PYTHON PRACTICAL

Practical No :- 01

A. Develop Program to understand the decision control structures of python.

1)

```
amount=0

units=int(input("Enter your units:  "))

if units<=100:

     print("No charges ")

if units>100 and units<=200:

   amount=(units-100)*5

if units>200:

   amount=500(units-100)*10

print("Electricity Bill =",amount)
```

2)

```
num=int(input("Enter any number :"))

last_digit=num%10

if(last_digit%3==0):

   print("{} is divisible by 3".format(last_digit))

else:

   print("{} is not divisible by 3".format(last_digit))
```

3)

```
per=int(input("Enter your percentage"))

if per>90:

   print("The having A grade in examination")

elif per>80 and per<=90:
```

```
    print("The having B grade in examination")
elif per>=60 and per<=80:
    print("The having c grade in examination")
elif per<60:
    print("The having D grade in examination")
```

Output:

```
========================================================================== RESTART: C:/Users/Admin/bill 1.py ========
Enter your units:  100
No charges
Electricity Bill = 0
Enter any number :100
0 is divisible by 3
Enter your percentage
```

Practical No :-02

Develop the program to understand the looping statement.

1)

```
str=input("Enter any word")

vowels=0

for i in str:

    if( i=='a' or i=='e' or i=='i' or i=='o' or i=='u'or i=='A' or i=='E' or i=='I' or
i=='O' or i=='U'):

        vowels=vowels+1

print("number of vowels are :")

print(vowels)
```

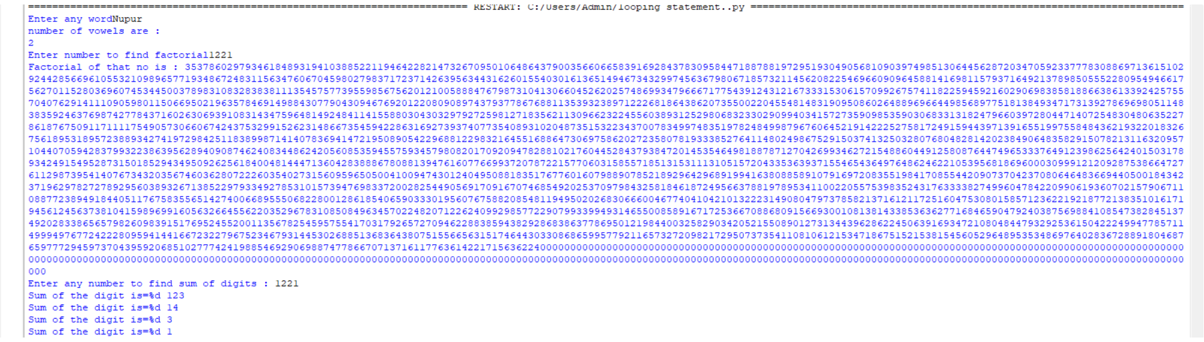
2)

```
no=int(input("Enter number to find factorial"))

fact=1

for i in range(1,no+1):

    fact=fact*i

print("Factorial of that no is :",fact)
```

3)

```
num=int(input("Enter any number to find sum of digits : "))

sum=0

while(num>0):

    reminder=num%10

    num=num//10

    sum=num+reminder

    print("Sum of the digit is=%d",sum)
```

