Roll No:-\_\_\_\_\_\_\_\_\_\_

**Sonopant Dandekar Shikshan Mandali’s**

**Sonopant Dandekar Arts, V.S. Apte Commerce,**

**M.H. Mehta Science College**



**DEPARTMENT OF COMPUTER SCIENCE**

**CERTIFICATE**

Certified that Mr./Miss.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ has satisfactorily completed a course of necessary experiment in\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Under my supervision in the FY BSc. Computer Science in the Year 20 -20

**Head of Department Subject teacher**

Date: / /20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR.  No. | Title of Exercise | Practical No. Date | Submission Date | Remarks |
| 1 | Write a Python Program to implement various file operation: | 01/04/2022 | 04/04/2022 |  |
| 2 | Write a Python program to demonstrate use of regular expression for suitable application | 01/04/2022 | 04/04/2022 |  |
| 3 | Write a python program to demonstrate concept of threading and multitasking in python. | 04/04/2022 | 05/04/2022 |  |
| 4 | Write a python program to work with database in python to perform operations such as  a) Connecting to databases  b) Creating and dropping tables.  c) Inserting and updating into tables | 04/04/2022 | 05/04/2022 |  |
| 5 | Write a python program to demonstrate different types of exception handling. | 05/04/2022 | 07/04/2022 |  |
| 6 | Write a GUI program in python to design application that demonstrates  a) Different fonts and colours  b) Different Layout Managers  c) Event Handling | 05/04/2022 | 07/04/2022 |  |
| 7 | Write Python program to create application which uses date and time in python | 07/04/2022 | 08/04/2022 |  |
| 8 | Write a python program to implement concepts of OOP such as  a) Types of Methods  b) Inheritance  c) Polymorphism | 08/04/2022 | 14/04/2022 |  |

**Practical No. 1:**

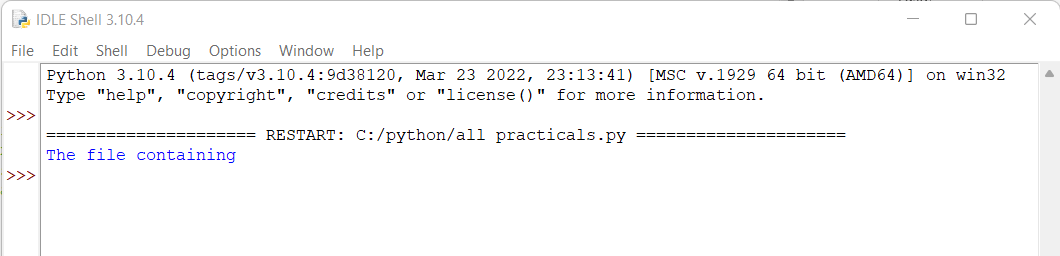
* **Write a Python Program to implement various file operation:**

1. read file fo=open("f2.txt",'r') c=fo.read()

print("The file containing ",c)

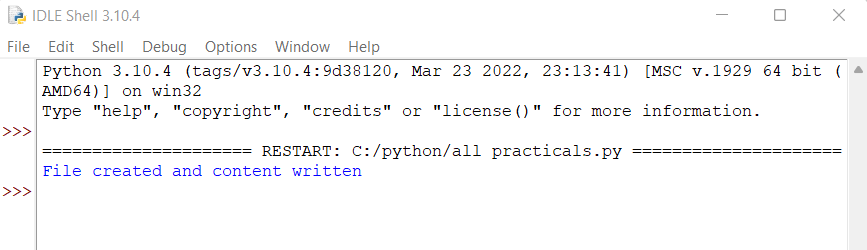
fo.close()

**OUTPUT**

****

1. create file fo=open("fycs1.txt","w") fo.write("Writers write, but fingers don't fing.") print ("File created and content written") fo.close()

**OUTPUT**

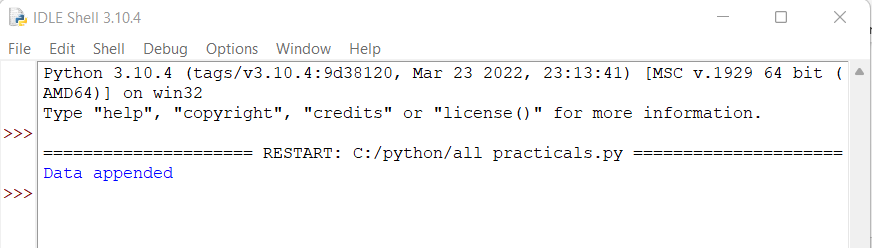
****

1. read & write fo=open("f2.txt","a+")

fo.write("This is file handeling program")

print("Data appended")

