**PART-A**

**LAB PROGRAM 1 A**

Design and Implement a Python program to accept 3 digits from the user and print all possible combination from digits.

**PROGRAM**

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

b=int(input("Enter second number:"))

c=int(input("Enter third number:"))

d=[]

d.append(a)

d.append(b)

d.append(c)

for i in range(0,3):

for j in range(0,3):

for k in range(0,3):

if(i!=j and j!=k and k!=i):

print(d[i],d[j],d[k])

**OUTPUT**

Enter first number:1

Enter second number:2

Enter third number:3

1 2 3 (0,1,2)

1 3 2 (0,2,1)

2 1 3 1,0,2

2 3 1 1,2,0

3 1 2 2,0,1

3 2 1 2,1,0

**LAB PROGRAM 1 B**

Create a Python program to take two command line inputs and compute the GCD and LCM of these two numbers.

**PROGRAM**

num1=int(input("Enter first number: "))

num2=int(input("Enter second number: "))

smaller = num2 if num1 > num2 else num1

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

if((num1 % i == 0) and (num2 % i == 0)):

gcd = i

print("The GCD is",gcd)

lcm = (num1\*num2)//gcd

print("The LCM is",lcm)

**OUTPUT**

Enter first number: 8

Enter second number: 12

The GCD is 4

The LCM is 24

***“””Using command line input“””***

import sys

program\_name=sys.argv[0]

arguments=sys.argv[1:]

num1=int(arguments[0])

num2=int(arguments[1])

print("Values Entered",num1,num2)

smaller = num2 if num1 > num2 else num1

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

if((num1 % i == 0) and (num2 % i == 0)):

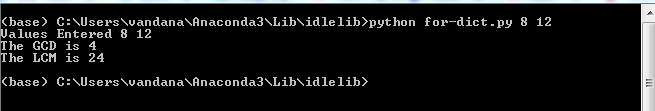
gcd = i

print("The GCD is",gcd)

lcm = (num1\*num2)//gcd

print("The LCM is",lcm)

**OUTPUT**



**LAB PROGRAM 2 A**

Create a Python program to find the sum of natural numbers up to n using recursive function

**PROGRAM**

def recur\_sum(n):

if n <= 1:

return n

else:

return n + recur\_sum(n-1)

num = 5

if num < 0:

print("Enter a positive number")

else:

print("The sum is",recur\_sum(num))

**OUTPUT**

Enter the value of n: 5

The sum is 15

**LAB PROGRAM 2 B**

Design and Develop a Python Program to Create a Dictionary with Key as First Character and Value as Words Starting with that Character.

**PROGRAM**

test\_string=input("Enter string:")

l=test\_string.split()

d={}

for word in l:

if(word[0] not in d.keys()): # char already in dictionary , so just add as value

d[word[0]]=[]

d[word[0]].append(word)

else:

if(word not in d[word[0]]):

d[word[0]].append(word)

for k,v in d.items():

print(k,":",v)

**OUTPUT**

Enter string:Iam at home and Iam happy to take rest

I : ['Iam']

a : ['at', 'and']

h : ['home', 'happy']

t : ['to', 'take']

r : ['rest']

**LAB PROGRAM 3A**

A list rotation consists of taking the last element and moving it to the front. For instance, if we rotate the list [1,2,3,4,5], we get [5,1,2,3,4]. If we rotate it again, we get [4,5,1,2,3]. Write a Python function rotatelist(ls,k) that takes a list ls and a positive integer k and returns the list ls after k rotations. If k is not positive, your function should return ls unchanged. Note that your function should not change ls itself, and should return the rotated list. Here are some examples to show how your function should work.