Output:



```
========================================================================= RESTART: C:/Users/Admin/looping statement..py ========================================================================
Enter any wordNupur
number of vowels are :
2
Enter number to find factorial1221
Factorial of that no is : 3537860297934618489319410388522119464228214732670950106486437900356606658391692843783095844718878819729519304905681090397498513064456287203470592337778308869713615102
924428566961055321098965771934867248311563476067045980279837172371426395634431626015540301613651494673432997456367980671857321145620822546966090964588141698115793716492137898505552280954946617
562701152803696074534450037898310832838381113545757739559856756201210058884767987310413066045262025748699347966671775439124312167333153061570992675741182259459216029069838581886638613392425755
704076291411109059801150669502196357846914988430779043094676920122080908974379377867688113539323897122268186438620735500220455481483190950860264889696644985689775181384934717313927869698051148
383592463769874277843716026306939108314347596481492484114155880304303297927259812718356211309662322455603893125298068323302909940341572735909853590306833131824796603972804471407254830480635227
861876750911711117549057306606742437532991526231486673545942286316927393740773540893102048735153223437007834997483519782484998796760645219142225275817249159443971391655199755848436219322018326
756189531895723889342741972984251183899871414078369414721950890542296881229832164551688647306975862027235807819333852764114802498675291503741325032807680482814202384906483582915078213116320957
104407059428379932238639562894090874624083448624205608535945575934579808201709209478288102176044528437938472014535464981887871270426993462721548860449125808764474965333764912398625642401503178
934249154952873150185294349509262561840048144471360428388867808813947616077669937207872215770603158557185131531113105157204335363937155465436497648624622105395681869600030999121209287538664727
611298739541407673432035674603628072226035402731560959650500410094743012404950881835176776016079889078521892964296891994163808858910791697208355198417085544209073704237080646483669440500184342
371962978272789295603893267138522979334927853101573947698337200282544905691709167074685492025370979843258184618724956637881978953411002205575398352431763333827499604784220990619360702157906711
088772389491844051176758355651427400668955506822800128618540659033301956076758820854811949502026830666004677404104210132223149080479737858213716121172516047530801585712362219218772138351016171
945612456373810415989699160563266455622035296783108508496345702248207122624099298577229079933994931465500858916717253667088680915669300108138143385363627716846590479240387569884108547382845137
492028338656579826098391517695245520011356782545957554170317926572709462288385943829286838637786950121984400325829034205215508901273134439628622450639169347210804844793292536150422249947785711
499949767724222809594144166723227967523467931445302688513683643807515566563151746443033086865995779211657327209821729507373541108106121534718675152153815456052964895353486976402836728891804687
659777294597370439592068510277742419885469290698874778667071371611776361422171563622400000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000
Enter any number to find sum of digits : 1221
Sum of the digit is=%d 123
Sum of the digit is=%d 14
Sum of the digit is=%d 3
Sum of the digit is=%d 1
```

Practical No :-03

Develop programs to learn different types of structures(list, dictionary, tuples in python.

1)Write a program to demonstrate list sequence.

```
# Update the elements of the list

a = ['Ansh', 2, 35, 'code', 2.5]
print("Length of the list", len(a))
print("Before list",a)
a[0] = "ash"
print("After update list",a)

# Add new elements in to the list – append() , insert()
a = ['Ansh', 2, 35, 'code', 2.5]
print("list",a)
print("Length of the list", len(a))
a.append("kash")
print("list",a)
print("Length of the list", len(a))
a.insert(1,25)
print("list",a)
print("Length of the list", len(a))
# Removing elements from the list – remove() del
a = ['Ansh', 2, 35, 'code', 2.5]
print("list",a)
print("Length of the list", len(a))
b = a.pop(1)
print("element deleted using pop()",b)
print("list",a)
a.remove(35)
print("element deleted using remove()",a)
del a[1:3]
print("element deleted using del keyword",a)

# Find out length of the list
```

```
a = ['Ansh', 2, 35, 'code', 2.5]
print("list",a)
print("Length of the list", len(a))

# Find out maximum between list

a = [1,5,8,6,2]
print("Largest element of list", max(a))

# Find out minimum between list

a = [1,5,8,6,2]
print("Smallest element of list", min(a))
```

Output:

```
========================================================================== RESTART: C:/Users/Admin/list.py ==========================================================================
Length of the list 5
Before list ['Ansh', 2, 35, 'code', 2.5]
After update list ['ash', 2, 35, 'code', 2.5]
list ['Ansh', 2, 35, 'code', 2.5]
Length of the list 5
list ['Ansh', 2, 35, 'code', 2.5, 'kash']
Length of the list 6
list ['Ansh', 25, 2, 35, 'code', 2.5, 'kash']
Length of the list 7
list ['Ansh', 2, 35, 'code', 2.5]
Length of the list 5
element deleted using pop() 2
list ['Ansh', 35, 'code', 2.5]
element deleted using remove() ['Ansh', 'code', 2.5]
element deleted using del keyword ['Ansh']
list ['Ansh', 2, 35, 'code', 2.5]
Length of the list 5
Largest element of list 8
Smallest element of list 1
>>>
```

