GLOBAL INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering

3rd Year VIth Semester

Python Programming Lab (2019-20)

Python programming lab:

		If (distance between two balls centers) <= (sum of their radii) then (they are colliding)	
		b) Find mean, median, mode for the given set of numbers in a list.	**15**
9	**Functions -Continued**	**a)** Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.	**15**
		b) Write a function dups to find all duplicates in the list.	**16**
		c) Write a function unique to find all the unique elements of a list.	**16**
10	**Functions - Problem Solving**	**a)** Write a function cumulative_product to compute cumulative product of a list of numbers.	**17**
		b) Write a function reverse to reverse a list. Without using the reverse function.	**18**
		c) Write function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line.	**18**
11	**Multi-D Lists**	**a)** Write a program that defines a matrix and prints	**18**
		b) Write a program to perform addition of two square matrices	**19**
		c) Write a program to perform multiplication of two square matrices	**20**

PYTHON PROGRAMMING PROGRAMES

1.

a. Running instructions in Interactive interpreter and a Python Script?

```
>>> 3+5

8
>>> 4-2

2
>>> 8/2

        4.0
```

b. Write a program to purposefully raise Indentation Error and Correct it

```
n1=int(input("enter n1 value"))
n2=int(input("enter n2 value"))
if n1>n2:
print("n1 is big")
else:
print("n2 is big")
```

Output:

Error: Excepted an indented block

Correct Program:

```
n1=int(input("enter n1 value"))
n2=int(input("enter n1 value"))
if n1>n2:
    print("n1 is big")
else:
    print("n2 is big")
```

Output:

enter n1 value10
enter n1 value20
n2 is big

2.

a. Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)

```
import math;
x1=int(input("Enter x1--->"))
y1=int(input("Enter y1--->"))

x2=int(input("Enter x2--->"))
y2=int(input("Enter y2--->"))

d1 = (x2 - x1) * (x2 - x1);
d2 = (y2 - y1) * (y2 - y1);
res = math.sqrt(d1+d2)
print ("Distance between two points:",res);
```

Output:

Enter x1--->10
Enter y1--->20
Enter x2--->15
Enter y2--->19
Distance between two points: 5.0990195135927845

b. Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

```
step 1: open notepad and write the below program

import sys;
n1=int(sys.argv[1]);
n2=int(sys.argv[2]);
print (n1+n2)

step 2: SAVE

step 3: Open DOS SHELL and go to file saved location (D:\)

step 4:  python filename.py argument1 argument2
```

Output:
python filename.py 10 20
30

3.

a. Write a Program for checking whether the given number is a even number or not.

```
n=int(input("Enter a number --- >"))
if n % 2 == 0:
    print ("EVEN Number");
else:
    print ("ODD Number");
```

Output:

Enter a number--- >10
EVEN Number

Enter a number--- >11
ODD Number

b. Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, ...,1/10

```
i=1;
for j in range(2,10):
    print("i:",i,"j:",j)
    print(i,"/",j)
    print (i/j);
```

Output:

i: 1 j: 2
1 / 2
0.5
i: 1 j: 3
1 / 3
0.3333333333333333
i: 1 j: 4
1 / 4
0.25
i: 1 j: 5
1 / 5
0.2
i: 1 j: 6
1 / 6
0.16666666666666666

```
i: 1 j: 7
1 / 7
0.14285714285714285
i: 1 j: 8
1 / 8
0.125
i: 1 j: 9
1 / 9
0.1111111111111111
```

c. Write a program using a for loop that loops over a sequence. What is sequence ?

```
str="i am python developer"
for i in str:
    print(i)
```

Output:

```
i

a
m

p
y
t
h
o
n

d
e
v
e
l
o
p
e
r
```

d. Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

```
n = int(input("Enter A Number--->"));
while n >=0:
    print (n);
    n = n - 1;
```

Output:

```
Enter A Number--->10
10
9
8
7
6
5
4
3
2
1
0
```

4.

a. Find the sum of all the primes below two million.

Each new term in the Fibonacci sequence is generated by adding the previous two terms.
By starting with 1 and 2, the first 10 terms will be:
1, 2, 3, 5, 8, 13, 21, 34, 55, 89,...

```
a, b = 1, 2
total = 0
print(a,end=" ")
while (a <=2000000-1):
    if a % 2 != 0:
        total += a
    a, b = b, a+b
    print(a,end=" ")
print("\n sum of prime numbers term in fibonacci series: ",total)
```

Output:

1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309
sum of prime numbers term in fibonacci series: 2435422

b. By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

```
a, b = 1, 2
total = 0
print(a,end=" ")
while (a <=4000000-1):
    if a % 2 == 0:
        total += a
    a, b = b, a+b
    print(a,end=" ")
print("\n sum of prime numbers term in fibonacci series: ",total)
```

Output:

1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711 28657 46368 75025 121393 196418 317811 514229 832040 1346269 2178309 3524578 5702887
sum of prime numbers term in Fibonacci series: 4613732

5.

