Page No. :

Date :

PYTHON ROUGH RECORD

1 Write a program in python to find the area of a circle.

Python Code:

```
import math
r = float(input("Enter the radius of the circle"))
area = math.pi * r * r
print("%0.2f" %area)
print("the area of circle is:" "%0.2f" %area)
```

Output:

Enter the radius of the circle: 5

the area of the circle is : 78.54

Page No :
Date :

2 Write a program in python to find the largest among the three numbers.

Python code:

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))
num3 = float(input("Enter third number: "))

if (num1 > num2) and (num1 > num3):
    largest = num1
elif (num2 > num3) and (num2 > num1):
    largest = num2
else:
    largest = num3

Print("The largest number is", largest)
```

Output:

Enter first number : 5
Enter second number : 8
Enter third number : 3

The largest number is 8.0

3 Write a program in python to find the square of a number

Python Code:

```
digit = int(input("Enter an integer number:"))

square = digit**2

Print(f"square of {digit} is {square}")
```

Output:

Enter an integer number : 10
Square of 10 is 100.

16. Write a program in python to count the words in a sentence.

Python Code:

```
str1 = input("please enter your own string:")
total = 1

for i in range(len(str1)):
    if (str1[i] == ' ' or str1 == '\n' or str1 == '\t'):
        total = total + 1

Print("Total number of Words in this string =",
      total).
```

Output:

Please enter your own string:

My name is Vimal Thimsm

Total number of words in this string = 5

5 Write a program in python to check the length of a ~~string~~ list.

Python code:

```
list_a = ["Hello", 2, 15, "World", 34]

n = len(list_a)

Print("Number of elements in the list:", n)
```

Output :

Number of element in the list : 5

6 Write a program in python to chuck the sum of a list.

Python code:

```
list1 = [11, 5, 17, 18, 23]

total = sum(list1)

Print("Sum of all the elements in the given
      list is:", total)
```

Output:

Sum of elements in give list = 74

7 Write a program in python to replace character in a string.

Python Code:

```
str1 = input("Please enter a string:")
ch = input("Please enter the character to be
            replaced:")
newch = input("Please enter the new character:")

str2 = str1.replace(ch,newch)

print("\n Original string:", str1)
print("\n Modified string:", str2)
```

Output:

Please enter a string: Computer Programming

Please enter the character to be replaced: Programming

Please enter the new character: Language

Original string: Computer Programming

Modified string: Computer Language

8 Python program to find gcd of a number,

Python Code:

```
def compute_gcd(x,y):

    if x>y:
        smaller = y
    else:
        smaller = x.
    for i in range(1, smaller+1):
        if((x%.i==0) and (y%.i==0)):
            gcd = i
    return gcd

num1 = int(input("enter the first number:"))
num2 = int(input("enter the second number:"))

Print(" The GCD is," compute_gcd(num1,num2))
```

Output:

enter the first number: 5

enter the second number: 35.

The GCD is 5

Page No.
Date:

* Write a python program to find the factorial of a number.

Python code:

```
num = int( input ("enter the number to
        calculate the factorial!"))

factorial = 1

if num < 0:
    print ("Sorry, factorial does not exist for
    negative numbers")
elif num == 0:
    Print ("The factorial of 0 is 1")
else:
    for i in range (1, num+1):
        factorial = factorial * i
    Print ("The factorial of", num, "is", factorial)
```

Output:

enter the number to find factorial: 7

The factorial of 7 is 5040

enter the number to find factorial: -7

Sorry, factorial doesnot exist for negative numbers

Page No :
Date :

10 Write a program in python to find fibonacci series.

Python Code:

```
n = int(input(" How many terms ? "))

n1, n2 = 0, 1
count = 0

if n <= 0:
   print(" please enter a positive integer")
elif n == 1:
   print("Fibonacci sequence upto", n":")
   print(n1)
else:
   print("Fibonacci sequence:")
   while count < n:
   print(n1)
   nth = n1 + n2

   n1 = n2
   n2 = nth
   count += 1
```

Output:

How many terms ? -5

Please enter a positive integer

How many terms ? 5

Fibonacci sequence:

0
1
1
2
3

Page No:
Date:

// Write a python program to find the vowels in a string.

Python code:

```
def vowel_count(str):
    count = 0
    vowel = "aeiouAEIOU"
    for alphabet in vowel:
        count = count+1
    print("No. of vowels:", count)
str = str(input("Enter the string"))
vowel_count(str)
```

OR

```
string = input("Enter string:")
vowels = 0
for i in string:
    if(i=='a' or i=='e' or i=='i' or i=='o' or
    i=='u' or i=='A' or i=='E' or i=='O' or
    i=='U'):
    vowels = vowels + 1
print("Number of vowels are:")
print(vowels)
```

Output:

Enter string : Vimal

Number of vowels are : 2

12 Write a python program to count a letter 'o' in a list

Python code:

```
list1 = ['a', 'e', 'i', 'o', 'u', 'i', 'c']

count = list1.count('i')

Print("The count of i is:", count)

Count = list1.count('p')

Print("The count of p is:", count)

str1 = input("enter the string:")

count = 0

for i in str1:
    if i == 'a':
        count = count + 1

print("Count of a is:", count)
```

Output:

Enter the list: aaa a hai a

Count of a is: 6.

Page No:
Date:

3. Write a python program to chuk the common element in two ~~bin~~ sets

Python code.

```
def common_member(a, b):
    a_set = set(a)
    b_set = set(b)

    if (a_set & b_set):
        print(a_set & b_set)
    else:
        print("No Commm element")

a = input("enter first set")
b = input("enter second set")
common_member(a, b)
```

Output:

enter first set: 1, 5 6.

enter second set: 1 6 7

{'1', '6'}

14 Write a python program to merge two dictionaries.

Python code:

```
def merge(dict1, dict2):
    return (dict2.update(dict1))

dict1 = {'a': 10, 'b': 8}.
dict2 = {'d': 6, 'c': 4}.

print.(merge(dict1, dict2))

Print(dict2)
```

Output:

{'d': 6, 'c': 4, 'a': 10, 'b': 8}

15 Write a python program to ~~print~~ asend and dsend a dictionary.

Python Code:

```
import operator
d = {1:2, 3:4, 4:3, 2:1, 0:0}
print ('original dictionary:', d)
sorted_d = sorted (d.items(),
        key = operator.itemgetter(1))
Print ('Dictionary in asending order : ', sorted_d
Sorted_d = dict (sorted (d.items(), key =
operator.itemgetter(1), reverse=True))
Print ('Dictionary in dsending order :', sorted_d)
```

Output:

Original dictionary : {1:2, 3:4, 4:3, 2:1, 0:0}

Dictionary in ascending order:

[(0,0), (2,1), (1,2), (4,3), (3,4)]

Dictionary in descending order:

{3:4, 4:3, 1:2, 2:1, 0:0}

16 Write a python program to exchange the first and last letter in a string.

Python Code:

```
def swap_string(str):
    if len(str) <= 1:
        return str
    mid = str[1:len(str)-1]
    return str[len(str)-1] + mid + str[0]

Print(swap_string('Python'))
Print(swap_string('Hello'))
Print(swap_string('G'))
Print(swap string('Vimal')).
```

Output:

mythoP
oellH
G
limax.

Page No.:
Date:

17 Write a program in python to find square of a numbers in a list.

Python Code

```
list1 = [11, 12, 13, 14, 15]
for i in range(len(list1)):
    list1[i] = list1[i] ** 2
print(list1)
```

Output:

[121, 144, 169, 196, 225]

Page No :
Date :

18 Write a python program to print string functions.

Python Code:

```
str1 = input("enter a string:")

