**Experiment No.: 1**

**Aim**

Familiarizing Text Editor, IDE, Code Analysis Tools etc//Use any IDE like PyCharm.

PyCharm is the most popular IDE for python and including great features such as excellent code completion and inspection with advanced debugger and support for web programming and various frame works. PyCharm is created by Czech company, Jet brain which focuses on creating integrated development environment for various web development language like JavaScript and PHP.

**Experiment No.: 2**

**Aim**

Display future leap years from current years to a final year entered by the user.

**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

f=int(input("Enter the current year:"))  
l=int(input("Enter the last year:"))  
for i in range(f,l):  
 if i%4==0 and i%100!=0 or i%400==0:  
 print(i)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 3**

**Aim**

List comprehension:

1.Find square of n numbers

2. Form a list of vowels selected from a given word.

3. Generate positive list of numbers from given list of integers.

**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

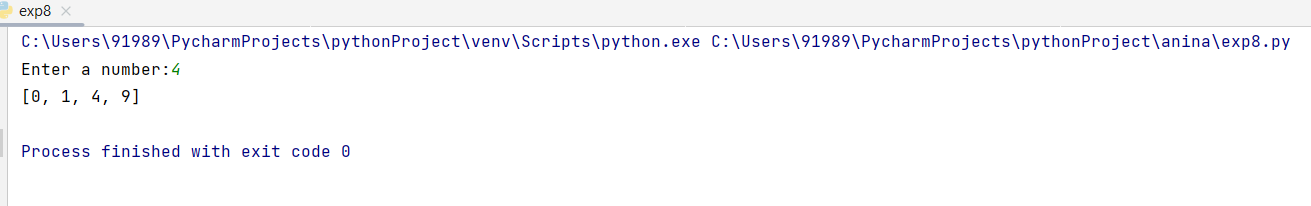
**Procedure**

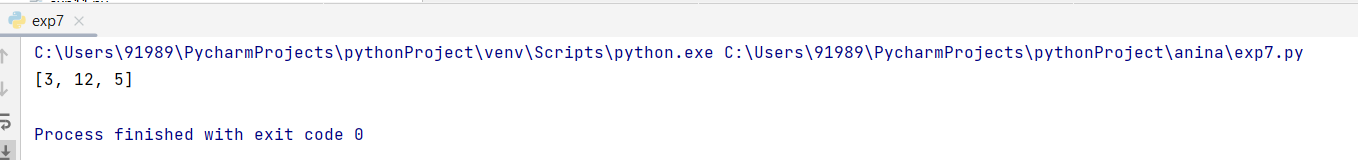
**1.**x=int(input("Enter a number:"))  
square=[i\*i for i in range(x)]  
print(square)

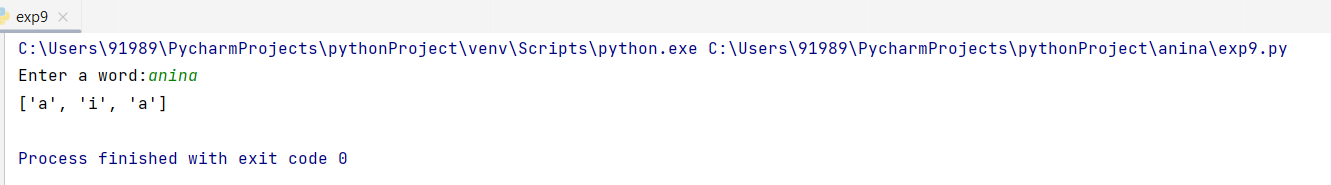
**2.** x=input("Enter a word:")  
y=[letter for letter in x if letter in 'aeiouAEIOU']  
print(y)

**3.** li1=[3,12,5,-5,-3,-24]  
li2=[n for n in li1 if(n>0)]  
print(li2)

**Output Screenshot**





****

**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.:4**

**Aim**

Count the occurrence of each word in a line of text.

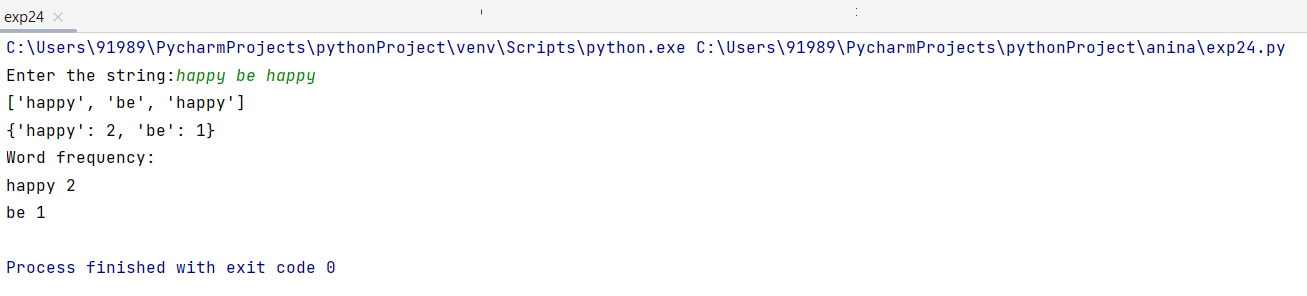
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

str=input("Enter the string:")  
dic={}  
a=str.split()  
print(a)  
for i in a:  
 if i in dic:  
 dic[i]+=1  
 else:  
 dic[i] = 1  
print(dic)  
print("Word frequency:")  
for x,y in dic.items():  
 print(x,y)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 5**