>>>rotatelist([1,2,3,4,5],1)

#output is [5, 1, 2, 3, 4]

>>>rotatelist([1,2,3,4,5],3)

#output is [3, 4, 5, 1, 2]

>>>rotatelist([1,2,3,4,5],12)

#output is [4, 5, 1, 2, 3]

**PROGRAM**

def rotatelist(ls,k):

if k<0:

return ls

else:

mylist=ls[:]

for i in range(0,k):

mylist=[mylist[-1]]+mylist[0:-1]

return mylist

size=int(input("Enter the size of the list"))

ls=[]

for i in range(0,size):

val=int(input("Enter the list element"))

ls.append(val)

print("Entered list is ", ls)

k=int(input("Enter the value of K"))

print("Final list",rotatelist(ls,k))

**OUTPUT**

Enter the size of the list4

Enter the list element12

Enter the list element2

Enter the list element3

Enter the list element56

Entered list is [12, 2, 3, 56]

Enter the value of K 2

Final list [3, 56, 12, 2]

**LAB PROGRAM 3B**

Design and implement a python code that accepts two string from user and displays the characters which are present in both the strings. Use Set sequence type to achieve the same.

**PROGRAM**

s1=input("Enter first string:")

s2=input("Enter second string:")

a=list(set(s1) & set(s2))

print("The common letters are:")

for i in a:

print(i)

**OUTPUT**

Enter first string:hello

Enter second string:world

The common letters are:

o

l

**LAB PROGRAM 4 A**

Implement a Python program to count the numbers of characters in the string and store them in a dictionary data structure

**PROGRAM**

test\_string=input("Enter string: ")

l=test\_string.split()

d={}

for word in l:

if(word not in d.keys()): # char already in dictionary , so just add as value

d[word]=len(word)

for k,v in d.items():

print(k,":",v)

**OUTPUT**

Enter string: iam learning python

iam : 3

learning : 8

python : 6

**LAB PROGRAM 4 B**

Develop a Python program to print first 10 lines and last 10 lines in a file.

**PROGRAM**

file=input("Enter a file name:")

with open(file) as fhand:

s=fhand.read() #s stores whole file contents

ls=s.split('\n') #split s based on new-line character

print('First 10 lines are:')

for i in ls[:10]:

print(i)

print('Last 10 lines are:')

for i in ls[-10:]:

print(i)

fhand.close()

**OUTPUT**

Enter a file name:for-dict.py

First 10 lines are:

file=input("Enter a file name:")

with open(file) as fhand:

s=fhand.read() #s stores whole file contents

ls=s.split('\n') #split s based on new-line character

print('First 10 lines are:')

for i in ls[:10]:

print(i)

print('Last 10 lines are:')

for i in ls[-10:]:

Last 10 lines are:

ls=s.split('\n') #split s based on new-line character

print('First 10 lines are:')

for i in ls[:10]:

print(i)

print('Last 10 lines are:')

for i in ls[-10:]:

print(i)

fhand.close()

**LAB PROGRAM 5 A**

Design a python program to compute the number of characters, words and lines in a file. Also Print the most frequent words read from the file.

**PROGRAM**

file=input("Enter a file name:")

d={}

with open(file) as fhand:

lCount=0

wCount=0

cCount=0

for line in fhand:

lCount+=1

cCount+=len(line)

ls=line.split()

for word in ls:

if(word not in d.keys()): # word not in dictionary , so just add as value

d[word]= 1

else:

d[word]=d[word]+1

wCount+=len(ls)

print('Number of lines=',lCount)

print('Number of words=', wCount)

print('Number of characters=', cCount)

print(d)

# Find item with Max Value in Dictionary

itemMaxValue = max(d.items(), key=lambda x: x[1])

listOfKeys = list()

# Iterate over all the items in dictionary to find keys with max value

for key, value in d.items():

if value == itemMaxValue[1]:

listOfKeys.append(key)

print('Word most frequent', listOfKeys)

fhand.close()

**OUTPUT**

tt.py

iam fine how

how are you you

how fine

-------------------------------------------

Enter a file name:tt.py

Number of lines= 3

Number of words= 9

Number of characters= 38

{'iam': 1, 'fine': 2, 'how': 3, 'are': 1, 'you': 2}

Word most frequent ['how']

**LAB PROGRAM 5B**

Apply import,from, \* and other module related concepts to create a module called ―calc‖ consists of 4 function that should return sum, division, multiplication and subtraction. Create another module caller ―user ,import the calc module and illustrate the use of all the functions of calc module.