**OUTPUT**

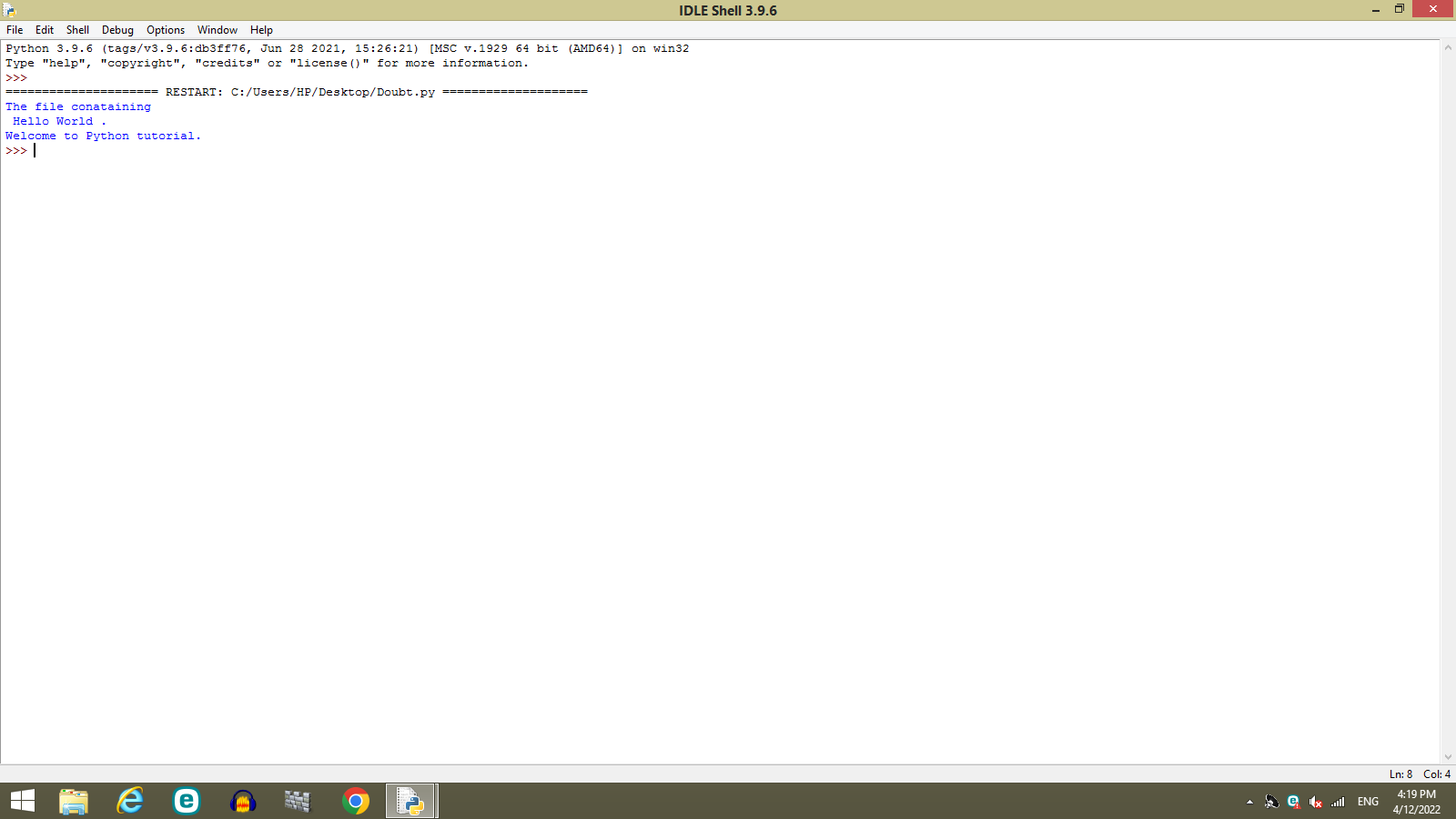
****

1. fo=open("f2.txt",'r') c=fo.readlines()

print(c)

fo.close()