**Aim**

Prompt the user for a list of integers. For all values greater than 100, store the word ‘over’ instead of the value.

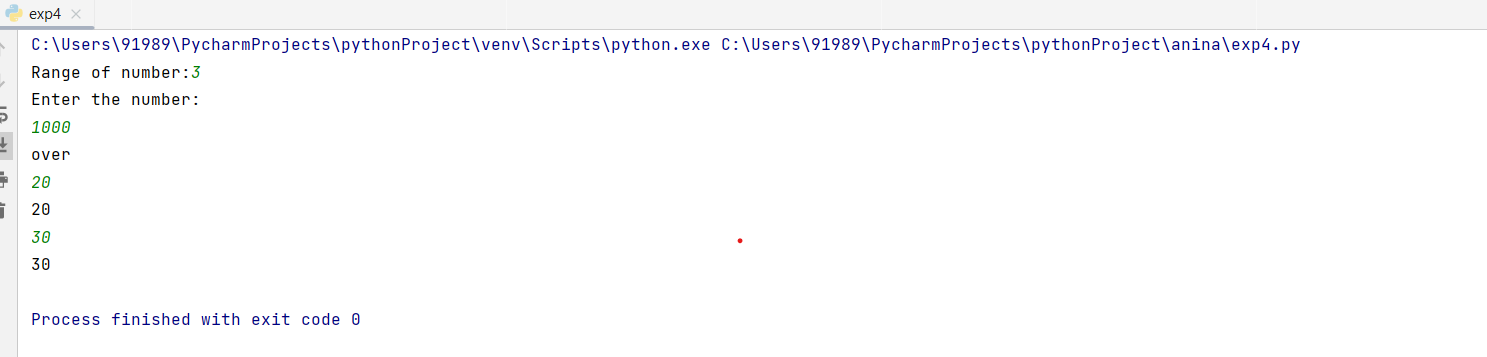
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

x=int(input("Range of number:"))  
print("Enter the number:")  
for i in range(0,x):  
 a=int(input())  
 if (a > 100):  
 a = "over"  
 print(a)  
 else:  
 print(a)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 6**

**Aim**

Store a list of first names. Count the occurrence of ‘a’ within the list.

**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

a=int(input("Enter the number of string:"))  
print("Enter the string:")  
li=[]  
count=0  
for i in range(0,a):  
 el=input()  
 li.append(el)  
print(li)  
for i in li:  
 for j in i:  
 if(j=='a'):  
 count=count+1  
print(count)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 7**

**Aim**

Enter two lists of integers. Check (a) Whether list are of same length (b) Whether list sums to same value. (c) Whether any value occur in both.

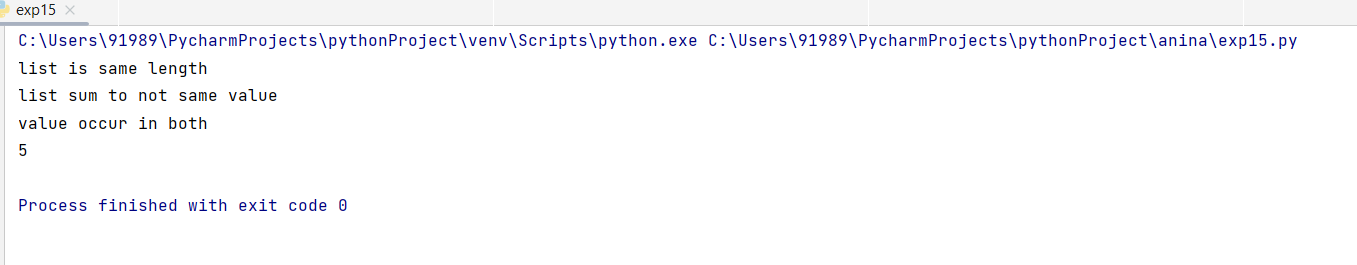
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

x=[10,11,5]  
y=[2,5,8]  
a=len(x)  
b=len(y)  
if a==b:  
 print("list is same length")  
else:  
 print("not a same length")  
if(sum(x)==sum(y)):  
 print("list sum to same value")  
else:  
 print("list sum to not same value")  
print("value occur in both")  
for i in x:  
 if i in y:  
 print(i)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.:8**

