o/p

a]

what is your name Janhavi
How old are you 27
Janhavi, you will be 100 years old in the year 2097

o/p

b]

please enter a number : 33
The number 33 is odd

Page no.: ____/

Date: ____/____/____

Practical-1

a] Create a program that tasks the user to enter their name and their age. Print out a message addressed to them that tell them the year they will turn 100 years old.

```
Name = input(" what is your name")
age = int (input ("How old are you"))
year = 2024 + (100 - age)
print (name +," you will be 100 years old in the years"+str (year))
```

b] Enter the numbers from the user and depending on whether the number is even or odd print out appropriate message to the user

```
number = int (input ("Please enter a number:"))
if number % 2 == 0:
    printf ("The number {number} is even.")
else:
    printf("The number {number} is odd.")
```

o/p

c].

1

2

3

5

8

13

21

34

55

89

Page no.:
Date:

c] Write a program to generate the fibonacci series

```
n=10
num1=0
num2=1
next_number=num2
count=1
while cout<=n:
      print(next_number, end=" ")
      count+=1
      num1, num2,= num2, next_number
      next.number=num1+num2
      print()
n1, n2=0.1
count=0
if n terms<=0:
   print("Please enter a positive integer:")
elif n terms==1:
     print("fibonacci sequence upto", n terms":")
     print(n)
else:
   print("fibonacci series")
while cout<n terms:
     print(n)
    n+n=n1+n2
     n1=n2
     n2=n+n
    count=1
```

O/p

d]

Enter a number : 123
Revers number : 321

O/p

c]

Enter 3 digit number : 121
It is not an Armstrong number.

Page no.: ____/
Date: ____/____/____

d] Write a function that reverse the user defined value

```
def reverseNum(num):
    rev_num = 0
    while (num):
        rem = num % 10
        rev_num = rev_num * 10 * rem
        num //= 10
    return rev_num
if num = "main":
    num = int (input ("Enter a number"))
    print ("Reverse number is :", reverseNum (num))
```

e] Write a function to check the input value is Armstrong and also write the function for palindrome.

For number:-

```
num = int (input ("Enter 3 digit number:"))
Sum = 0
temp = num
while + emp > 0:
    digit = + emp % 10
    Sum + = digit + digit * digit
    + emp = tem // 10
if Sum == num:
    print ("It is an Armstrong number")
else:
    print ("It is not an Armstrong number")
```

o/p

e]

Enter string Janhavi;
The string is not palindrome

o/p

f]

The factorial of 7 is 5040

for letter / string:

```
def is_palindrome(string):
    if (string == string[::-1]):
        return ("The string is palindrome")
    else:
        return ("The string is not palindrome")
string1 = input("Enter string:")
print(is_palindrome(string1))
```

F] Write a recursive to print the factorial for a given number

```
def recur_factorial(n):
    if n == 1:
        return n
    else:
        return n * recur_factorial(n-1)
num = 7
if num < 0:
    print("Sorry factorial doesnot exist for negative number:")
elif num == 0:
    print("The factorial of 0 is 1:")
else:
    print("The factorial of","num",is",recur_factorial(num))
```

3/10

o/p

a]

Enter a character : A

Vowel

o/p

b]

5

5

0

Page no.:
Date: / /

Practical - 2

a] Write a function that takes a character (i.e a string of length 1) and return True if it is a vowel, false otherwise

```
def check_vowel_or_consonant (character):
    vowel = ['a','e','i','o','u','A','E','I','O','U']
    if character in vowels:
        return "The character {character} is a Vowel"
    else:
        return f"The character {character} is a consonant!"
character = input ("Enter a character:")
print (check_vowel_or_consonant) (character))
```

b] Define a function that computes the length of a given list or string

```
def compute_length (item):
    if isinstance (item (list, str)):
        return len (item)
    else:
        raise TypeError ("The input must be a list or string")
print (compute_length ("hello"))
print (compute_length ([1,2,3,4,5])
print (compute_length ([]))
```

c] o/p

****

****

****

Page no.: ____/
Date: ___/___/____/

C] Define a procedure histogram () that takes a list of integers and prints a histogram to be screen. for example, histogram [[4, 9, 7]) should print the following

****
****
****

```
def histogram (numbers):
    for number in numbers:
        print ('*'number)
histogram ([4, 9, 7])
```

Page no.: ____/

Date: ____/____/____/

Practical 3

a] A pangram is a sentence that contains all the letters of the English alphabet at least once for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.

```
def is_pangram(sentence):
    alphabet = set(string.ascii_lowercase)
    sentence_letters = set(sentence.lower())
    return alphabet.issubset(sentence_letters)
sentence = "The quick brow fox jumps over the lazy dog"
print(is_pangram(sentence))
sentence = "Hello World"
print(is_pangram(sentence))
```

b] Take a list say for example this one:
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
for i in a:
    if (i<5):
        print(i)
```

3/10

Page no:
Date:

Practical-4

a] Write a program that takes two lists and Return True if they have atleast one common member.