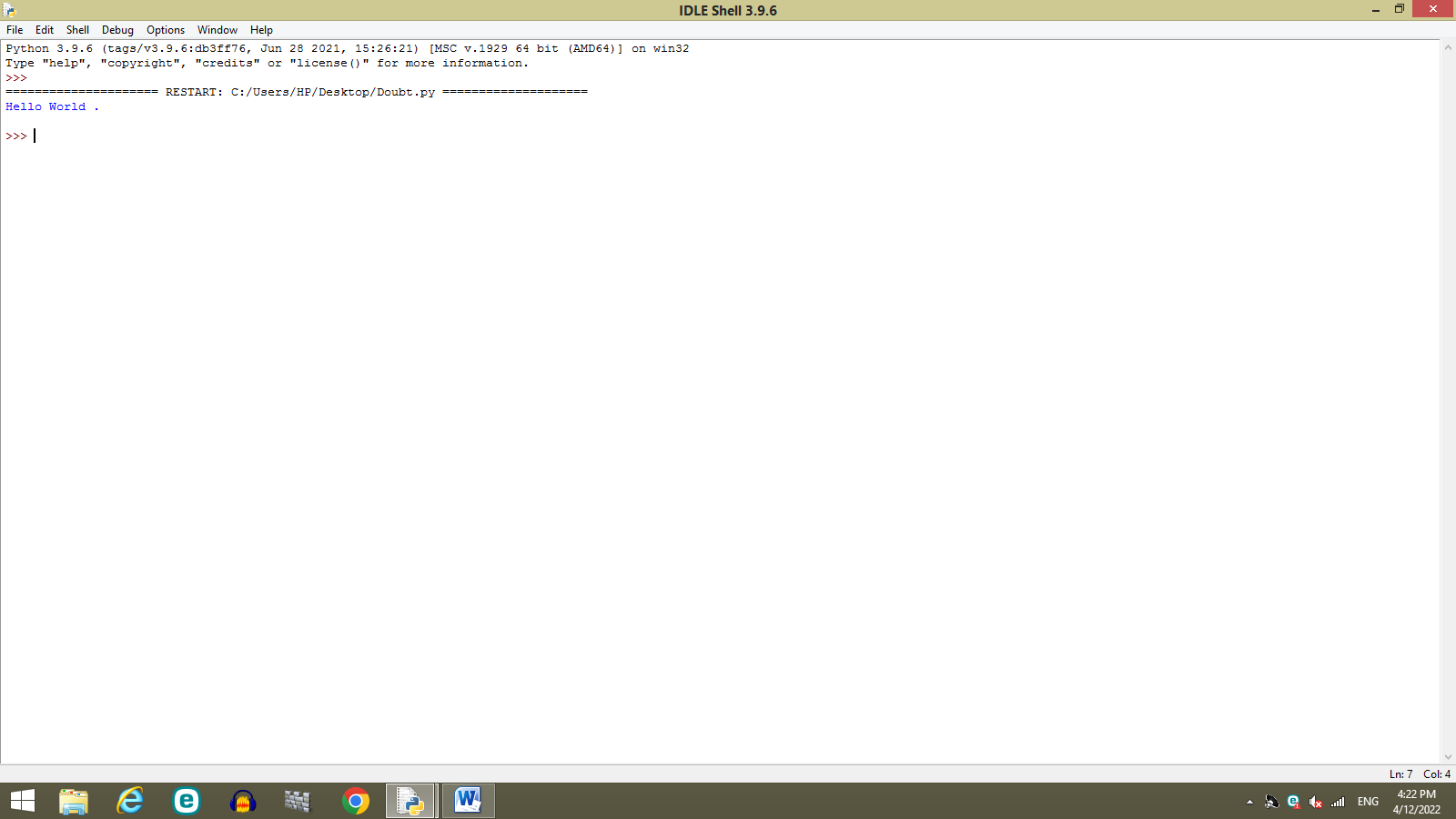
**OUTPUT**



1. fo=open("f2.txt",'r') c=fo.readlines() f=c[len(c)-2] print(f)

fo.close()

**OUTPUT**



**Practical No. 2:**

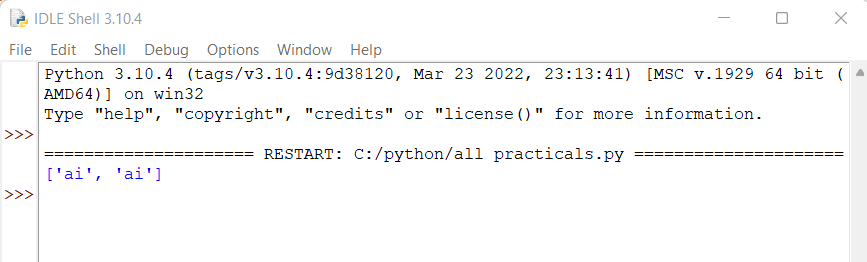
 **Write a Python Program to demonstrate use of regular expression for suitable application:**

1. import re

txt=”The rain in spain”

x=re.findall ("ai",txt) print(x)