**Aim**

Get a string from an input string where all occurrence of first character replaced with “$” expect first character.[eg:onion->oni$n]

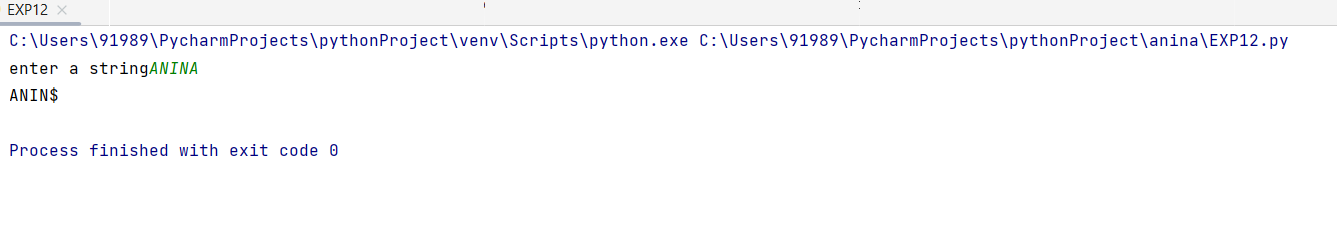
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

x=input("Enter a string:")  
a=x[0]  
for i in x:  
 if i==a:  
 x=x.replace(i,'$')  
 x=a+x[1:]  
print(x)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 9**

**Aim**

Create a string from given string where first and last characters are exchanged. [eg: python->nythop]

**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

x=input("enter a string:")

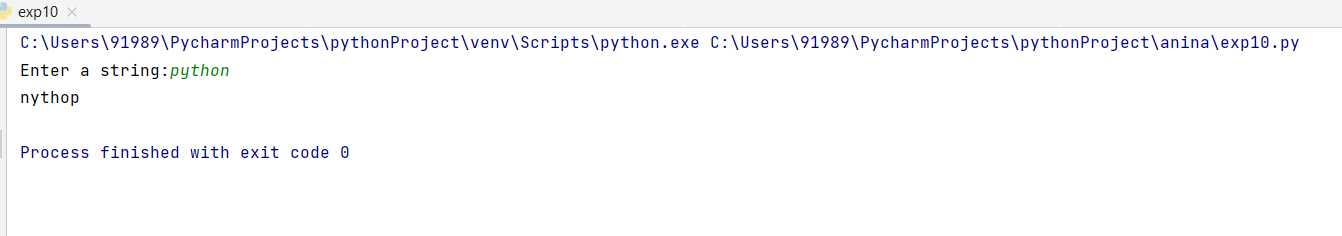
a=x[-1]

b=x[1:-1]

c=x[0]

print(a+b+c)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 10**

**Aim**

Accept the radius from user and find the area of a circle.

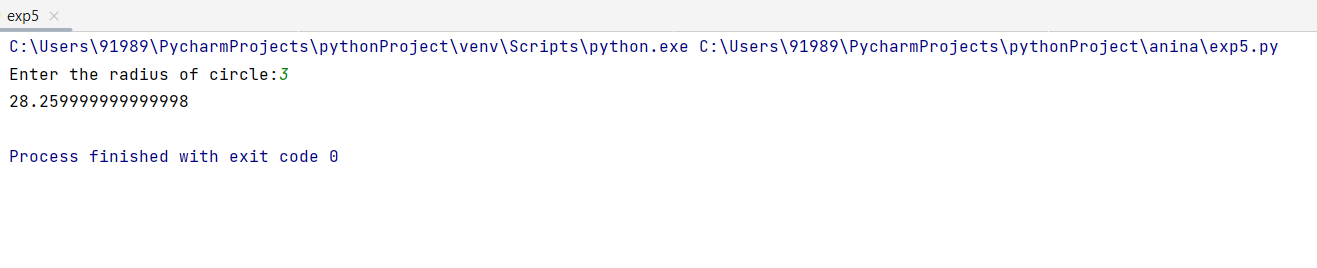
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

r=int(input("Enter the radius of circle:"))  
area=3.14\*r\*r  
print(area)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 11**

**Aim**

Find biggest of three numbers entered.