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

```
str=input("Enter a String:")
dict = {}
for n in str:
    keys = dict.keys()
    if n in keys:
        dict[n] += 1
    else:
        dict[n] = 1
print (dict)
```

(OR)

```
str=input("Enter a String")
dict = {}
for i in str:
```

```
    dict[i] = str.count(i)
print (dict)
```

Output:

Enter a String: guntur
{'g': 1, 'u': 2, 'n': 1, 't': 1, 'r': 1}

(OR)

```
str=input("Enter a String")
dist={}
L=len(str);
d={str:L};
print(d)
```

Output:

Enter a Stringguntur
{'guntur': 6}

b. Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.

```
a="hi i am python programmer"
b=a.split()
print (b)
c=" ".join(b)
print(c)
```

Output:

['hi', 'i', 'am', 'python', 'programmer']
hi i am python programmer

6.

a. Write a program combine_lists that combines these lists into a dictionary.

```
l1=[1,2,'cse',4,5]
l2=['khit',7,'cse',9]
l=l1+l2
d={}
for i in range(len(l)):
    d[i]=l[i]
```

```
print("The dictionary is ----->",d)
```

Output:

The dictionary is----- > {0: 1, 1: 2, 2: 'cse', 3: 4, 4: 5, 5: 'khit', 6: 7, 7: 'cse', 8: 9}

----------OR----------

```
l=[1,'python',4,7]
k=['cse',2,'guntur',8]
m=[]
m.extend(l);
m.extend(k);
print(m)
d={1:l,2:k,'combine_list':m}
print(d)
```

Output:

[1, 'python', 4, 7, 'cse', 2, 'guntur', 8]
{1: [1, 'python', 4, 7], 2: ['cse', 2, 'guntur', 8], 'combine_list': [1, 'python', 4, 7, 'cse', 2, 'guntur', 8]}

----------OR----------

```
l=[1,'python',4,7]
k=['cse',2,'guntur',8]
m=[]
m.append(l);
m.append(k);
print(m)
d={1:l,2:k,'combine_list':m}
print(d)
```

Output:

[[1, 'python', 4, 7], ['cse', 2, 'guntur', 8]]
{1: [1, 'python', 4, 7], 2: ['cse', 2, 'guntur', 8], 'combine_list': [[1, 'python', 4, 7], ['cse', 2, 'guntur', 8]]}

b. Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

```
import os
count =0
file=open("D:/a.txt")
for line in file:
    for l in range(0,len(line)):
        count+=1;
print("count:",count)
filename,file_extension=os.path.splitext("D:/a.txt");
```

```
print("file_extension==",file_extension);
if(file_extension=='.py'):
    print("its python program file");
elif(file_extension==".txt"):
    print("its a txt file");
elif(file_==extension==".c"):
rint("its a c program file");
```

Output: a.txt contain "hi"

count: 2
file_extension== .txt
its a txt file

7.

a. Write a program to print each line of a file in reverse order.

```
input_file=open('D:/a.txt','r')
for line in input_file:
    l=len(line)
    s=' '
    while(l>=1):
        s=s+line[l-1]
        l=l-1
    print(s)
input_file.close()
```

a.txt file contain khit

Output:

tihk

b. Write a program to compute the number of characters, words and lines in a file.

```
k=open('D:/a.txt','r')
char,wc,lc=0,0,0
for line in k:
    for k in range(0,len(line)):
        char +=1
        if(line[k]==' '):
            wc+=1
        if(line[k]=='\n'):
            wc,lc=wc+1,lc+1
```

```
print("The no.of chars is %d\n The no.of words is %d\n The
no.of lines is %d"%(char,wc,lc))
```

a.txt contain

khit
Guntur
Hyderabad

Output:

The no. of chars is 21
The no. of words is 2
The no. of lines is 2

8.

a. Write a function ball_collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius
If (distance between two balls centers) <= (sum of their radii) then (they are colliding)

```
import math
def ball_collide(x1,y1,r1,x2,y2,r2):
    dist=math.sqrt((x2-x1)**2+(y2-y1)**2);
    print("Distance b/w two balls:",dist)
    center=dist/2;
    print("Collision point",center);
    r=r1+r2;
    print("Sum of radious",r)
    if(center<=r):
        print("They are Colliding")
        return True;
    else:
        print("Not Colliding")
        return False;

c=ball_collide(4,4,3,2,2,3)
print(c)
```

```
c=ball_collide(100,200,20,200,100,10)
print(c)
```