**PROGRAM**

Calc.py

def add(a, b):

result = a + b

return result

def sub(a, b):

result = a-b

return result

def mul(a, b):

result = a \* b

return result

def div(a, b):

result = a / b

return result

User.py

#import calc

from calc import add,sub

#from calc import \*

import calc as c

print(add(5,2))

print(sub(5,2))

print(c.mul(5,2))

#print(calc.div(5,2

=========== RESTART: C:\Users\vandana\Anaconda3\Lib\idlelib\use.py ===========

7

3

10

**LAB PROGRAM 6**

Design & Implement the program in python to demonstrate sending Email and Text messages over the web.

**PRE-REQUISTICS**

1. While logged into your gmail at gmail.com, go to https://myaccount.google.com/security
2. Scroll down to the part that says "Allow less secure apps"
3. Turn ON "allow less secure apps".

If you don't want to make your main gmail less secure, or if you don't already have gmail, then sign up for a new gmail solely for this purpose.



**CONCEPT & PROCEDURE**

>>> import smtplib

>>> smtpObj = smtplib.SMTP('smtp.gmail.com', 587)

>>> smtpObj.ehlo()

(250, b'smtp.gmail.com at your service, [103.6.156.153]\nSIZE 35882577\n8BITMIME\nSTARTTLS\nENHANCEDSTATUSCODES\nPIPELINING\nCHUNKING\nSMTPUTF8')

>>> smtpObj.starttls()

(220, b'2.0.0 Ready to start TLS')

>>> smtpObj.login('vandana.xyz@gmail.com','xyzxyz')

(235, b'2.7.0 Accepted')

>>> smtpObj.sendmail('vandana.hareesh@gmail.com','hareesh.r@gmail.com','Subject: Testing mail check through Python code.\n Thank you')

{}

>>> smtpObj.quit()

(221, b'2.0.0 closing connection u23sm11794379pgj.58 - gsmtp')

**PROGRAM**

import smtplib

gmailaddress = input("what is your gmail address? \n ")

gmailpassword = input("what is the password for that email address? \n ")

mailto = input("what email address do you want to send your message to? \n ")

msg = input("What is your message? \n ")

mailServer = smtplib.SMTP('smtp.gmail.com' , 587)

mailServer.ehlo()

mailServer.starttls()

mailServer.login(gmailaddress , gmailpassword)

mailServer.sendmail(gmailaddress, mailto, msg)

print(" \n Sent!")

mailServer.quit()

**OUTPUT**

what is your gmail address?

vandana.hareesh@gmail.com

what is the password for that email address?

xyxxyz

what email address do you want to send your message to?

vandanacp@newhorizonindia.edu

What is your message?

Subject: Python Test Mail

Sent!



**PART-B**

**LAB PROGRAM 7**

Design and Develop a Python Program to Append, Delete and Display Elements of a List Using Classes and Objects.

**PROGRAM**

class check():

def \_\_init\_\_(self):

self.n=[]

def add(self,a):

return self.n.append(a)

def remove(self,b):

self.n.remove(b)

def dis(self):

return (self.n)

obj=check()

choice=1

while choice!=0:

print("0. Exit")

print("1. Add")

print("2. Delete")

print("3. Display")

choice=int(input("Enter choice: "))

if choice==1:

n=int(input("Enter number to append: "))

obj.add(n)

print("List: ",obj.dis())

elif choice==2:

n=int(input("Enter number to remove: "))

obj.remove(n)

print("List: ",obj.dis())

elif choice==3:

print("List: ",obj.dis())

elif choice==0:

print("Exiting!")

else:

print("Invalid choice!!")

print()

OUTPUT

========== RESTART: C:/Users/vandana/Anaconda3/Lib/idlelib/lab7.py ==========

0. Exit

1. Add

2. Delete

3. Display

Enter choice: 1

Enter number to append: 12

List: [12]

0. Exit

1. Add

2. Delete

3. Display

Enter choice: 1

Enter number to append: 13

List: [12, 13]

0. Exit

1. Add

2. Delete

3. Display

Enter choice: 1

Enter number to append: 14

List: [12, 13, 14]

0. Exit

1. Add

2. Delete

3. Display

Enter choice: 3

List: [12, 13, 14]

0. Exit

1. Add

2. Delete

3. Display

Enter choice: 2

Enter number to remove: 13

List: [12, 14]

0. Exit

1. Add

2. Delete

3. Display

Enter choice:0

**LAB PROGRAM 8**

Design and Implement a Python Program to perform addition, subtraction, multiplication of two complex numbers using binary operators overloading.