**OUTPUT**

****

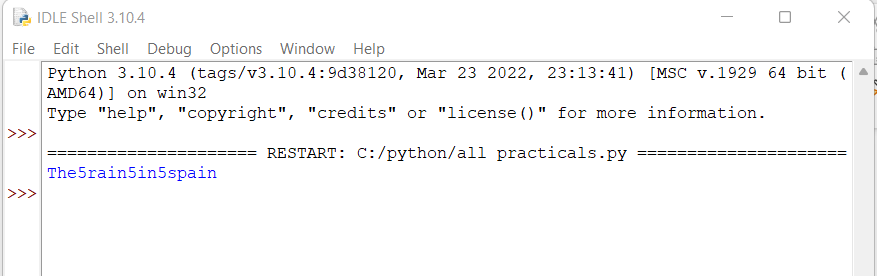
1. import re

txt="The rain in Spain"

x = re.sub ("\s","9", txt)

print (x)

**OUTPUT**

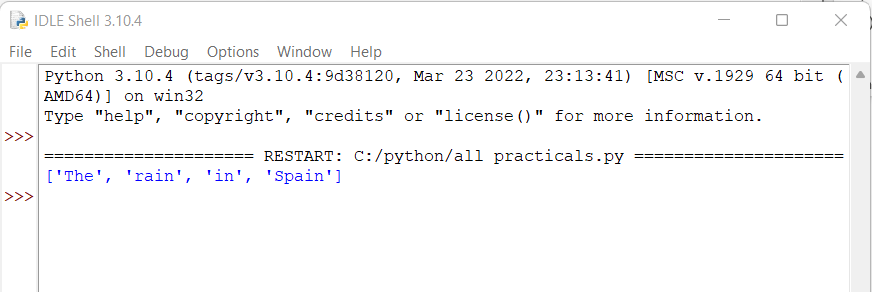
****

1. import re

txt="The rain in Spain"

x=re.split("\s",txt) print(x)

**OUTPUT**



1. import re txt="The rain in Spain"

x=re.search("^The.\*India$", txt)

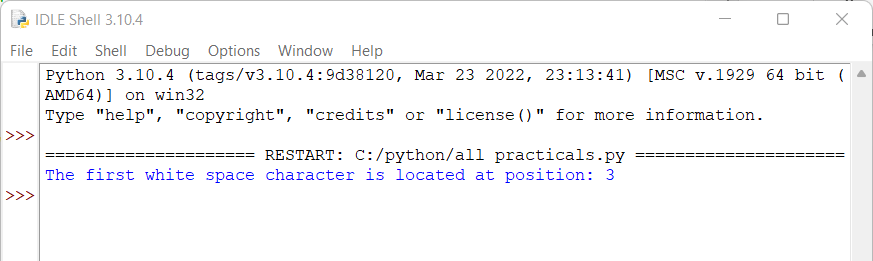
if x:

print("YES! We have a match")

else:

print("No match")

**OUTPUT**

****

**Practical No. 3:**

 **Write a Python Program demonstrate concept of threading and multitasking:**

import time

def cal\_sqre(num):

print(" Calculate the square root of the given

number")

for n in num:

time.sleep(0.3)

print(' Square is : ', n \* n)

def cal\_cube(num):

print (“Calculate the cube of the given number”)

for n in num:

time.sleep(0.3)

print(" Cube is : ", n \* n \*n)

arr = [4, 5, 6, 7, 2]

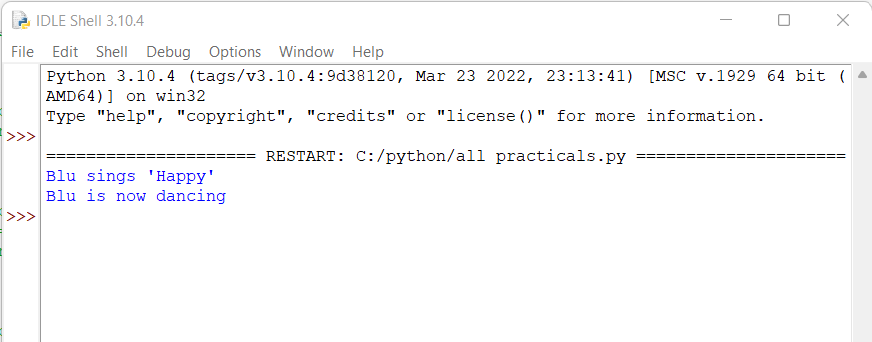
t1 = time.time()

cal\_sqre(arr)

cal\_cube(arr)

print(" Total time taken by threads is :", time.time() - t1)

