**Program :1. a) /\*Develop a Python program to find the H.C.F of two input numbers.\*/**

def hcf(x,y):

if x>y:

smaller=y

else:

smaller=x

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

if(x%i==0)and(y%i==0):

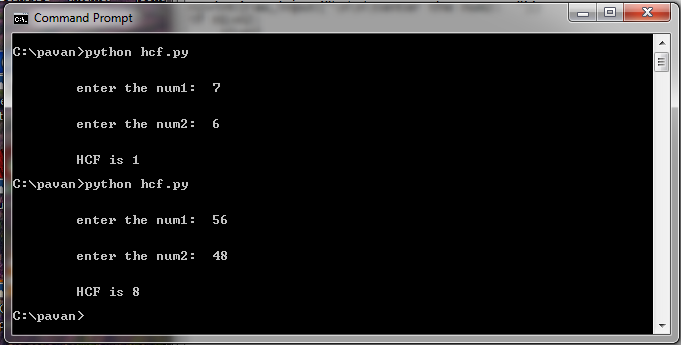
hcf=i

return hcf

x=int(raw\_input('enter the num1'))

y=int(raw\_input('enter the num2'))

print 'HCF is,"\n",hcf(x,y)

**OUTPUT :** 

**1 .b)/\* Design a Python Program to find the L.C.M. of two input numbers.\*/**

def lcm(x,y):

if x>y:

greater=x

else:

greater=y

while True:

if((greater%x==0)and(greater%y==0)):

lcm=greater

break

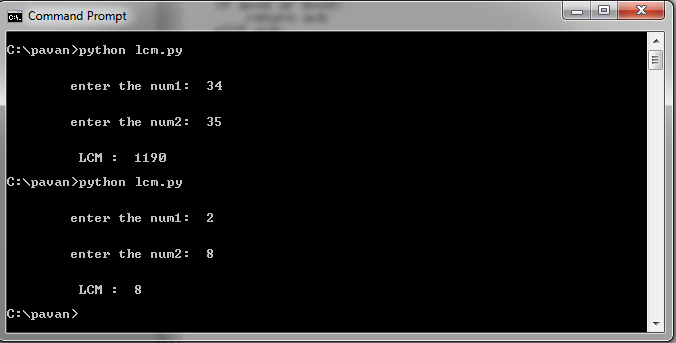
greater+=1

return lcm

a=int(raw\_input('enter num1'))

b=int(raw\_input('enter num2'))

print 'LCM :',lcm(a,b)

**OUTPUT :** 

**Program :2. a)/\* Write a Python Program to find factorial of a number.\*/**

def recurse\_factorial(n):

if n==1:

return n

else:

return n\*recurse\_factorial(n-1)

num=int(input('Enter a Number to find the Factorial: '))

if num<0:

print "sorry,factorial does not exist for negative number"

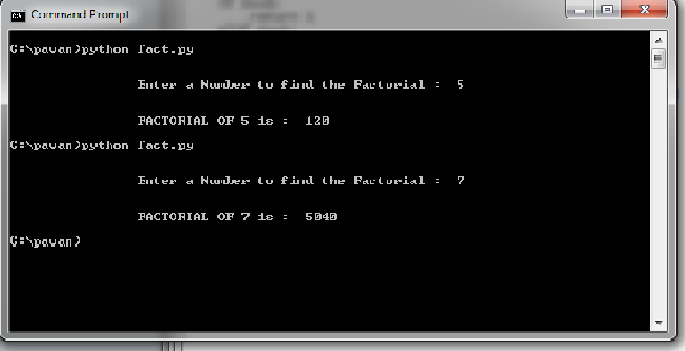
elif num==0:

print "The factorial of 0 is 1"

else:

print " Factorial OF",num,"is:”,recurse\_factorial(num)

