

```
w = int (input ())
```

```
x = lambda l, w: l * w
```

```
print ("Area of rectangle is: ", x (l, w))
```

```
print ("Area of square")
```

```
c = int (input ("Enter the length of square:"))
```

```
x = lambda c: c * c
```

```
print ("Area of square is : ", x(c))
```

```
print ("Area of triangle")
```

```
print ("Enter the breadth and height of  
the triangle : ")
```

```
b = int (input ())
```

```
h = int (input ())
```

```
x = lambda b, h : b * h / 2
```

```
print ("Area of triangle is : ", x(b, h))
```

Result

The program has been executed and
output was verified.

Output

write lambda functions to find area of square , rectangle and triangle

Area of rectangle

Enter the length and width of rectangle : 5 , 9

Area of rectangle is : 45

Area of square

Enter the length of square : 11

Area of square is : 121

Area of triangle

Enter the base and height of the triangle :

12, 24

Area of triangle is : 144