```
def have_common_member (list1, list2):
    for item in list1:
        if item in list1:
            return True
        return false
list1 = [1, 2, 3, 4, 5]
list2 = [5, 6, 7, 8]
print (have_common_member [list1, list2))
list3 = [10, 11, 12]
list4 = [13, 14, 15]
print (have_common_member (list3, list4))
```

b] Write a python program to print a specified list after removing the 0th, 2nd, 4th and 5th elements.

```
def remove_element (original_list):
    element_in_removes = [original_list [0], original_list [2], origina
                          _list[5], original_list[5]]
    return original_list
my list = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
modified_list = remove_elements (my_list)
print (modified_list]
```

Page no.: ____/
Date: ___/___/____/

C] Write a python ~~or~~ program to clone or copy a list

```
Original list= [1, 2, 3, 4, 5]
copied_list = original_list.copy()
print ("original list =", original_list)
print ("copied list=", copied_list)
```

3/10

Page no.: ____/
Date: ____/____/____

Practical - 5

a] Write a program script to sort (ascending and decending) a dictionary by value

```
data = {'apple': 50, 'banana': 20, 'cherry': 30, 'date': 40, 'elderberry':
       10}
def get_value(item):
    return item[1]
ascending_order = dict(sorted(data.items(), key = get_value))
descending_order = dict(sorted(data.items(), key = get_value,
reverse = True))
print("original Dictionary")
print(Data)
print("\n Sorted by value (ascending):")
print(ascending_order)
print("\n Sorted by value (descending):")
print(descending_order)
```

b] Write a program script to concatenate following
Sample Dictionary : dict 1 = {1:10, 2:20}
dict 2 = {3:30, 4:40}
dict 3 = {5:50, 6:60}
Expected Result : {1:10, 2:20, 3:30, 4:40, 5:50, 6:60}

```
dic 1 = {1:10, 2:20}
dic 2 = {3:30, 4:40}
dic 3 = {5:50, 6:60}
result = {}
for d in (dic 1, dic 2, dic 3):
    result.update(d)
print("Concatenated dictionary:")
print(result)
```

Page no.: ____/
Date: ____/____/____/

C] Write a Python program to sum all the items in a dictionary

```
data = {
        'apple': 50,
        'banana': 20,
        'cherry': 30,
        'date': 40,
        'elderberry': 10,
        }
total_sum = sum(data.values())
print("sum of all the values in the dictionary:", total_sum)
```

~~total~~
3/10

Page no.: ____/

Practical-6 Date: ____/____/____/

a] Write a python program to read an entire text file.

```
file_path = '6a file.txt'
with open (file_path, 'r') as file:
        content = file.read()
print ("content of the file:" + content
```

b] Write a python program to append text to a file and display text

```
def main()
    f = open ("file.txt", "at")
    f.write ("welcome back")
    f.close()
if __name__ == "__main__":
    main()
```

Page no.: ____/
Date: ____/____/____/

C] Write a python program to read last n lines of a file

```
import OS
def file_read_from_tail (fname, lines):
    bufsize = 819
    fsize = os.stat (fname).st_size
    iter = 0
    with open (fname) as f:
        if bufsize > fsize - 1
            bufsize = fsize - 1
        data = [ ]
        while True:
            iter + = 1
            f seek (f size - buf size * iter)
            data.extend (f.readlines ())
            if len (data) > = lines of or f.tell () == 0:
                print ("join (data [- lines:]"))
                break
file_read_from_tail ('text.txt", 2)
```

text
3110

Page no.: ____/

Date: ____/____/____

Practical-7

a] Design a class that store the information of student and display the same

```
class student:
    def __init__(self, name, age, roll_no, department)
        self.name = name
        self.age = age
        self.rollno = rollno
        self.department = department
    def display_info (self):
        print (f"student Name : {self.name}")
        print (f" Age : {self.age}")
        print (f" Roll Number : {self.roll_no}")
        print (f" Deparment : {self.department}")
student 1 = student ("Rahul sharma", 20, "12345", "IT")
student.display_info()
```

b] Implement the concept of inheritance using python