B. Write aprogram to demonstrate Tuple sequence.

```
# Write a program to demonstrate Tuple sequence.

# Creating Tuple

a=()
tup = tuple()
print(type(a))
print(type(tup))

# Read the elements of the tuple from the user

lst=[]
a = int((input("Enter the size of tuple")))
for i in range(0,a):
    b = (input("Add tuple element"))
    lst.append(b)
```

```
tup = tuple(lst)
print(tup)
print(type(tup))

# Print the elements using slice operator.

tup=('apple', 'realme', 'redmi', 'moto')
print(tup[:])
print(tup[:3])
print(tup[3:])
print(tup[::-1])
print(tup[-1:])

# Deleting Tuple

tup = ('apple', 'realme', 'redmi', 'moto')
print("Tuple",tup)
del tup

# Find out length of the Tuple

tup = ('apple', 'realme', 'redmi', 'moto')
print(tup)
print(" Length of Tuple",len(tup))

# Find out maximum between list

tup = (25,80,74,62,250)
print(tup)
print(type(tup))
print(" Maximum of Tuple",max(tup))

# Find out minimum between list

tup = (25,80,74,62,250)
print(tup)
print(" Minimum of Tuple",min(tup))
```

Output:

```
= RESTART: C:/Users/Admin/tuple.py
<class 'tuple'>
<class 'tuple'>
Enter the size of tuple3
Add tuple elementapple
Add tuple elementmango
Add tuple elementbanana
('apple', 'mango', 'banana')
<class 'tuple'>
('apple', 'realme', 'redmi', 'moto')
('apple', 'realme', 'redmi')
('moto',)
('moto', 'redmi', 'realme', 'apple')
('moto',)
Tuple ('apple', 'realme', 'redmi', 'moto')
('apple', 'realme', 'redmi', 'moto')
 Length of Tuple 4
(25, 80, 74, 62, 250)
<class 'tuple'>
 Maximum of Tuple 250
(25, 80, 74, 62, 250)
 Minimum of Tuple 25
```

C. Demonstrate the Dictionary.

```
# C. Demonstrate the dictionary

# Creating Dictionary
dic={}
my_dict={"Car1": "Audi", "Car2":"BMW",
        "Car3":"Mercidies Benz","Car4":"Range Rover"}
print(type(dic))
print(my_dict)
print(type(my_dict))

# Sort dictionary by values(Ascending and descending)

import operator
#
d = {1: 2, 3: 4, 4: 3, 2: 1, 0: 0}
#
s= sorted(d.items(), key=operator.itemgetter(1))
#
```

```
print('ascending order : ',s)
#
s1= dict( sorted(d.items(), key=operator.itemgetter(1),reverse=True))
#
print('descending order : ',s1)

# concatenate two dictionaries to create one

car1_model={'Mercedes':1960}
car2_model={'Audi':1970}
car2_model.update(car1_model)
print(car2_model)

# check whether the key exist or not

my_dict={"Car1": "Audi", "Car2":"BMW",
        "Car3":"Mercidies Benz","Car4":"Range Rover"}
#
key = input("Enter the key you want to search:\n")
#
if key in my_dict.keys():
    print("Present")
else:
    print("Not Present")

# iterate the keys of dictionary

my_dict={"Car1": "Audi", "Car2":"BMW",
        "Car3":"Mercidies Benz","Car4":"Range Rover"}
for x in my_dict:
    print(x)

# iterate the values of dictionary

my_dict={"Car1": "Audi", "Car2":"BMW",
        "Car3":"Mercidies Benz","Car4":"Range Rover"}
for x in my_dict.values():
```

```
    print(x)

# iterate the items of dictionary

my_dict={"Car1": "Audi", "Car2":"BMW",
         "Car3":"Mercidies Benz","Car4":"Range Rover"}
for x in my_dict.items():
    print(x)

# remove the specific values

my_dict={"Car1": "Audi", "Car2":"BMW",
        "Car3":"Mercidies Benz","Car4":"Range Rover"}
print("Original Dict \n",my_dict)
my_dict.pop('Car3')
print("Element removed using pop \n",my_dict)
del my_dict['Car2']
print("Element removed using del keyword \n",my_dict)
```

Output:

```
= RESTART: C:/Users/Admin/(Ascending and descending).py
<class 'dict'>
{'Car1': 'Audi', 'Car2': 'BMW', 'Car3': 'Mercidies Benz', 'Car4': 'Range Rover'}
<class 'dict'>
ascending order :  [(0, 0), (2, 1), (1, 2), (4, 3), (3, 4)]
descending order :  {3: 4, 4: 3, 1: 2, 2: 1, 0: 0}
{'Audi': 1970, 'Mercedes': 1960}
Enter the key you want to search:
```

Practical No :- 04

Develop program to learn concept of functions scoping, recursion and list ,mutability.