**PROGRAM**

class complex:

def \_\_init\_\_(self, a, b):

self.a = a

self.b = b

def \_\_add\_\_(self, other):

return self.a + other.a, self.b + other.b

def \_\_str\_\_(self):

return self.a, self.b

def \_\_sub\_\_(self, other):

return self.a - other.a, self.b - other.b

def \_\_mul\_\_(self,other):

return self.a \* other.a, self.b \* other.b

def display(self):

if(self[1]>0):

print(self[0],"+",self[1],"j",sep="")

else:

print(self[0],self[1],"j",sep="")

Ob1 = complex(1, 2)

Ob2 = complex(2, 3)

Ob3 = Ob1+Ob2

Ob4 = Ob1-Ob2

Ob5 = Ob1\*Ob2

print("Addition result is ",Ob3)

complex.display(Ob3)

print("Substraction result is ",Ob4)

complex.display(Ob4)

print("Multiplication result is ",Ob5)

complex.display(Ob5)

OUTPUT

Addition result is (3, 5)

3+5j

Substraction result is (-1, -1)

-1-1j

Multiplication result is (2, 6)

2+6j

**LAB PROGRAM 9**

Demonstrate the concept of Method Resolution order in multiple inheritance in Python Program.

**PROGRAM**

import math

class Rectangle:

def calc\_area(length):

breadth=length

area = length \* breadth

print("Area of Rectangle",area)

class Square:

def calc\_area(side):

area = side \* side

print("Area of Square",area)

class Circle:

def calc\_area(radius):

area = math.pi\*radius\*\*2

print("Area of Circle",area)

class Shape(Square,Circle,Rectangle):

pass

Shape.calc\_area(10)

Output

========== RESTART: C:/Users/vandana/Anaconda3/Lib/idlelib/lab9.py ==========

Area of Square 100

**LAB PROGRAM 10**

Create a Python Program to take care of Number Format Exception if user enters values other than integer for calculating average marks of 2 students. The name of the students and marks in 3 subjects are taken from the user while executing the program.

 In the same Program create your own Exception classes to take care of Negative values and values out of range (i.e. other than in the range of 0-100)  Include finally to output the statement ―Program terminated.

**PROGRAM**

class MyException(Exception):

def \_\_init\_\_(self, arg):

self.msg = arg

def acceptMarks(stud\_name):

stud\_marks=list(map(float,input("Enter marks for student separated by space").split()))

for mark in stud\_marks:

if mark < 0 or mark >100:

raise MyException("Entered Mark is negative or outside the range(0-100)")

else:

sum\_marks= sum(stud\_marks)

print(stud\_name,"got total", sum\_marks, "marks")

return sum\_marks

try:

stud1\_name=input("Enter Student1 Name: ")

sum1=acceptMarks(stud1\_name)

stud2\_name=input("Enter Student2 Name: ")

sum2=acceptMarks(stud2\_name)

print("Average marks obtained",(sum1+sum2)/6)

except MyException as ex:

print(ex.msg)

except Exception as ex:

print("Number Format Exception",ex)

finally:-

print("Program terminated")

**OUTPUT**

========== RESTART: C:/Users/vandana/Anaconda3/Lib/idlelib/lab10.py ==========

Enter Student1 Name: vandana

Enter marks for student separated by space12 13 14

vandana got total 39.0 marks

Enter Student2 Name: Hareesh

Enter marks for student separated by space12 15 19

Hareesh got total 46.0 marks

Average marks obtained 14.166666666666666

Program terminated

========== RESTART: C:/Users/vandana/Anaconda3/Lib/idlelib/lab10.py ==========

Enter Student1 Name: vandana

Enter marks for student separated by space12 900

User Defined Exception Entered Mark is negative or outside the range(0-100)