Print ("Upper case:", str1.upper())
Print ("Lower Case:", str1.lower())
Print ("Swap Case:", str1.swapcase())
Print ("islower:", str1.islower())
```

Output:

enter a string : My name is Vimal

Upper Case : MY NAME IS VIMAL
Lower Case : my name is vimal
Swap Case : mY NAME IS vIMAL
islower : False.

Write a pythn program to generate Fibnauci series of N-turns.

Pythn Code:

```
n = int (input ("How many terms?"))
n1 = n2 = 0, 1
Count = 0

if n<=0:
   Print("please enter a+ve integer")
elif n==1:
   print ("Fibonacci sequence upto:", n, ":")
   print(n1)
else:
      print ("Fibonacci sequence:")
      while count <n:
      print(n1)
      nth = n1 + n2
      n2 = nth
      Count += 1.
```

22 Aim: To find the sum of all items in a list.

Python Code:

```
a: [4,5,8,2,1,9]
b = Sum(a)
print("Sum of list elements is:", b)
```

Result: The program was successfully executed and the output has obtained.

Output:

Sum of list elements is : 29.

Page No :
Date :

23 Aim: Generate a list of 4-digit numbers in a given range with all their digits even and the number a perfect square.

Python Code:

```
for i in range(1000, 10000, 1):
    for j in range(32, 100, 1):
        if i == j*j:
            string = str(i)
            if int(string[0]) % 2 == 0 and int(string[1]) % 2 == 0 and int(string[2]) % 2 == 0 and int(string[3]) % 2 == 0:
                print(i)
```

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

Output:

4624
6084
6400
8464.

24 Aim: 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:

```
lines = int(input("Enter the no of lines:"))
i=1
j=1
while i<=lines:
    j=1
    while j<=i:
        temp = i*j
        print(temp, end='', flush=True)
        print(" ", end='', flush=True)
        j=j+1
    print(" ");
    i=i+1;
```

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

Output:

Enter the no of lines: 4

1

2 4

3 6 9

4 8 12 16

Page No :
Date :

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

Python Code:

```
string = input("enter the string :")
Count = 0
for i in range(0, len(string)):
    if (string[i] != ' '):
        count = count+1;
print(" Total no of characters in the
    string: " str(count));
```

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

Output:

Enter the string : Vimal Thomas

The total no of characters in the string : 12.

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

Python code:

```
string = input("Enter a string:")
if len(string) < 3:
    print(string)
elif string[-3:] == 'ing':
    print(string + 'ly')
else:
    print(string + 'ing')
```

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

Output:

Enter a string: Try

Trying

Enter a string: Trying

Tryingly

Page No :
Date :

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

Python Code:

```
def longestLength(a):
    maxl = len(a[0])
    temp = a[0]

    for i in a:
        if(len(i)> maxl ):

            maxl = len(i)
            temp = i

    print("The word with the lngest length is:", temp, "and the lngth is:", maxl)

a = ["one", "Two", "Three", "Four"].
longestLength(a)
```

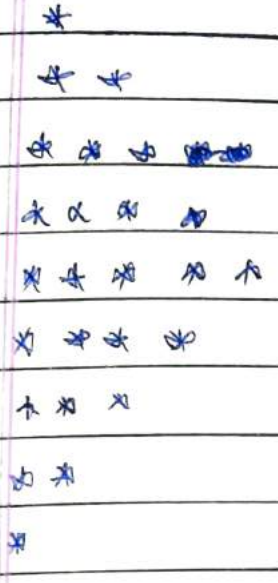
Result: The program was executed and the output has been verified.

Output:

The word with the longest length
is: third and the length is: 5

Page No:
Date:

28 Aim: Construct the following pattern using nested loop.



Python Code:

```
rows = int(input("Enter the no of rows: "))
for i in range(0, rows):
    for j in range(0, i+1):
        print("*", end='')
    print("\r")

for i in range(rows, 0, -1):
    for j in range(0, i-1):
        print("*", end='')
    print("\r")
```

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

Output:

Enter the no of rows : 5

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

Page No:
Date:

29 Aim: Generate all factors of a number.

Python Code:

```
N = int(input("Enter a number : "))
print (" The factors of ", format(N))
for i in range (1, N+1):
    if N % i == 0:
        print(i)
```

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

Output:

Enter a number: 100

The factors of 100 are,

1
2
4
5
10
20
25
50
100

Page No :
Date :

30 Aim: Write lambda functions to find area of square, rectangle and triangle.

Python Code:

```
l = int(input("length:"))
b = int(input("breadth:"))
h = int(input("height:"))

x = lambda a : a * a
print("Area of square is :", (x(l)))
y = lambda a, b : a * b
print("Area of rectangle is :", (y(l, b)))
z = lambda a, b : 1/2 * (a * b)
print("Area of triangle is :", (z(b, h)))
```

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

Output

length: 5
breadth = 10
height = 10.

Area of square is: 25
Area of rectangle is: 50
Area of Triangle is: 50.0

Page No.:
Date:

3. Aim: Work with build in packages

```
import datetime
x = datetime.datetime.now()
print(x)
x = datetime.datetime.now()
print(x.year)
print(x.strftime("%A"))
```

Result: The program has been executed and the outcome has been verified.

Output:

2021-03-28 04:13:33.347291

2021

Sunday:

Page No:
Date:

32 Aim: Create a package named graphics with modules rectangle, circle and sub package 3D-graphics with module cuboid and sphere. Include methods to find area and perimeter of different respective figures in each module. Write program that finds area and perimeter of figures by different importing statements.

Python code:

```
from graphics.rectangle import *
from graphics.circle import *
from graphics.3dgraphics.sphere import *
from graphics.3dgraphics.cuboid import *

print("Rectangle")
l = int(input("enter length:"))
b = int(input("enter breadth:"))
print("Area of Rectangle:", rarea(l,b))
Print("Perimeter of Rectangle:" perimeter(l,b))
print("Circle:")
r=int(input("enter a radius:"))
print("area of circle:" (area(r))
print("circumference of circle:" (perimeter(r))
print("sphere")
print("area of sphere:"
r = int(input("enter a radius:"))
print("area of sphere:" sarea(r))
```

Page No :
Date :

```
print("perimeter of sphere :" sperimeter(r))
print("Cuboid")
l = int(input(" enter a length :"))
w = int(input(" enter a width :"))
h = int(input(" enter a hight :"))
print(" area of cuboid :", carea(l,w,h))
print(" perimeter of cuboid =", cperimeter(l,w,h))

# circle
from math import pi
def carea(r):
    return (pi * r * r)
def cperimeter(r):
    return (2* pi* r)

# rectangle
def rarea(l,b):
    return (l*b)
def rperimeter(l,b):
    return (2*(l+b))

# Sphere
from math import pi
def sarea(r):
    return (4* pi * r* r)
def sperimeter(r):
    return ((4/3) * pi * r * r * r);

# Cuboid
def carea(l,w,h)
    return (( 2*l*w) + (2* l * h) + ( 2* w*h))
```

```
def cuperimeter(l,w,h)
    return(4 * (l+w+h))
```

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

Output:

Rectangle.
Enter length: 5
Enter Breadth: 10.
Area of rectangle: 50
Perimiter of rectangle: 30

Circle
Enter radius of circle: 4.
Area of circle: 50.265482457
Perimeter of circle: 25.132741228

Sphere
Enter radius of Sphere: 3
Area of sphere: 113.09733552923255
Perimeter of sphere: 113.09733552923255

Cuboid:
Enter length: 7
Enter width: 12.
Enter hight: 8.
Area of cuboid: 472.
Perimeter of cuboid: 108.

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

Python Code:

```
class Rectangle:
    def __init__(self, length, breadth):
        self.length = length
        self.breadth = breadth

    def area(self):
        return self.length * self.breadth
    def peri(self):
        return 2*(self.breadth + self.length)

a = int(input("Enter length of 1st rectangle:"))
b = int(input("Enter breadth of 1st rectangle:"))
c = int(input("Enter length of 2nd rectangle:"))
d = int(input("Enter breadth of 2nd rectangle:"))

obj = Rectangle(a,b)
obj1 = Rectangle(c,d)

print("area of 1st rectangle:", obj.area())
print("area of 2nd rectangle:", obj1.area())
print("Perimeter of 1st rectangle:" obj.peri())
print("Perimeter of 2nd rectangle:" obj1.peri())
```

```
if obj.area() == obj1.area():
    print("Equal")
else:
    print("Not Equal")
```

Result: The program was executed and the result was verified.

Output:

Enter length of 1st rectangle : 5

Enter breadth of 1st rectangle : 10

Enter length of 2nd rectangle : 7

Enter breadth of 2nd rectangle : 10.

Area of first rectangle : 50.

Area of 2nd rectangle : 70.

Perimeter of 1st rectangle is : 30.
Perimeter of 2nd rectangle is : 34.

Not Equal.

34 Aim: Create a Bank account with numbers account, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.

Python Code:

```
Class Bank_Account:
    def __init__(self):
        self.balance = 0

    def deposit(self):
        amount = float(input("Enter amount to be deposited:"))

        self.balance += Amount
        print("\n Amount deposited:", amount)

    def withdraw(self):
        amount = float(input(" Enter amount to be withdrawn:"))

        if self.balance >= amount:
            self.balance -= amount
            print("\n You withdrew:", amount)
        else:
            print("\n insufficient balance")
```

```
def display(self):
    print("\n. Net available Balance:",
          self.balance)

s = BankAccount()
s.deposit()
s.withdraw()
s.display()
```

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

Output:

Enter amount to be deposited: 900

Amount Deposited: 900.0

Enter amount to be Withdrawn: 400

You withdrew: 400.0

Net available balance: 500.0

35 Aim: Create a class rectangle with private attributes length and width. Overload '<' to compare the area of two rectangles.

Python Code:

```
class A:
    __length = 0
    __width = 0
    __area = 0
    def __init__(self, l, w):
        self.__length = l
        self.__width = w
    def area(self):
        self.__area = self.__length * self.__width
    def __gt__(self, other):
        if (self.__area > other.area):
            return True
        else:
            return False

rect1 = A(3,4)
rect1.area()
rect2 = A(6,5)
rect2.area()
```

```
if (rect1 > rect2):
    print("rect1 is greater than rect2")
else:
    print("rect2 is greater than rect1")
```

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

Output:

num2 is greater than num1

36 Aim: Create a class time with private attributes hour, minute, second. Overload '+' operator to find the sum of times.

Python code:

```
class Time:
    def __init__(self, h, m, s):
        self.__hour = h
        self.__minute = m
        self.__seconds = s
    def time(self):
        if self.__seconds >= 60:
            self.__seconds -= 60
            self.__minute += 1
        if self.__minute >= 60:
            self.__minute -= 60
            self.__hour += 1

    def __add__(self, other):
        self.__hour = self.__hour + other.__hour
        self.__minute = self.__minute + other.__minute
        self.__seconds = self.__seconds + other.__seconds
        return (self.__hour, self.__minute,
                self.__seconds)
```

```
obj1 = Time (3,25,45)
obj1.time()
obj2 = Time(6,30,5)
obj2.time()
print("Sum of two time:")
print(obj1+obj2)
```

Result: The program was executed and the output was obtained.

Output:

Sum of two time:

(9, 55, 50)

37 Aim: Creat a class publisher. Derive class book from publisher with attributes title and author. Derive class Python from Book attributes price and no_of_pages. Write a program that display information about a python book. Use base class constructor invocation and method overriding

Python Code