→Function Scoping :-

1. **# It is global function**

```
x = "global"
def fun():
    print("It is global scope inside :", x)
fun()
print("It is global scope outside :", x)

# It is local function
def myfunc():
  x = ("It is local variable")
  print(x)

myfunc()
```

Output:

```
= RESTART: C:/Users/Admin/local fuction.py
It is global scope inside : Nupur
It is global scope outside : Nupur
It is local variable
```

2. **# It is recursion Function**

```
def factorial(x):
    """This is a recursive function
    to find the factorial of an integer"""
   if x == 1:
        return 1
    else:
        return (x * factorial(x-1))

num = 3
```

```
print("The factorial of", num, "is", factorial(num))
```

Output:

```
================= RESTART: C:/Users/Admin/recursive function.py =
The factorial of 3 is 6
```

Practical No :- 05

D. Develop program to understand object oriented programming using python.

```
class Raisoni:
    def student(self):
        print("All Raisoni Students")
class Education(MCA):
    def division(self):
        print(“Welcome")

e=Education()
e.student()
e.division()
```

Output:

```
= RESTART: C:/Users/Admin/k.py
All Raisoni Students
Welcome
```

Practical No :- 06

Develop programs for data structure algorithms using python.

1. **Searching:-**

#Linear Search

```
class linearsearch:
    ele=[]
    def get(self):
        self.a=int(input("Enter no of element you want to insert"))
        for i in range(0,self.a):
            b=(input("Add element you want to search"))
            self.ele.append(b)
    def search(self):
        c=int(input("Enter elemnt you want to search"))
        for i in range(0,self.a):
            if self.ele[i]==c:
                break;
        if i<self.a:
            print("Element found at index : ",i+1)
        else:
            print("Not Found .....")

s=linearsearch()
s.get()
s.search()
```

Output:

```
======================= RESTART: C:/Users/Admin/linear.py ======
Enter no of element you want to insert1
Add element you want to search0
Enter elemnt you want to search0
Element found at index :  1
```

2. **Sorting :-**

#Bubble Sort

```
class sort:
    ele=[]
```

```
    def get(self):
        self.a=int(input("Enter no of element you want to insert"))
        for i in range(0,self.a):
            b=int((input("Add element you want to sort")))
            self.ele.append(b)
        print(self.ele)
    def bsort(self):
        for i in range(0,self.a):
            for j in range(0,self.a-i-1):
                if self.ele[j]>self.ele[j+1]:
                    temp=self.ele[j]
                    self.ele[j]=self.ele[j+1]
                    self.ele[j+1]=temp
    def show(self):
        print("Sorted list")
        print(self.ele)
s=sort()
s.get()
s.bsort()
s.show()
```

Output:

```
======================== RESTART: C:/Users/Admin/sort.py ========================
Enter no of element you want to insert1
Add element you want to sort2
[2]
Sorted list
[2]
```