**OUTPUT :**



**Program : 2. b) /\*Develop Python Program to make a simple calculator that can add, subtract, multiply and divide.\*/**

def add(a,b):

sum=a+b

return sum

def sub(a,b):

diff=a-b

return diff

def mul(a,b):

prod=a\*b

return prod

def div(a,b):

try:

q=a/b

return q

except:

print'Denominator is 0'

print'1.Addtion\n 2.Subtraction\n 3.Multiplication\n 4.Dividtion\n'

ch=raw\_input('ENT'ER YOUR CHOICE’)

print'ENTER YOUR NUM\_1:'

num1=raw\_input()

print'ENTER YOUR NUM\_2:'

num2=raw\_input()

try:

a=float(num1)

b=float(num2)

if ch=='1':

res=add(a,b)

print'Sum =',res

if ch=='2':

res=sub(a,b)

print'The difference =:',res

if ch=='3':

res=mul(a,b)

print'The product =:',res

if ch=='4':

res=div(a,b)

print'The quotient =:',res

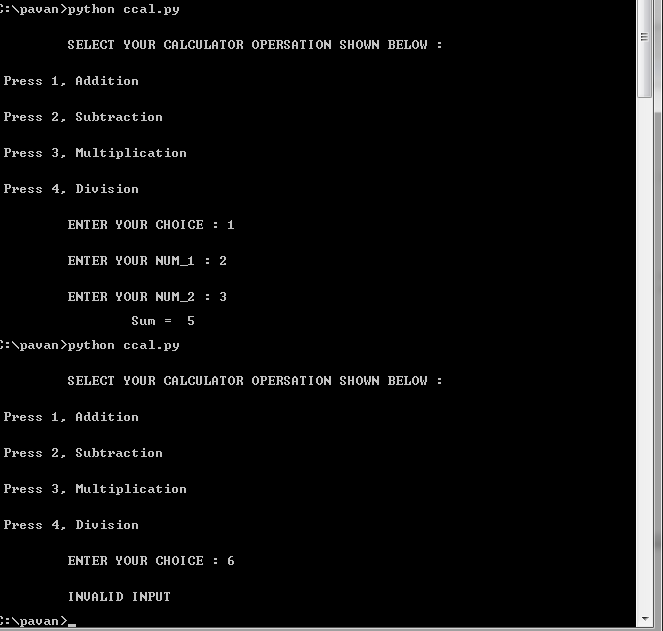
else:

print'INVALID INPUT’'

except:

print'INVALID INPUT’

**OUTPUT :**



**Program : 3. a)/\* Write a python program to display the Fibonacci sequence up to nth term where n is provided by the user.\*/**

n=int(raw\_input('Enter a number:'))

a=0

b=1

if n==0:

print a

elif n==1:

print b

else:

print a

print b

for i in range(3,n+1):

k=a+b

print k

a=b

b=k

**OUTPUT :**



**Program : 3.b)/\* Develop a Python program to find sum of natural numbers up to n where n will be provided by user.\*/**

def sum\_n(n):

if n==0:

return 0

else:

sum=0

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

sum=sum+i

return sum

num=raw\_input('Enter a number:')

try:

n=int(num)

if n<0:

print'Enter only +ve number'

else:

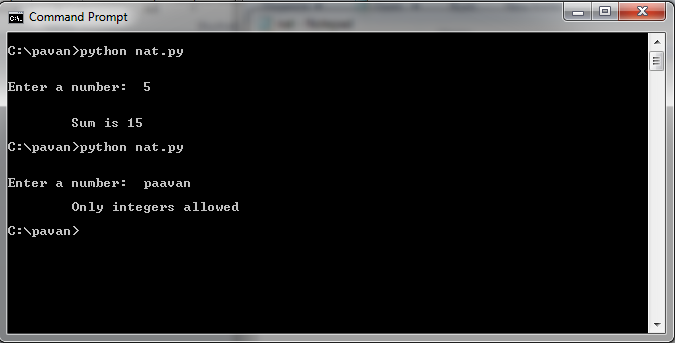
sum=sum\_n(n)

print'Only integers allowed',sum

except:

print'Invalid input'