**OUTPUT**

****

**Practical No. 4:**

* **Write a Python Program to work with databases in python to perform operations such as:**

**i)Creating and dropping.**

**ii)Inserting and updating into tables.**

**i)Creating and dropping.**

import mysql.connector

db=mysql.connector.connect(user='root', password='',host='127.0.0.1',database='test')

cur= db.cursor()

cur.execute ("drop table if exists Item")

cur.execute ('create table Item (Itemno int(4)

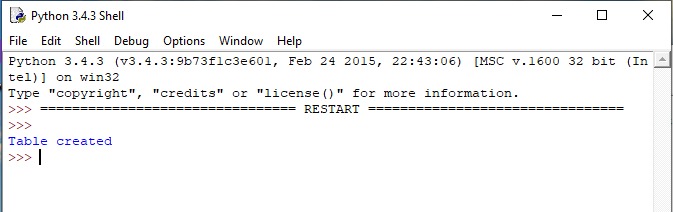
PRIMARY KEY, Itemname varchar (20),Price decimal(9,2),

Quantity int(6))')

print ("Table created")

db.close()

**OUTPUT**

****

**ii) Inserting and updating into tables.**

import mysql.connector db=mysql.connector.connect(user='root',

password='',host='127.0.0.1',database='test') cur=db.cursor()

try:

cur.execute("insert into Item

values(101,'Geometry Box ','50',100)") cur.execute("insert into Item

values(102,'Soap','100',50)")

cur.execute("insert into Item

values(103,'Perfume','150',25)")

cur.execute("insert into Item

values(104,'Pen','50',200)")

cur.execute("insert into Item

values(105,'Pencil','20',100)") db.commit()

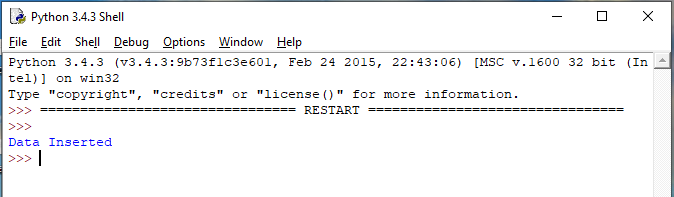
except:

db.rollback()

print(“Data Inserted”)

db.close()

**OUTPUT**

****

**Practical No. 5:**

 **Write a Python Program to demonstrate different types of exception handling:**

1. try:

fh =open("testfile2","w")

fh.write ("this file is my test file

for exception handling!!")

except IOError:

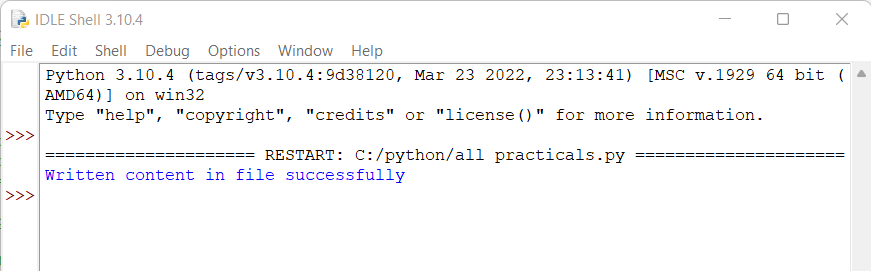
print("Can't find file or read data")

else:

print("Written content in file successfully”)

fh.close()

**OUTPUT**

****

1. try:

a=[40,25,64]

print[a(5)]

except LookUpError:

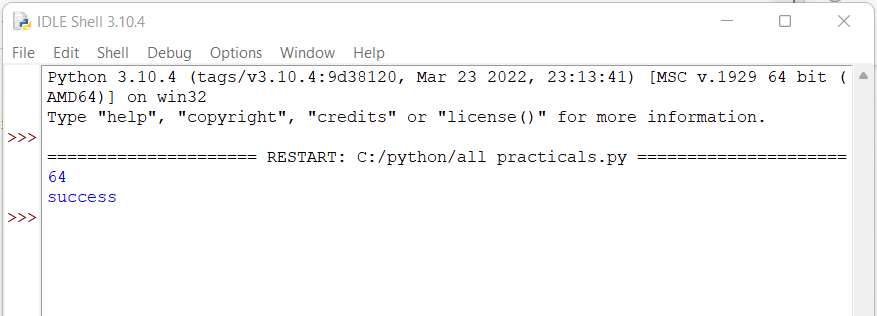
print ("Index out of

bound error”)

else:

print("success")