Output:

Distance b/w two balls: 2.8284271247461903
Collision point 1.4142135623730951
Sum of radious 6
They are Colliding
True
Distance b/w two balls: 141.4213562373095
Collision point 70.71067811865476
Sum of radious 30
Not Colliding
False

b. Find mean, median, mode for the given set of numbers in a list.

```
from statistics import mean,median,mode
l = [15, 18, 2, 36, 12, 78, 5, 6, 9,18]
print("Mean",mean(l))
print("Median",median(l))
print("Mode",mode(l))
```

Output:

Mean 19.9
Median 13.5
Mode 18

9.

a. Write a function nearly_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.

```
from difflib import SequenceMatcher
def Nearly_Equal(a,b):
    return SequenceMatcher(None,a,b).ratio();
a="khit"
b="khitc"
c=Nearly_Equal(a,b)
if(c*100>80):
    print("Both Strings are similar")
    print("a is mutation on b")
```

```
else:
    print("Both Strings are Different")
```

Output:

Both Strings are similar
a is mutation on b

b. Write a function dups to find all duplicates in the list.

```
def FindDuplicates(list):

    for i in list:
        count = list.count(i)
        if count > 1:
            print ('There are duplicates in list')
            return True
    print ('There are no duplicates in list'  )
    return False


a = [8, 64, 16, 32, 4, 24]
b = [2,2,3,6,78,65,4,4,5]
print(a)
FindDuplicates(a)
print(b)
FindDuplicates(b)
```

Output:

[8, 64, 16, 32, 4, 24]
There are no duplicates in list
[2, 2, 3, 6, 78, 65, 4, 4, 5]
There are duplicates in list

c. Write a function unique to find all the unique elements of a list.

```
def FindUnique(list):
    unique = set(list)
    for i in unique:
        count = list.count(i)
        if count > 1:
            print ('There are no unique elements in list')
```

```
            return True
    print ('There are unique elements in list'  )
    return False


a = [8, 64, 16, 32, 4, 24]
b = [2,2,3,6,78,65,4,4,5]
print(a)
FindUnique(a)
print(b)
FindUnique(b)
```

Output:

[8, 64, 16, 32, 4, 24]
There are unique elements in list
[2, 2, 3, 6, 78, 65, 4, 4, 5]
There are no unique elements in list

10.

a. Write a function cumulative_product to compute cumulative product of a list of numbers.

```
def product(list):
    p =1
    for i in list:
        p *= i
        print(p)
    return p
arr= [1,2,3,4,5,6,7,8,9,10]
c=product(arr)
```

Output:

1
2
6
24
120
720
5040
40320
362880
3628800

b. Write a function reverse to reverse a list. Without using the reverse function.

```
l = [1,2,3,4,5]
print (l[::-1])
```

Output:

[5, 4, 3, 2, 1]

c. Write function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line.

```
import fractions
n1 = int(input("Enter n1 value:"))
n2 = int(input("Enter n2 value:"))
gcd = fractions.gcd(n1, n2)
print("GCD value is:",gcd)
def lcm(n, m):
    return n * m / gcd

print("LCM value is:",int(lcm(n1,n2)))
```

Output:

Enter n1 value: 5
Enter n2 value: 9
GCD value is: 1
LCM value is: 45

11.

a. Write a program that defines a matrix and prints

```
row=int(input("Enter No of Rows for 1st Matrix:"))
column=int(input("Enter No of column for 1nd Matrix:"))
row1=int(input("Enter No of Rows for 2st Matrix:"))
column1=int(input("Enter No of column for 2nd Matrix:"))
X = [[int(input(("Enter value for X[",i,"][",j,"]:"))) for j
in range(column)] for i in range(row)]
Y = [[int(input(("Enter value for Y[",i,"][",j,"]:"))) for j
in range(column1)] for i in range(row1)]
```

```
print("1st Matrix X:",X)
print("2st Matrix Y:",Y)
```