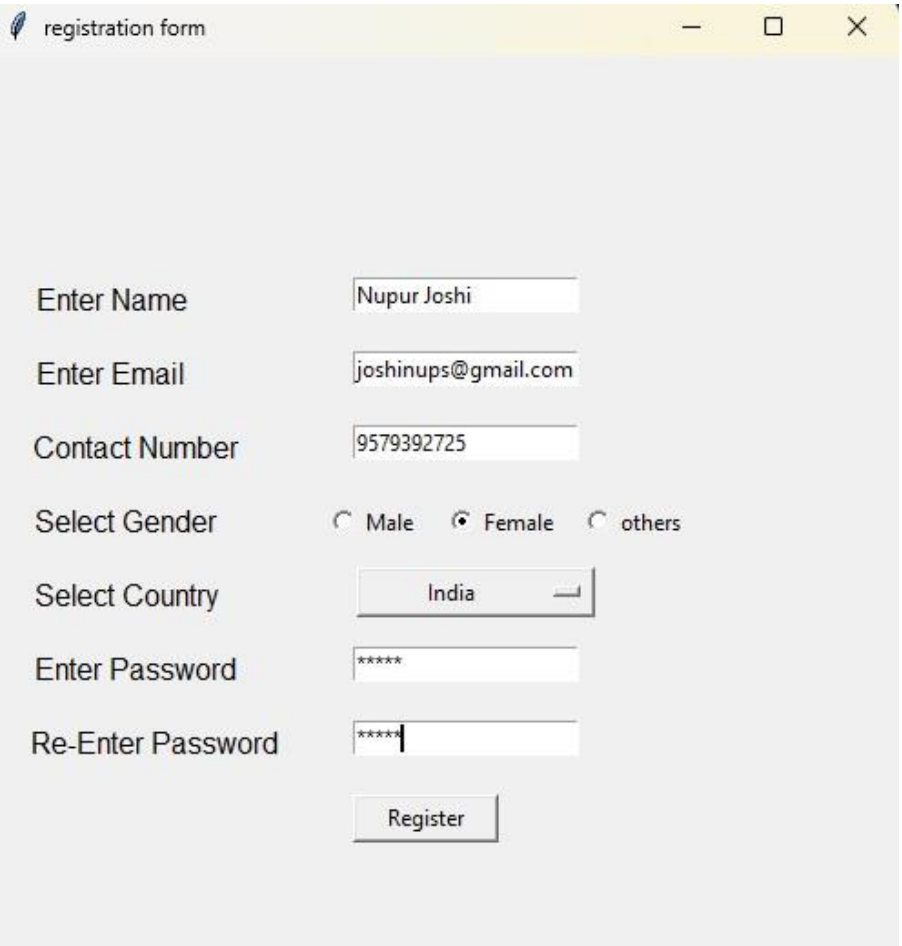
Practical:-7

1.Develop program to learn GUI programming using Tkinter

```
from tkinter import *

base = Tk()

base.geometry("500x500")

base.title("registration form")

lb1= Label(base, text="Enter Name", width=10, font=("arial",12))

lb1.place(x=20, y=120)

en1= Entry(base)

en1.place(x=200, y=120)

lb3= Label(base, text="Enter Email", width=10, font=("arial",12))

lb3.place(x=19, y=160)

en3= Entry(base)

en3.place(x=200, y=160)

lb4= Label(base, text="Contact Number", width=13,font=("arial",12))

lb4.place(x=19, y=200)

en4= Entry(base)

en4.place(x=200, y=200)

lb5= Label(base, text="Select Gender", width=15, font=("arial",12))

lb5.place(x=5, y=240)

vars = IntVar()

Radiobutton(base, text="Male", padx=5,variable=vars, value=1).place(x=180,
y=240)

Radiobutton(base, text="Female", padx =10,variable=vars,
value=2).place(x=240,y=240)
```

```
Radiobutton(base, text="others", padx=15, variable=vars,
value=3).place(x=310,y=240)

list_of_cntry = ("United States", "India", "Nepal", "Germany")

cv = StringVar()

drplist= OptionMenu(base, cv, *list_of_cntry)

drplist.config(width=15)

cv.set("United States")

lb2= Label(base, text="Select Country", width=13,font=("arial",12))

lb2.place(x=14,y=280)

drplist.place(x=200, y=275)

lb6= Label(base, text="Enter Password", width=13,font=("arial",12))

lb6.place(x=19, y=320)

en6= Entry(base, show='*')

en6.place(x=200, y=320)

lb7= Label(base, text="Re-Enter Password", width=15,font=("arial",12))

lb7.place(x=21, y=360)

en7 =Entry(base, show='*')

en7.place(x=200, y=360)

Button(base, text="Register", width=10).place(x=200,y=400)

base.mainloop()
```

Output:


registration form
Enter Name
Nupur Joshi
Enter Email
joshinups@gmail.com
Contact Number
9579392725
Select Gender
Male
Female
others
Select Country
India
Enter Password
*****
Re-Enter Password
*****
Register