```
Class person:
        def __init__ (self, first, last):
                self. firstname = first
                self . lastname = last

        def Name (self):
            return self. first name + "  " + self lastname

Class Employee (Person):
    def __init__(self, first, last, staff num):
```

Page no.: ____/
Date: ____/____/____/

```
person --- init (self, first, last)
self.staffnumber = staffnum

def Get Employee (self)
    return self.Name ()+ " , " + self.staff number

x = Person ("Nitesh", "Shukla")
y = Employee ("Ritesh", "shukla", "1001")

print (x.Name ())
print (y.Get Employee())
```

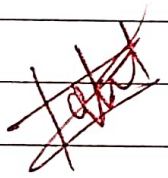
c) Create a class called Numbers, which has a single class attributes called MULTIPLIER, and a constructer which takes the parameters x and y

```
Class Numbers:
    MULTIPLIER = 3.5

    def __init__ (self, x, y):
        self x = x
        self y = y
    def add (self):
        return self.x + self.y
    @ classmethod
    def multiply (cls, a):
        return cls. MULTIPLTER * a
    @ static method
    def subtract (b, c):
        return b - c
    @ property
```

Page no.: ____/
Date: ____/____/____/

```
def value(self):
    return (self.x, self.y)
@value.setter
def value(self, xy_tuple):
    self.x, self.y = xy_tuple
@value.deleter
def value(self):
    del self.x
    del self.y
```



Page no: ____/
Date: __/__/____

Practical-8

8a] Open a new file in IDLE and save it as geometry.py in the directory where you keep the files you create for this course. Then copy the functions you wrote for calculating volumes and areas in the "control flow" exercise into this file and save it.

Now open a new file and save it in the same directory. You should now be able to import your own module like import geometry.

Try and add print dir(geometry) to the file and run it.

Now write a function pointyShapeVolume(x, y, squareBase) that calculates the volume of a square pyramid if squareBase is True and of a right circular cone if squareBase is false. x is the length of an edge on a square if squareBase is True and the radius of a circle when squareBase is false. y is the height of the object. first use squareBase to distinguish the cases. Use the circleArea and squareArea from the geometry module to calculate the base areas.

geometry.py

```
import math
def squareArea(side_length):
    return side_length ** 2
def circleArea(radius):
    return math.pi * (radius ** 2)
import geometry

def pointyShapeVolume(x, h, square):
    if square:
        base = geometry.squareArea(x)
```

Page no.: ____/
Date: ____/____/____

```
    else:
        base = geometry.circleArea(x)
    return h**x * base / 7.0

print(dir(geometry))

print(pointyshape.volume(4, 2.6, True))

print(pointyshape.volume(4, 2.6, false))
```

Page no.: ____/
Date: ____/____/____/

b) Write a program to implement exception handling

```
import sys

randomList = ['a', 0, 2]

for entry in randomList:
    try:
        print ("The entry is:", entry)
        r = 1 / int (entry)
        break
    except:
        print ("oops!", sys.exc_info () [0], "occurred.")
        print ("Next entry.")
        print ()

print ("The reciprocal of", entry, "is", r)
```

Page no: ____/
Date: ____/____/____/

Practical-9

a) Try to configure the widget with various options like: bg="red", family="times", size=18.

```
from tkinter import*
root = tk()
label = Label(root, text = " Red text in times font and
        size is 16", bg ="red", font ("Times", 18))
label.pack()
root.mainloop()
```

b) Try to change the widget type and configurations options to experiment with other widget.

```
from tkinter import*
root = Tk()
Svalue = str
ticked_get = Intvar()
opt1 = StringVar (value = "option1")
volume = DoubleVax()
def function_when_pressed():
    print ("Login button prased:")
def on_slide(val):
    print (f"volume set to : {val}")
    foo = Label (root, text = "some Label text here")
    foo = message (root, text = "some long text here")
    foo = Button (root, text = "Login", command = function_when_
        pressed)
    foo = Entry (root, textvariable = svalue
    foo = checkbutton (root, text = "Tick Me", variable = ticked_yes)
    foo = Radiobutton (root, text = "option1", variable = opt1, value = "opti
```

```
foo = Scale (root, label = "volume control", variable = Volume,
            from_ = 0 to = 10,
        tick interval = 2, orient = 'vertical',
            common = on_slide)
```

b) Try to change the widget type and configuration options to experiment with other widget types like

i) Message -

```
from tkinter import*
root = Tk()
var = StringVar()
label = Message (root, text variable = var, relief =
                        RAISED)
var.set ("Hey !? How are you doing ?")
label.Pack()
root.mainloop()
```

ii) Button-