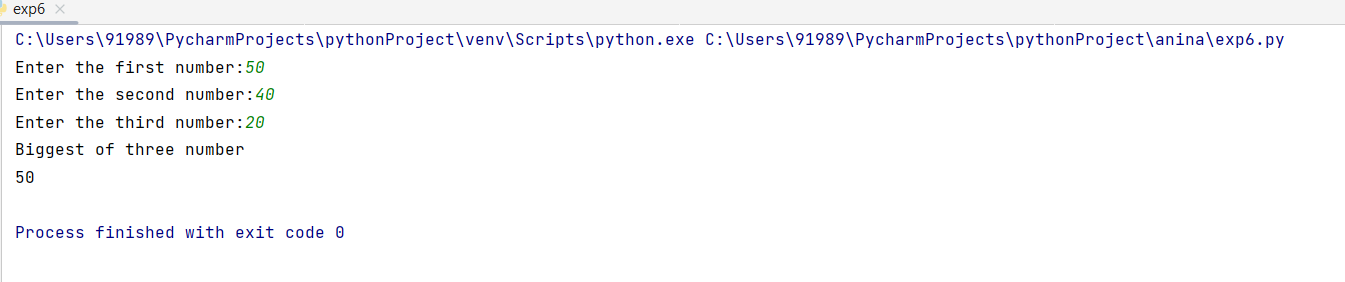
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

x=int(input("Enter the first number:"))  
y=int(input("Enter the second number:"))  
z=int(input("Enter the third number:"))  
print("Biggest of three number")  
if((x>y)and(x>z)):  
 print(x)  
if((y>x)and(y>z)):  
 print(y)  
if((z>x)and(z>y)):  
 print(z)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 12**

**Aim**

Accept a file name from user and print extension from that.

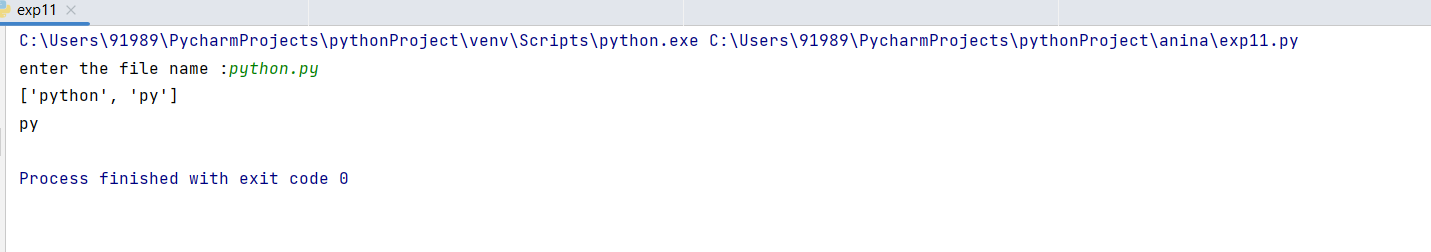
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

f=input("Enter the file name :")  
text=f.split(".")  
print(text)  
print(text[1])

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.:13**

**Aim**

Create a list of colours from comma-separated colour names entered by user. Display the first and last colours.

**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

color=input("Enter the color:")  
print(color)  
color\_list=color.split(',')  
print(color\_list)  
print(color\_list[0])  
print(color\_list[-1])

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 14**

**Aim**

Print out all colours from colour list 1 not contains in colour list 2.

**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

list1=[]  
n = int(input("Enter number of elements : "))  
for i in range(0, n):  
 i = input()  
 list1.append(i) # adding the element  
print(list1)  
list2=[]  
k = int(input("Enter number of elements : "))  
for i in range(0, k):  
 i = input()  
 list2.append(i) # adding the element  
print(list2)  
print("colours present in list 1 but not in list 2 are:")  
c=list(set(list1).difference(list2))  
print(c)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.:15**

**Aim**

Create a single string separated with space from two strings by swapping the character at position 1.

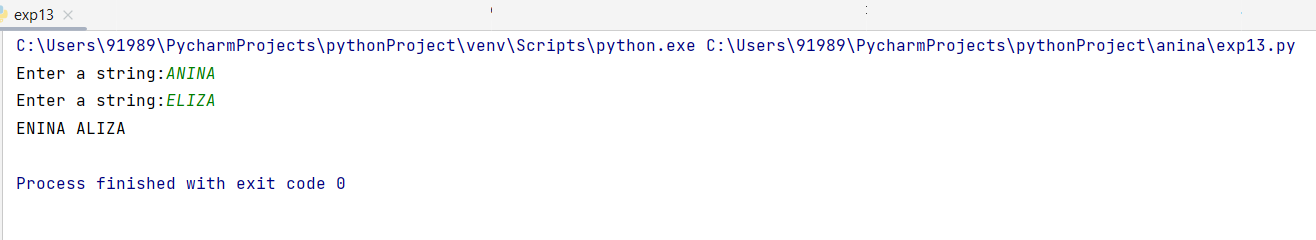
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

x=input("Enter a string:")  
y=input("Enter a string:")  
a=x[0]  
b=y[0]  
new\_x=b+x[1:]  
new\_y=a+y[1:]  
print(new\_x+ " "+new\_y)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 16**

**Aim**

Sort dictionary in ascending and descending orders.