**OUTPUT**



**Practical No. 6:**

 **Write a GUI Program in Python to design application that demonstrate:**

**i) Different fonts and colours.**

**ii)Different Layouts Managers & Event Handling.**

1. **Different fonts and colours.**

from tkinter import \*

top=Tk()

l=canvas(top,width=100, height=100)

l.pack()

c2=l.create\_line(30,60,90,40,fill=”red”)

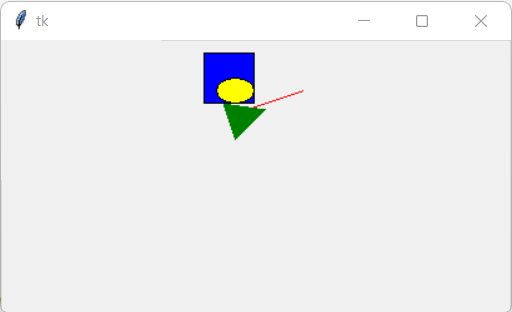
c3=l.create\_rectangle(10,50,50,10,fill=”red”)

c4=l.create\_oval(20,50,50,30,fill=”red”)

c5=l.create\_polygon(25,50,35,80,60,55,fill=”red”)

top.mainloop()

**OUTPUT**

****

**ii)Different Layouts Managers and Event Handling**

from tkinter import \*

def addNumbers():

res=int(e1.get())+int(e2.get())

myText.set(res)

master = Tk()

myText=StringVar()

Label(master, text="First").grid(row=0, sticky=W)

Label(master, text="Second").grid(row=1, sticky=W)

Label(master, text="Result:").grid(row=3, sticky=W)

result=Label(master, text="", textvariable=myText).grid

(row=3,column=1, sticky=W)

e1 = Entry(master)

e2 = Entry(master)

e1.grid(row=0, column=1)

e2.grid(row=1, column=1)

b = Button(master, text="Calculate",command=

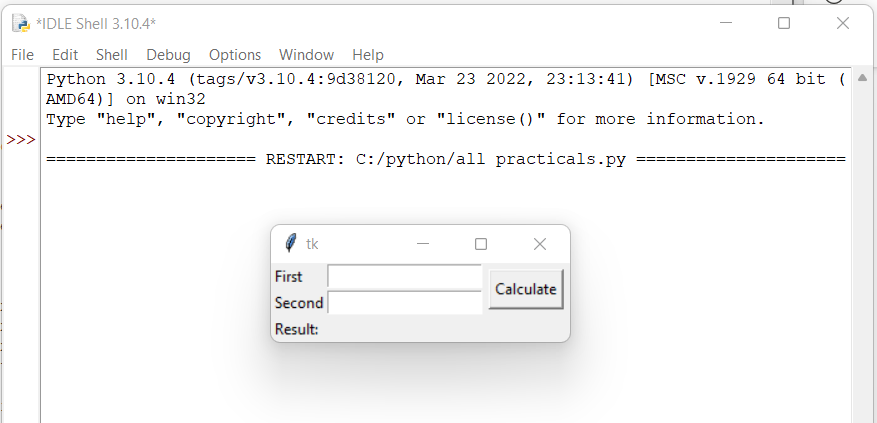
addNumbers)

b.grid(row=0, column=2,columnspan=2,rowspan=2,

sticky=W+E+N+S, padx=5, pady=5)

mainloop()

**OUTPUT**



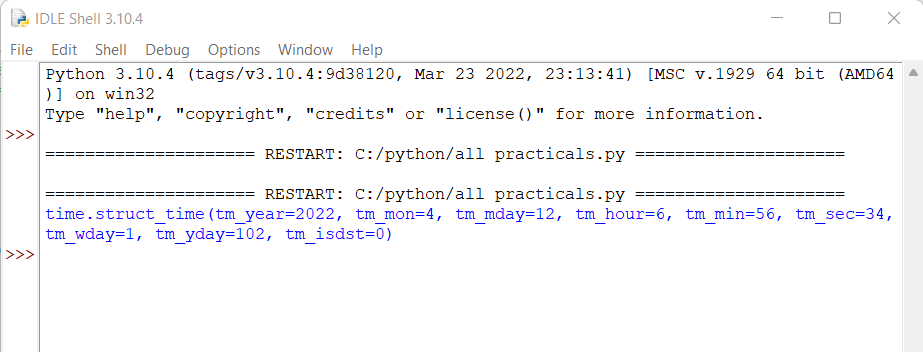
**Practical No. 7:**

* **Write a Python Program to application which uses date and time in python:**

1. import time;

print(time.localtime(time.time()))