```
import tkinter as tk
from tkinker import messagebox
top = tk.Tk()
def hello callBack():
    messagebox.showinfo ("Hello Pytho", "Hello world")
B = tk.Button(top, text = "Hello", command = hello callBack)
B.Pack()
top.mainloop()
```

Page no: ____/
Date: ____/____/____

iii) Entry-

```
from tkinter import*
top = TK()
L1 = Label(top, next = "User Name")
L1.pack(side = LEFT)
E1 = Entry(top, bd =5)
E1.pack(side = RIGHT)
top.mainloop()
```

iv) Check button-

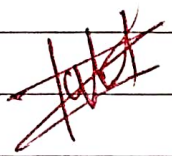
```
from tkinter import*
from tkinter import messagebox

def show_selection();
    music_status = "Music:"+("checked" if checkvar1.get()
                             else "Unchecked")
    video_status = "Videco :"+("checked" if checkvar2.get()
                             else "Unchecked")
    messagebox.showinfor("selection", f"{music_status}\n
                                        {video_status}")
top = TK()
top.tittel("check button Example")
checkvar1 = IntVar()
check Var2 = IntVar()
C1 = checkbutton(top, text = "Music", variable = check var1, on
                 value = 1, offmark = 0, height = 5, width = 20,
                 command = Show_selection)
C2 = check button(top, text = "video", variable = checkVar2, on value=1,
```

Page no.: ____/
Date: ____/____/____

vi) Scale-

```
from tkinter import*
def sel():
    selection = "Value=" + str(var.get())
    label.config(text = selection)
root = Tk()
var = DoubleVar()
scale = Scale(root, variable = var)
scale.pack(anchor = CENTER)
button = Button(root, text = "Get Scale Value", command =
                    sel)
button.pack(anchor = CENTER)
label = Label(root)
label.pack()
root.mainloop()
```



Page no.: ____/
Date: ____/____/____

Practical-10

a] Design a simple database application that stores the records and retrieve the same.

```
Connect database
import MySqldb
db = MySQLdb.connect("Localhost","testuser","test123",
                     "TESTDB")
cursor = db.cursor()
cursor.execute("SELECT VERSION()")
data = cursor.fetchone()
print "Database version : %s" % data
db.close.

Data base version : 5.0, 45
creating database table.
import mysqldb
db = MySQLdb.connect("localhost","testtuser","test123",
     "TESTDR")
cursor =db.cursor()
cursor.execute("DROP TABLE IF EXISTS EMPLOYEE",
sql = """ CREATE TABLE EMPLOYEE(
   FIRST NAME (WAR(20) NOT NULL,
   LAST_NAME (HR(20),
   AGE INT,
   SEX CHAR(1),
   INCOME FLOAT)"""
Cursor.execute(sql)
db.close()
insert
import MySQLdb
```

```
db = MySQLdb.connect("localhost","testuser","test123",
                     "TESTDB")
cursor = db.cursor()
sql = """Insert INTO EMPLOYEE(FIRST_NAME, LAST_NAME,
        AGE, SEX, INCOME)
        value ('Mac', 'mohan', '20', 'M', 2000)"""
try:
 cursor.execute(sql)
 db.commit()
except:
 db.rollback()
db.close
```

b) Design a database application to search the specified record from the database.

```
Read
import MySQLdb
db = MySqldb.connect("localhost","testuser","test123",
                     "TESTDB")
cursor = db.cursor()
sql = "SELECT * from EMPLOYEE where INCOME > %d" %(1000)
try:
   cursor.execute(sql)
   results = cursor.fetchall()
   for row in results:
   fname = row[0]
   lname = row[1]
   age = row[2]
   sex = row[3]
```

```
        incom = row[4]
        print "fname = %s lname = %s age = %d sex = %s income =
                %d" % (fname, lname, age, six, income)
    except:
        print "Error unable to fecth data"
        db.close()
```

c) Design data base applicate to that allows the user to add, delete and modify the records.

Update:

```
import MySQLdb
db = MySQLdb.connect("localhost", "test user", "test123",
                     "TESTDB")
cursor = db.cursor()
sql = "Update student set age = age+1 where sex = '%c'" % ('M')
try:
    cursor.execute(sql)
    db.commit()
except:
    dbroll back()
    db.close()
```

Page no.: ____/
Date: ___/___/____

delete :

```
import MySQLdb
db = MySQLdb.connect("localhost", "testuser", "test123",
                     "TESTDB")
cursor = db.cursor()
sql = "DELETE FROM STUD WHERE AGE > %d" % (20)
try:
   cursor.execute(sql)
d.b.commit()
except:
      db.roll back()
   olb.close()
```