Practical:-9

Demonstrate the concept of exception handling programs For Try Catch And Finally :

1) Try and Catch :

```
n = int(input("enter 1st no: "))
m = int(input("enter 2nd no: "))
try:
   x = n/m
except ZeroDivisionError:
      print("Sorry ! You are dividing by zero ")
else:
   print("Division of two nos. is : ",x)
```

2) Try and Finally:

```
 n = int(input("enter 1st no: "))

m = int(input("enter 2nd no: "))

try:

   x = n/m

   print("Division of two nos. is : ",x)

except ZeroDivisionError:

      print("Sorry ! You are dividing by zero ")

finally:

   print("This statement execute anyways")
```

Output:

```
====================== RESTART: C:/Users/Admin/trycatch.py =====================
enter 1st no: 10
enter 2nd no: 20
Division of two nos. is :  0.5
enter 1st no: 30
enter 2nd no: 40
Division of two nos. is :  0.75
This statement execute anyways
```

Practical 10

E. Demonstrate implementation of the Anonymous Function Lambda.

A. Simple Lambda

```
x = lambda a, b, c : a + b + c

print(x(5, 6, 2))
```

Output:

```
======================== RESTART: C:/Users/Admin/simple.py =======================
13
```

B. Cube Lambda

```
# Python code to illustrate cube of a number

# showing difference between def() and lambda().

def cube(y):

        return y*y*y

lambda_cube = lambda y: y*y*y

# using the normally

# defined function

print(cube(5))

# using the lambda function

print(lambda_cube(5))
```

<u>OR</u>

```
def cube(y):

    return y*y*y
```

```
c = int(input("Enter the no: "))

lambda_cube = lambda y: y*y*y


print("Lanbda Cube is: ",lambda_cube())
```

Output:

```
====================== RESTART: C:/Users/Admin/hbfhdfv.py ======
125
125
```

Practical 11

Demonstrate implementation Mapping Functions over Sequences.

```
def mul(i):
    return i * i
num = (3, 5, 7, 11, 13)
update = map(mul, num)#map(square,num)
print(update)
# making the map object readable
mul_output = list(update)
print(mul_output)
```

Output:

```
========= RESTART: C:/Users/Admin/Mapping Functions over Sequences..py ===
<map object at 0x000001FC46D737C0>
[9, 25, 49, 121, 169]
```

Practical 12

Demonstrate implementation functional programming tools such as filter And Reduce

a) **********

```
scores = [66, 90, 68, 59, 76, 60, 88, 74, 81, 65]

def is_A_student(score):

        return score > 75

over_75 = list(filter(is_A_student, scores))

print(over_75)
```

b) **********

```
dromes = ("demigod", "rewire", "madam", "freer", "anutforajaroftuna", "kiosk")

palindromes = list(filter(lambda word: word == word[::-1], dromes))

print(palindromes)
```

c) **********

```
# Python 3

from functools import reduce

numbers = [3, 4, 6, 9, 34, 12]

def custom_sum(first, second):

        return first + second

result = reduce(custom_sum, numbers)

print(result)
```

d) **********

```
from functools import reduce

numbers = [3, 4, 6, 9, 34, 12]

def custom_sum(first, second):

return first + second

result = reduce(custom_sum, numbers, 10)
```

print(result)

Output:

```
= RESTART: C:/Users/Admin/programming tools such as filter And Reduce.py
[90, 76, 88, 81]
['madam', 'anutforajaroftuna']
68
78
```

Practical 13

Demonstrate the Module Creation, Module usage, Module Namespaces,Reloading Modules, Module Packages, Data Hiding in Modules.

a) **Module Creation:**

```
def add(a, b):
        """This program adds two
        numbers and return the result"""
        result = a + b
        return resultimport math
print("The value of pi is", math.pi)
import math as m
print("The value of pi is", m.pi)
#from...import statement
from math import pi
print("The value of pi is", pi)
from math import *
print("The value of pi is", pi)
```

Output:

```
= RESTART: C:/Users/Admin/Reloading Modules, Module Packages, Data Hiding in Mod
ules..py
The value of pi is 3.141592653589793
The value of pi is 3.141592653589793
The value of pi is 3.141592653589793
The value of pi is 3.141592653589793
```