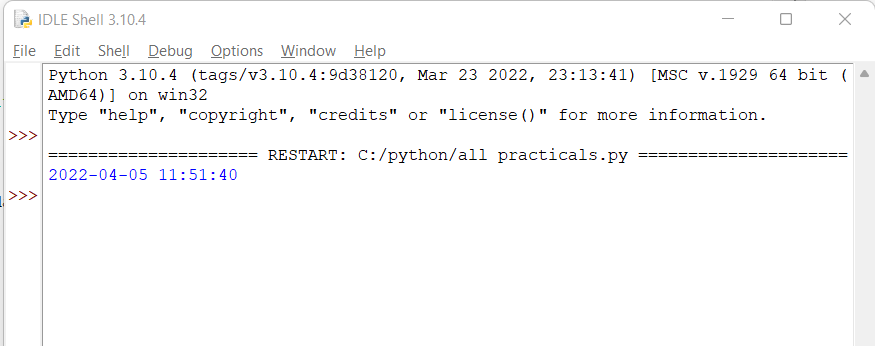
**OUTPUT**

****

1. import datetime;

print(datetime.datetime(2022,4,5,11,51,40))

**OUTPUT**

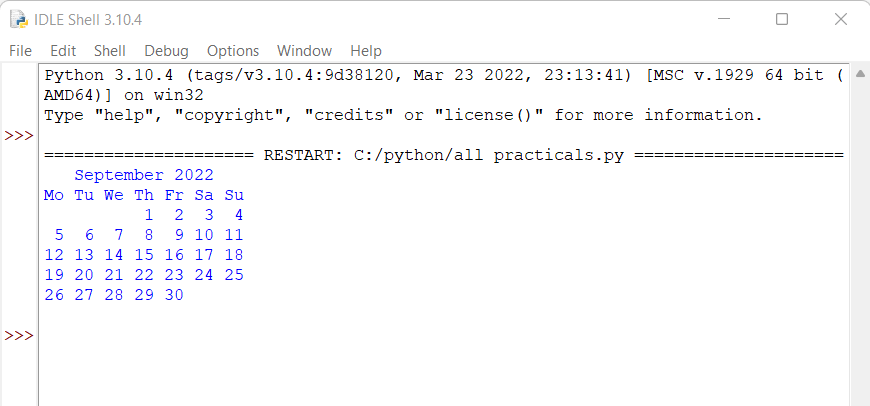
****

1. import calendar;

cal= calendar.month(2022,9)

print(cal)