Program terminated

========== RESTART: C:/Users/vandana/Anaconda3/Lib/idlelib/lab10.py ==========

Enter Student1 Name: Hari

Enter marks for student separated by space12 jil

Number Format Exception could not convert string to float: 'jil'

Program terminated

========== RESTART: C:/Users/vandana/Anaconda3/Lib/idlelib/lab10.py ==========

Enter Student1 Name: vandana

Enter marks for student separated by space12 12 -1

User Defined Exception Entered Mark is negative or outside the range(0-100)

Program terminated

**LAB PROGRAM 11**

Design & Implement the program in python to Manipulate images

**PRE-REQUISTIC**

Pillow package must be installed so that it can be imported as PIL in the python script.

>>python -m pip install pillow (On Windows Installation)

>>pip install pillow (On Linux Installation)

**PROGRAM**

from PIL import Image

def crop\_and\_paste():

catCopyIm = catIm.copy()

faceIm = catIm.crop((335, 345, 565, 560))

catCopyIm.paste(faceIm, (0, 0))

catCopyIm.paste(faceIm, (400, 500))

catCopyIm.save('pasted.png')

print("Image cropped and Pasted successfull and saved as pasted.png")

def tiled():

faceIm = catIm.crop((335, 345, 565, 560))

catImWidth, catImHeight = catIm.size

faceImWidth, faceImHeight = faceIm.size

catCopyTwo = catIm.copy()

for left in range(0, catImWidth, faceImWidth):

for top in range(0, catImHeight, faceImHeight):

print(left, top)

catCopyTwo.paste(faceIm, (left, top))

catCopyTwo.save('tiled.png')

print("Image Tiled successfully and saved as tiled.png")

def reduce\_quarter():

width, height = catIm.size

quartersizedIm = catIm.resize((int(width / 2), int(height / 2)))

quartersizedIm.save('quartersized.png')

print("Image resized and saved as quartersized.png")

def rotations():

catIm.rotate(90).save('rotated90.png')

catIm.rotate(6, expand=True).save('rotated6\_expanded.png')

print("Rotation completed and saved as rotated90.png and rotated6\_expanded.png")

def Hor\_Ver\_Flip():

catIm.transpose(Image.FLIP\_LEFT\_RIGHT).save('horizontal\_flip.png')

catIm.transpose(Image.FLIP\_TOP\_BOTTOM).save('vertical\_flip.png')

print("Flipping is done and saved as horizontal\_flip.png and vertical\_flip.png")

catIm = Image.open('zophie.png')

print(catIm.size)

print("Filename = ",catIm.filename,"File Format = ",catIm.format)

choice=1

while choice!=0:

print("0. Exit")

print("1. Crop and Paste")

print("2. Tiling")

print("3. Reduce\_Quarter")

print("4. Rotations")

print("5. Horizontal and Vertical flips")

choice=int(input("Enter choice: "))

if choice==1:

crop\_and\_paste()

elif choice==2:

tiled()

elif choice==3:

reduce\_quarter()

elif choice==4:

rotations()

elif choice==5:

Hor\_Ver\_Flip()

elif choice==0:

print("Exiting!")

else:

print("Invalid choice!!")

print()

///////////////////////////////////////OR////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

from PIL import Image

catIm=Image.open('zophie.png')

print(catIm.size)

print("Filename =",catIm.filename,"File format =",catIm.format)

im=Image.new('RGBA',(100,200),'purple')

im.save('purpleimage.png')

im2=Image.new('RGBA',(20,20))

im2.save('transparentimage.png')

croppedIm=catIm.crop((335,345,565,560))

croppedIm.save('cropped.png')

catCopyIm=catIm.copy()

faceIm=catIm.crop((335,345,565,560))

catCopyIm.paste(faceIm,(0,0))

catCopyIm.paste(faceIm,(400,500))

catCopyIm.save('pasted.png')

catImWidth,catImHeight=catIm.size

faceImWidth,faceImHeight=faceIm.size

catCopyTwo=catIm.copy()

for left in range(0,catImWidth,faceImWidth):

for top in range(0,catImHeight,faceImHeight):

print(left,top)

catCopyTwo.paste(faceIm,(left,top))