**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

d={1:4,9:5,6:7}

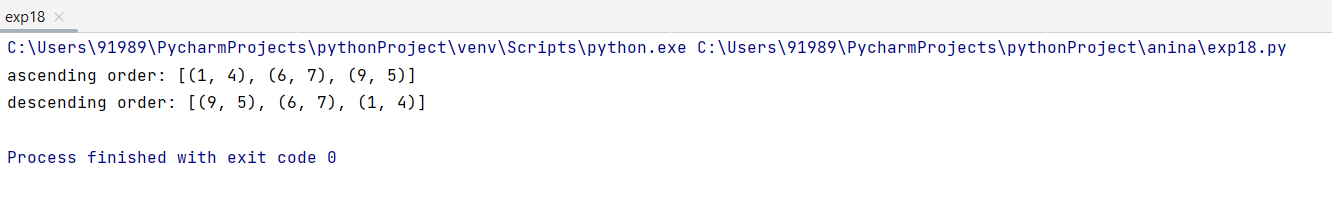
sort\_d=sorted(d.items())

print("ascending order:",sort\_d)

sort\_d=sorted(d.items(),reverse=True)

print("descending order:",sort\_d)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 17**

**Aim**

Merge two dictionaries.

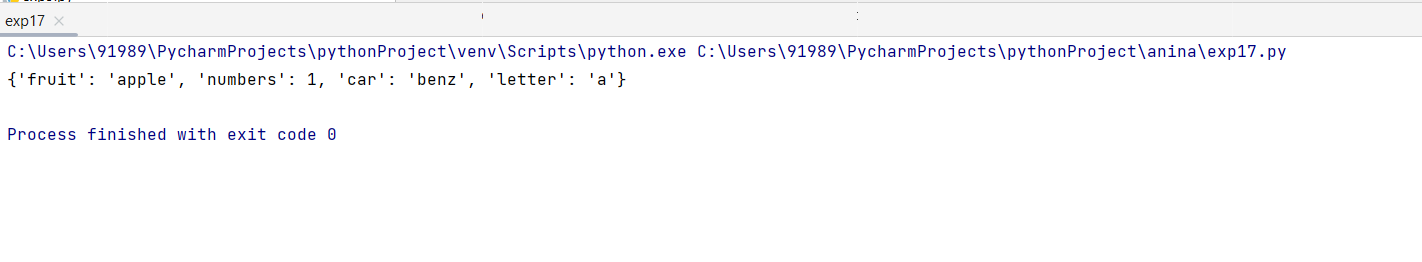
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

d1={'fruit':'apple','numbers':1,'car':'benz'}  
d2={'letter':'a'}  
d1.update(d2)  
print(d1)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 18**

**Aim**

Find GCD of two number.

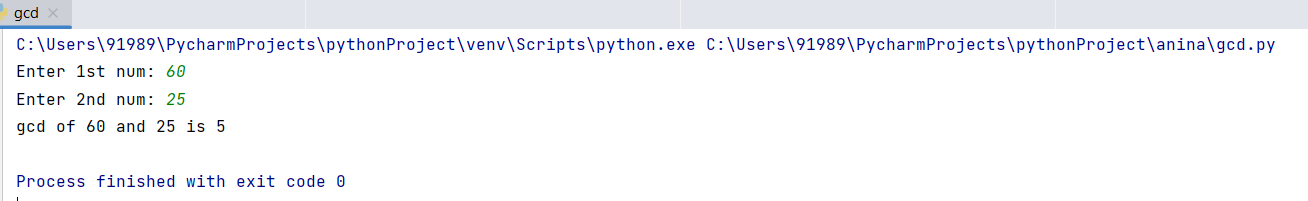
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

x=int(input("Enter 1st num: "))  
y=int(input("Enter 2nd num: "))  
gcd=1  
for i in range(1,max(x,y)):  
 if x%i==0 and y%i==0:  
 gcd=i  
print("gcd of",x,"and",y,"is",gcd)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 19**

**Aim**

From a list of integers, create a list removing even numbers.