Output:

```
Enter No of Rows for 1st Matrix: 3
Enter No of column for 1nd Matrix: 3
Enter No of Rows for 2st Matrix: 3
Enter No of column for 2nd Matrix: 3
('Enter value for X[', 0, '][', 0, ']:')5
('Enter value for X[', 0, '][', 1, ']:')8
('Enter value for X[', 0, '][', 2, ']:')2
('Enter value for X[', 1, '][', 0, ']:')1
('Enter value for X[', 1, '][', 1, ']:')9
('Enter value for X[', 1, '][', 2, ']:')2
('Enter value for X[', 2, '][', 0, ']:')3
('Enter value for X[', 2, '][', 1, ']:')4
('Enter value for X[', 2, '][', 2, ']:')5
('Enter value for Y[', 0, '][', 0, ']:')6
('Enter value for Y[', 0, '][', 1, ']:')0
('Enter value for Y[', 0, '][', 2, ']:')1
('Enter value for Y[', 1, '][', 0, ']:')4
('Enter value for Y[', 1, '][', 1, ']:')2
('Enter value for Y[', 1, '][', 2, ']:')5
('Enter value for Y[', 2, '][', 0, ']:')6
('Enter value for Y[', 2, '][', 1, ']:')7
('Enter value for Y[', 2, '][', 2, ']:')8
1st Matrix X: [[5, 8, 2], [1, 9, 2], [3, 4, 5]]
2st Matrix Y: [[6, 0, 1], [4, 2, 5], [6, 7, 8]]
```

b. Write a program to perform addition of two square matrices

```
row=int(input("Enter No of Rows for 1st Matrix:"))
column=int(input("Enter No of column for 1nd Matrix:"))
row1=int(input("Enter No of Rows for 2st Matrix:"))
column1=int(input("Enter No of column for 2nd Matrix:"))
X = [[int(input(("Enter value for X[",i,"][",j,"]:"))) for j
in range(column)] for i in range(row)]
Y = [[int(input(("Enter value for Y[",i,"][",j,"]:"))) for j
in range(column1)] for i in range(row1)]
print("1st Matrix X:",X)
print("2st Matrix Y:",Y)
if (row==row1 and column==column1):
```

```
    result = [[X[i][j] + Y[i][j]  for j in range(len(X))] for
i in range(len(X[0]))]
    print(result)
else:
    print("Adition 2 Matrix not Possible")
```

Output:

```
Enter No of Rows for 1st Matrix: 2
Enter No of column for 1nd Matrix: 2
Enter No of Rows for 2st Matrix: 2
Enter No of column for 2nd Matrix: 2
('Enter value for X[', 0, '][', 0, ']:')1
('Enter value for X[', 0, '][', 1, ']:')2
('Enter value for X[', 1, '][', 0, ']:')3
('Enter value for X[', 1, '][', 1, ']:')4
('Enter value for Y[', 0, '][', 0, ']:')5
('Enter value for Y[', 0, '][', 1, ']:')6
('Enter value for Y[', 1, '][', 0, ']:')7
('Enter value for Y[', 1, '][', 1, ']:')8
1st Matrix X: [[1, 2], [3, 4]]
2st Matrix Y: [[5, 6], [7, 8]]
[[6, 8], [10, 12]]
```

c. Write a program to perform multiplication of two square matrices

```
row=int(input("Enter No of Rows for 1st Matrix:"))
column=int(input("Enter No of column for 1nd Matrix:"))
row1=int(input("Enter No of Rows for 2st Matrix:"))
column1=int(input("Enter No of column for 2nd Matrix:"))
if(column==row1):
    X = [[int(input(("Enter value for X[",i,"][",j,"]:"))) for
j in range(column)] for i in range(row)]
    Y = [[int(input(("Enter value for Y[",i,"][",j,"]:"))) for
j in range(column1)] for i in range(row1)]
    result = [[0 for j in range(column1)] for i in range(row)]
    print("result",result)
    print("1st Matrix X:",X)
    print("2st Matrix Y:",Y)
    for i in range(len(X)):
         for j in range(len(Y[0])):
             for k in range(len(Y)):
               result[i][j] += X[i][k] * Y[k][j]

    for r in result:
       print(r)
```

```
else:
    print("Multiplication is not possible")
```

Output:

```
Enter No of
Rows for 1st
Matrix:2 Enter
No of column
for 1nd
Matrix:2 Enter
No of Rows for
2st Matrix:2
Enter No of
column for 2nd
Matrix:2
('Enter value
for X[', 0, ']['
0, ']:')4
('Enter value for X[', 0, '][', 1, ']:')3
('Enter value for X[', 1, '][', 0, ']:')5
('Enter value for X[', 1, '][', 1, ']:')6
('Enter value for Y[', 0, '][', 0, ']:')2
('Enter value for Y[', 0, '][', 1, ']:')1
('Enter value for Y[', 1, '][', 0, ']:')7
('Enter value for Y[', 1, '][', 1, ']:')3
result [[0, 0], [0, 0]]
1st Matrix X: [[4, 3], [5, 6]]
2st Matrix Y: [[2, 1], [7, 3]]
[29, 13]
[52, 23]
```