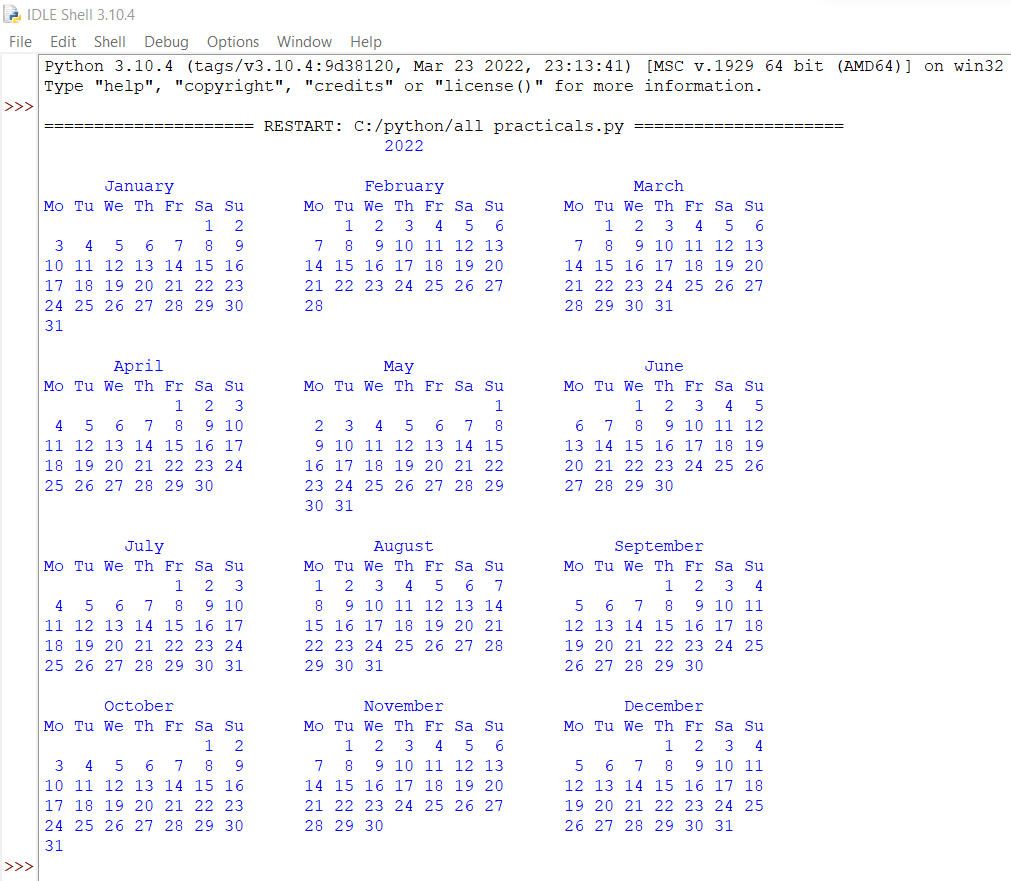
**OUTPUT**

****

1. import calendar;

s=calendar.prcal (2022)

**OUTPUT**

****

**Practical No. 8:**

 **Write a Python Program to implement concepts of OOP such as:**

**i) Types of method**

**ii)Inheritance**

**iii)Polymorphism**

1. **Types of method**

class Parrot;

def \_\_init\_\_(self, name,age):

self.name = name self.age= age

def sing(self, song):

return "{} sings {}".format(self.name, song)

def dance(self):

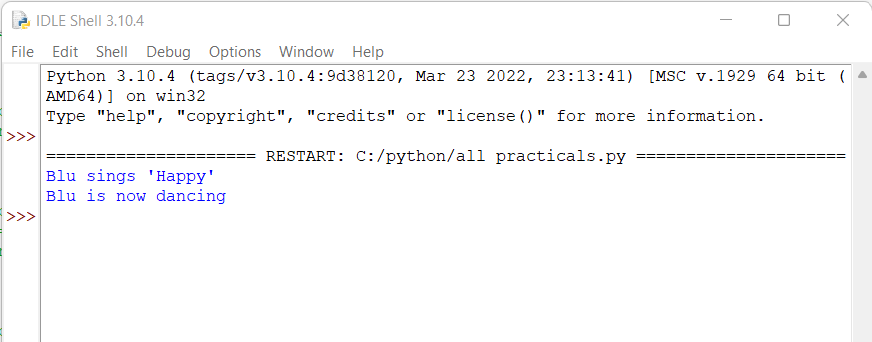
return "{} is now dancing".format(self.name)

blu = Parrot("Blu", 10)

print (blu.sing("'Happy'"))

print (blu.dance())

**OUTPUT**

****

1. **Inheritance:**

class Bird:

def \_\_init\_\_(self, name, age):

print(“Bird is ready”)

def whoisthis(self):

print(“Bird”)

def swim(self):

print(“Swin faster”)

class penguin(Bird):

def\_\_ init\_\_(self):

super().\_\_init\_\_()

print(“Penguin is ready”)

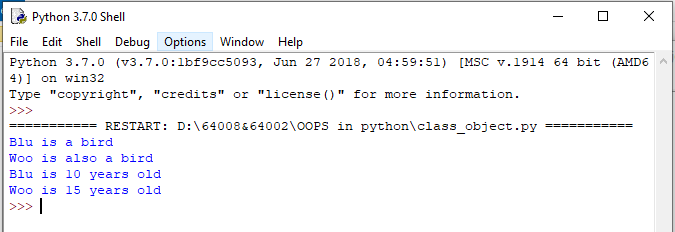
def whoisthis(self):

print (“Penguin”)

def run(self):

print (“Run faster”)

**OUTPUT**

****

1. **Polymorphism:**

class Parrot:

def fly(self):

print("Parrot can fly") def swim(self):

print("Parrot can't swim")

class Penguin:

def fly(self):

print("Penguin can't fly") def swim(self):

print("Penguin can swim")

def flying\_test(bird):

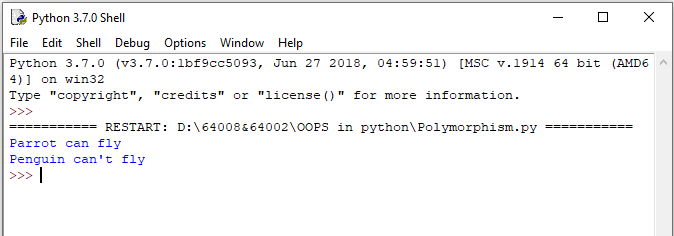
bird.fly()

blu = Parrot()

peggy = Penguin() flying\_test(blu)

flying\_test(peggy)

**OUTPUT**

****