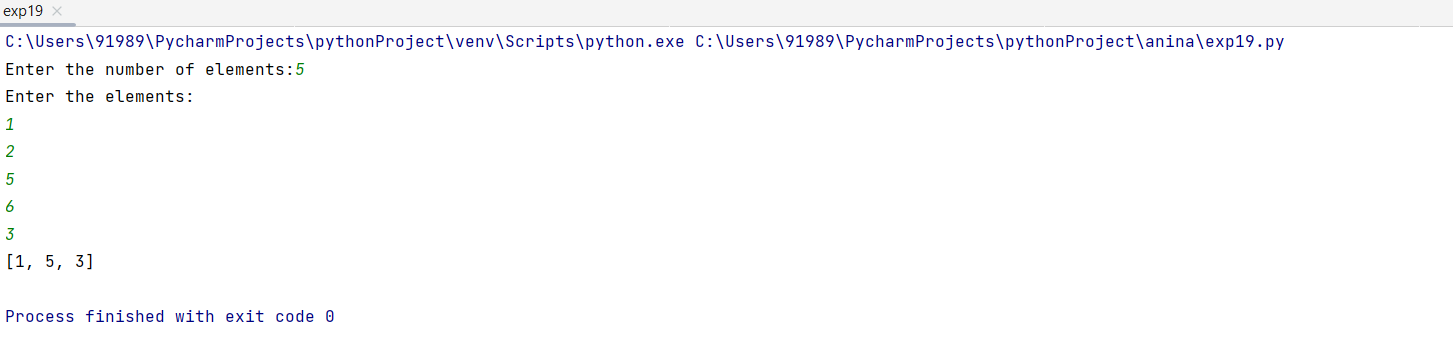
**CO1**

Understand basics of python programming language including input/output functions, operators, basic and collection data types.

**Procedure**

n=int(input("Enter the number of elements:"))  
print("Enter the elements:")  
li=[]  
for i in range(0,n):  
 el=int(input())  
 if( el%2!=0):  
 li.append(el)  
print(li)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO1 was obtained.

**Experiment No.: 20**

**Aim**

Program to find the factorial of a number.

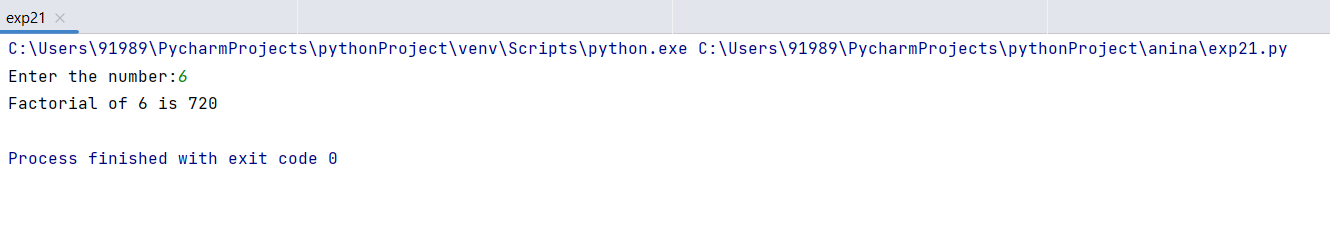
**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

x=int(input("Enter the number:"))  
fact=1  
i=1  
while (i <= x):  
 fact = fact \* i  
 i = i + 1  
print("Factorial of", x, "is", fact)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 21**

**Aim**

Generate Fibonacci series of N terms.

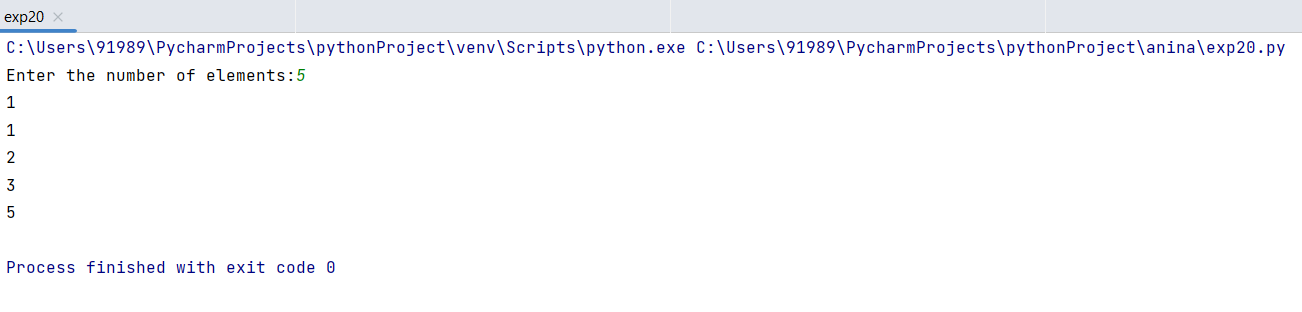
**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

n=int(input("Enter the number of elements:"))  
a=0  
b=1  
sum=0  
if(n<=0):  
 print("Enter a valid number:")  
else:  
 for i in range(0,n):  
 a=b  
 b=sum  
 sum=a+b  
 print(sum)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 22**

**Aim**

Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.