```
class Publisher:
    def __init__(self, pubname):
        self.pubname = pubname
    def display(self):
        print("Publisher Name:", self.pubname)

class Book(Publisher):
    def __init__(self, pubname, title, author):
        Publisher.__init__(self, pubname)
        self.title = title
        self.author = author

    def display(self):
        print("Title", self.title)
        print("Author:", self.author)

class Python(Book):
    def __init__(self, pubname, author, price,
                 no_of_pages):
```

Page No.
Date:

```
Book.__init__(self, pubname, title, author)
    self.price = price
    self.no_of_pages = no_of_pages

def display(self):
    print("Title:", self.title)
    print("Author:", self.author)
    print("Price:", self.price)
    print("Number of pages:", self.no_of_pages)

S1 = Python("ak books", "Learning Python By Programming",
            "Jeeva Jose", 200, 219)

S1.display()
```

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

Output:

Title : Taming Python by Programming

Author : Jeeva Jose

Price : 200.

Number of pages : 219

38 Aim: Write a python program to read a file line by line and store it into a list

Python Code:

```
str1 = "Welcome to Python programming" "\n" "python"
                                        "\n"
fw = open("Afile.txt", "w")
fw.write(str1)
fw.close()

fr = open("Afile.txt", "r")
str2 = fr.readlines()
for i in str2:
    print(i)
```

Result: The program was executed and has been successfully verified.

Output:

Welcome to Pythm Programming

Pythm

Aim: Python program to copy odd lines of one file to another

Python code:

```
fo = open("file1.txt",'r')
str1 = fo.readlines()
fo.close()
fo = open("file2.txt",'w')
n = 0;
for i in str1:
n = n+1
if n%2!=0:
    fo.write(i)
fo.close()
fo = open("file2.txt",'r')
str2 = fo.readlines()
ptr(str2)
```

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

Output :

['haiiiiiiiiii']

40 Aim: Write a python program to read each row from a given csv file and print a list of strings.

Python code:

```
import csv
with open('movie1.csv','w',newline='')
as file:
writer = csv.writer(file)
writer.writerow(["SN", 'Movie', "Rating"])
writer.writerow([1, "Lord of Rings", 5])
writer.writerow([2, "Harry Potter", 6])
with open('movie1.csv') as csvfile:
    data = csv.reader(csvfile)
for row in data:
    print(','.join(row))
```

Result: The above program was executed and the result was obtained.

Output:

SN, Movie, Rating
1, Lord of the rings, 5.
2, Harry Potter, 6.

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

Python Code.

```
import csv
with open("fruit1.csv", 'w', newline="") as file:
    write = csv.writer(file)
    write.writerow(["Sl.No", "fruits", "rate"])
    write.writerow(["1", "apple", "60"])
    write.writerow(["2", "orange", "55"])
    write.writerow(["3", "grapes", "70"])
    write.writerow(["4", "banana", "85"])

with open("fruit1.csv", 'r') as file:
    data = csv.reader(file)
print("Contents in column 'fruits':")
    print(r[1])
for r in data:
```

Result: The above program was successfully executed and the result verified.

Output:

Contents in the column 'fruits':

fruits
apple
Orange
grapes
banana

Page No :
Date :

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

Python Code:

```
import csv
f = open("fruits.csv","w")
writer = csv.DictWriter(f,fieldnames=["fruit","count"])
writer.writeheader()
writer.writerow({"fruit":"Apple","count":"1"})
writer.writerow({"fruit":"Banana","count":"2"})
f.close()
c = 0
f = open("fruits.csv")
reader = csv.DictReader(f)
for row in reader:
    if c == 0:
        print(f'{",".join(row)}')
    print(f'{row["fruit"]},{row["count"]}')
f.close()
```

Result: The program was successfully executed and the result was verified.

Output:

fruit : count

Apple , 1

fruit count

Banana , 2