catCopyTwo.save('tiled.png')

width,height=catIm.size

quatersizedIm=catIm.resize((int(width/2),int(height/2)))

quatersizedIm.save('quatersized.png')

svelteIm=catIm.resize((width,height+300))

svelteIm.save('svelte.png')

catIm.rotate(90).save('rotated90.png')

catIm.rotate(180).save('rotated180.png')

catIm.rotate(270).save('rotated270.png')

catIm.rotate(6).save('rotated6.png')

catIm.rotate(6,expand=True).save('rotated6\_expanded.png')

catIm.transpose(Image.FLIP\_LEFT\_RIGHT).save('horizontal\_flip.png')

catIm.transpose(Image.FLIP\_TOP\_BOTTOM).save('vertical\_flip.png')

**OUTPUT**

The various image manipulations must be visualized

**LAB PROGRAM 12**

Design & Implement the program in python to handle the events in an Application.

**PRE-REQUISTIC**

**>>python -m pip install pyautogui**

**>>pip install pyautogui**

**PROGRAM**

import pyautogui, time

time.sleep(5)

def rotate():

for i in range(10):

pyautogui.moveTo(100, 100, duration=0.25)

pyautogui.moveTo(200, 100, duration=0.25)

pyautogui.moveTo(200, 200, duration=0.25)

pyautogui.moveTo(100, 200, duration=0.25)

def drag():

time.sleep(5)

pyautogui.click() # click to put drawing program in focus

distance = 200

while distance > 0:

pyautogui.dragRel(distance, 0, duration=0.2) # move right

distance = distance - 5

pyautogui.dragRel(0, distance, duration=0.2) # move down

pyautogui.dragRel(-distance, 0, duration=0.2) # move left

distance = distance - 5

pyautogui.dragRel(0, -distance, duration=0.2) # move up

def takesnapshot():

im = pyautogui.screenshot()

im.save('screenshot.png')

def typing():

time.sleep(5)

pyautogui.typewrite('Hello world how are you!...New HOrizon college of engineering',1)

choice=1

while choice!=0:

print("0. Exit")

print("1. Rotate Mouse")

print("2. Drag in Paint")

print("3. Takesnapshot")

print("4. Autotyping")

choice=int(input("Enter choice: "))

if choice==1:

rotate()

elif choice==2:

drag()

elif choice==3:

takesnapshot()

elif choice==4:

typing()

elif choice==0:

print("Exiting!")

else:

print("Invalid choice!!")

print()

///////////////////////////////////////////////////////////OR////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

import pyautogui

print(pyautogui.size())

for i in range(10):

pyautogui.moveTo(100, 100, duration=0.25)

pyautogui.moveTo(200, 100, duration=0.25)

pyautogui.moveTo(200, 200, duration=0.25)

pyautogui.moveTo(100, 200, duration=0.25)

print(pyautogui.position())

x, y = pyautogui.position()

positionStr = 'X: ' + str(x).rjust(4) + ' Y: ' + str(y).rjust(4)

print(positionStr)

pyautogui.click(15, 20)

import pyautogui, time

time.sleep(5)

pyautogui.click() # click to put drawing program in focus

distance = 200

while distance > 0:

pyautogui.dragRel(distance, 0, duration=0.2) # move right

distance = distance - 5

pyautogui.dragRel(0, distance, duration=0.2) # move down

pyautogui.dragRel(-distance, 0, duration=0.2) # move left

distance = distance - 5

pyautogui.dragRel(0, -distance, duration=0.2) # move up

import time

time.sleep(5)

pyautogui.scroll(-100)

import pyautogui

im = pyautogui.screenshot()

im.save('screenshot.png')

pyautogui.click(100, 100)

pyautogui.typewrite('Hello world how are you!...New HOrizon college of engineering',1)

import pyautogui, time

def commentAfterDelay():

pyautogui.click(100, 100)

pyautogui.typewrite('In IDLE, Alt-3 comments out a line.')

time.sleep(5)

pyautogui.hotkey('alt', '3')

commentAfterDelay()

**OUTPUT**

The mouse and keyboard events are automated with the program and observed.