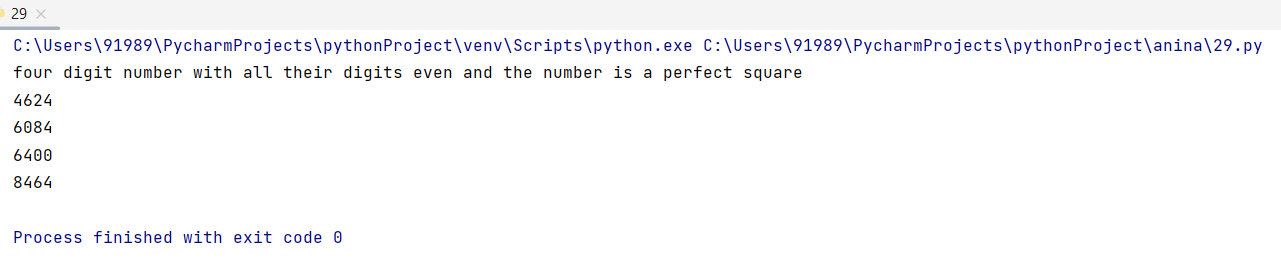
**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

print("four digit number with all their digits even and the number is a perfect square")  
for i in range(1000,10000,1):  
 for j in range(32,100,1):  
 if i==j\*j:  
 string=str(i)  
 if int(string[0])%2==0 and int(string[1])%2==0and int(string[2])%2==0and int(string[3])%2==0:  
 print(i)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 23**

**Aim**

Display the given pyramid with step number accepted from user.

Eg:N=4

1

2 4

3 6 9

4 8 12 16

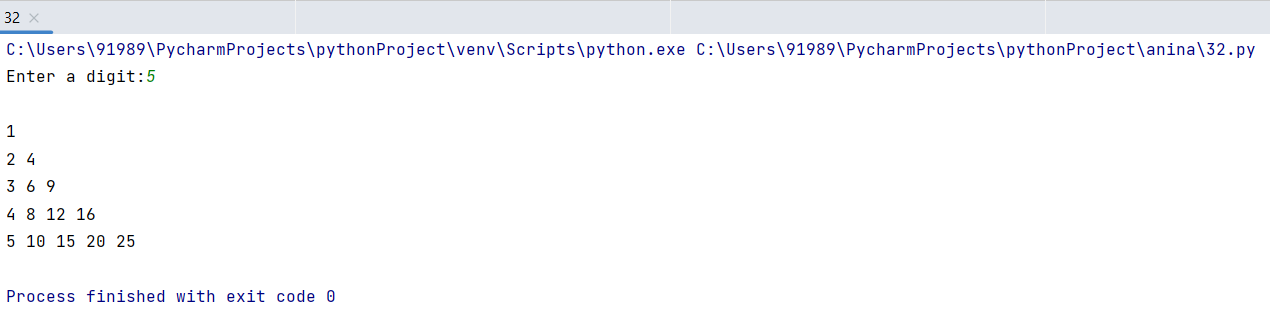
**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

n=int(input("Enter a digit:"))  
for i in range(0,n+1):  
 for j in range(1,i+1):  
 j=i\*j  
 print(j,end=" ")  
 print(" ")

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 24**

**Aim**

Count the number of characters (character frequency) in a string.

**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

str=input("Enter the string:")  
dic={}  
for n in str:  
 if n in dic:  
 dic[n]+=1  
 else:  
 dic[n]=1  
print(dic)  
print("Character frequency:")  
for x,y in dic.items():  
 print(x,y)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 25**

**Aim**

Add ‘ing’ at the end of a given string. If it already ends with ing then add ’ly’.

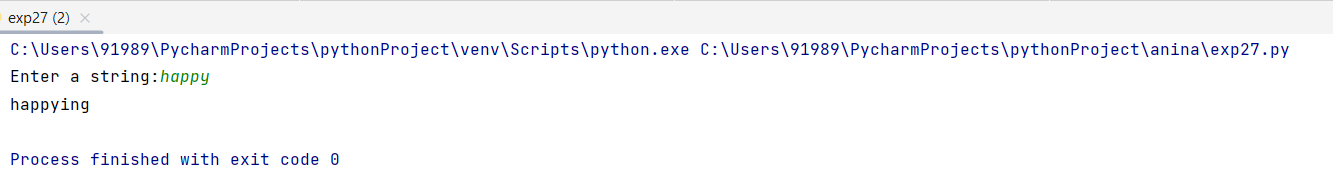
**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

str = input("Enter a string:")  
if (str[-3:] == "ing"):  
 str=str.replace(str[-3:],"ly")  
else:  
 str=str+'ing'  
print(str)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 26**

**Aim**

Accept a list of words and return length of longest word using function.

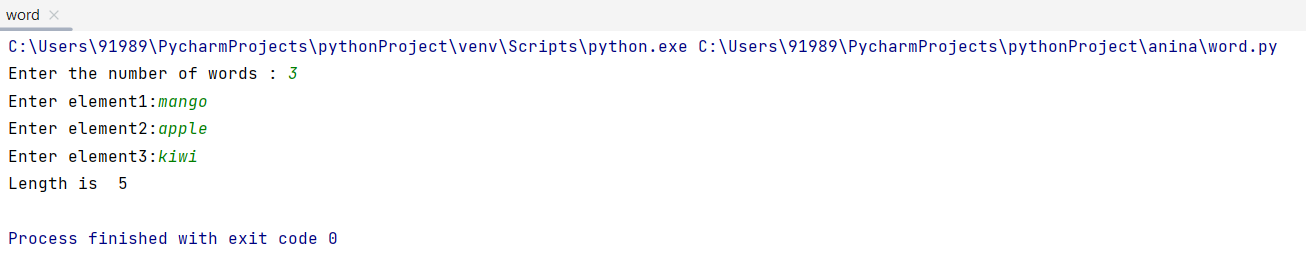
**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

def longestLength(a):  
 max1 = len(a[0])  
 for i in a:  
 if (len(i) > max1):  
 max1 = len(i)  
 print("Length is ", max1)  
  
a=[]  
n =int(input("Enter the number of words : "))  
for j in range (0,n):  
 i=input("Enter element"+str(j+1)+":")  
 a.append(i)  
  
longestLength(a)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 27**

**Aim**

Construct following pattern using nested loop.

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

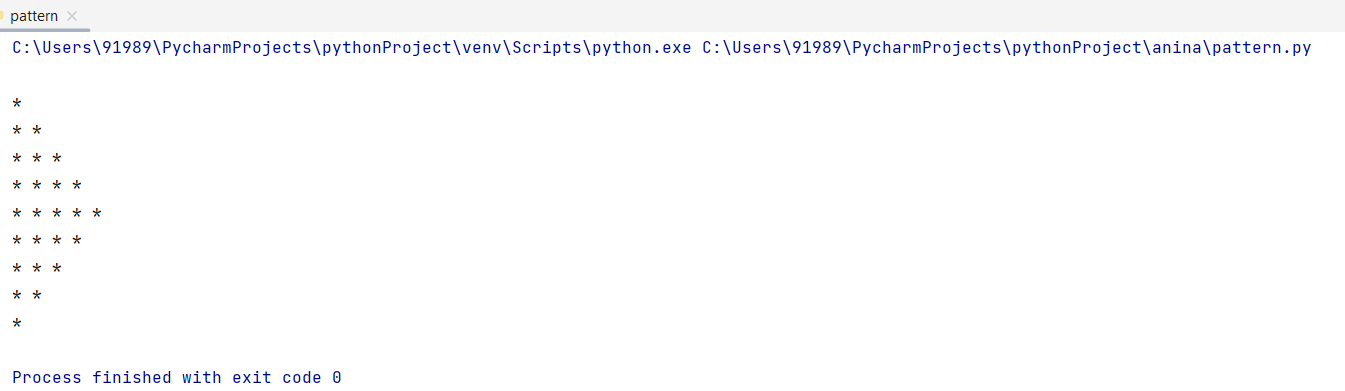
**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

for i in range(0,5):  
 for j in range(i):  
 print ('\* ', end="")  
 print('')  
  
for i in range(5,0,-1):#reverse  
 for j in range(i):  
 print('\* ', end="")  
 print('')

**Output Screenshot**

****

**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 28**

**Aim**

Generate all factors of a number.

**CO2**

Implement decision making, looping constructs and functions.

**Procedure**

def findfact(x):  
 print("The factors of", x, "are:")  
 for i in range(1, x + 1):  
 if x % i == 0:  
 print(i)  
print("Enter the number:")  
num = int(input())  
findfact(num)

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO2 was obtained.

**Experiment No.: 29**

**Aim**

Create a package graphics with modulus rectangle, circle and sub-package 3d graphics with modulus cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write program that find area and perimeter of figures by different importing statement. (Include selective import of modules and import \*statements).

**CO3**

Design modules and packages-built in and user defined packages.

**Procedure**

**Main.py**

from graphics import circle,rect  
rect.area(10,20)  
rect.perim(10,20)  
circle.area(6)  
circle.perim(6)  
from graphics.graphics3d import cub,sphere  
cub.area(2,5,9)  
cub.perim(2,5,9)  
sphere.area(6)  
sphere.perim(6)

**Graphics**

**Rect.py**

def area(l,w):  
 print("area of rectangle ",l\*w)  
def perim(l,w):  
 print("perimeter of rectangle ",2\*(l+w))

**Circle.py**

from math import\*  
def area(r):  
 print("area of circle ",pi\*r\*r)  
def perim(r):  
 print("perimeter of circle ",2\*pi\*r)

**Graphics3d**

**Sphere.py**

from math import\*  
def area(r):  
 print("area of sphere ",4\*pi\*r\*r)  
def perim(r):  
 print("perimeter of sphere ",2\*pi\*r)

**Cub.py**

def area(l,b,h):  
 print("area of cuboid ",2\*(l\*b)+(b\*h)+(h\*l))  
def perim(l,b,h):  
 print("perimeter of cuboid ",4\*(l+b+h))

**Output Screenshot**



**Result**

The program was executed and the result was successfully obtained. Thus, CO3 was obtained.