**OUTPUT :**



**Program : 4. a) /\*Design a Python program to count each vowel in a given string .\*/**

ipstr=raw\_input('Enter a string:')

ipstr=ipstr.lower()

count=0

count\_a=0

count\_e=0

count\_i=0

count\_o=0

count\_u=0

for char in ipstr:

if char=='a':

count\_a+=1

elif char=='e':

count\_e+=1

elif char=='i':

count\_i+=1

elif char=='o':

count\_o+=1

elif char=='u':

count\_u+=1

else:

continue

count=count\_a+count\_e+count\_i+count\_o+count\_u

print 'count of vowel:\n'

print 'a:',count\_a

print 'e:',count\_e

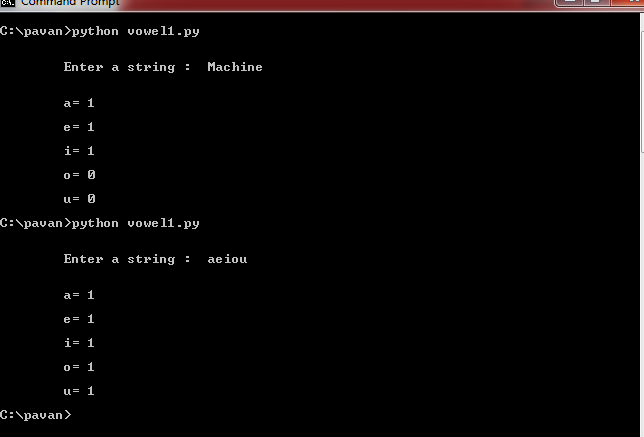
print 'i:',count\_i

print 'o:',count\_o

print 'u:',count\_u

print 'Total count:',count

**OUTPUT :**



**Program : 4. b) /\*Program to check if a string is palindrome or not.\*/**

try:

mystr=raw\_input("enter a string:")

mystr=mystr.lower()

revstr=mystr.lower()

revstr=reversed(mystr)

if list(mystr)==list(revstr):

print “It is Palindrome"

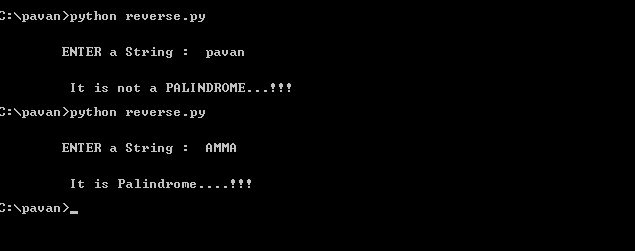
else:

print "it is not PALINDROME…!!!"

except:

print 'Invalid input'

**OUTPUT :**



**Program :5. a) /\*Develop Python program to check if the input number is prime or not.\*/**

try:

n=int(raw\_input('Enter a NUMBER TO CHECK or NOT: '))

i=2

while i<=n/2:

if n%i==0:

print n ,’ is not prime number'

break

i=i+1

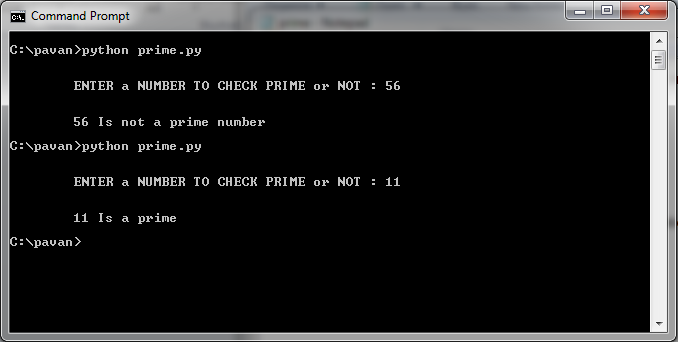
else:

print n , is a 'prime '

except:

print 'enter natural number'

**OUTPUT :**



**Program :5. b)/\* Write a python program to compute roots of a quadratic equation.\*/**

import math

try:

a=int(raw\_input(“ENTER the Value of a: '))

b=int(raw\_input(' ENTER the Value of b:'))

c=int(raw\_input(' ENTER the Value of c: '))

d=(b\*b)-(4\*a\*c)

if d==0:

print 'Roots are equal\n'

r1=r2=-b/(2\*a)

print 'roots are',r1,r2

elif d>0:

print 'Roots are real and distinct'

r1=(-b+math.sqrt(d))/(2\*a)

r2=(-b-math.sqrt(d))/(2\*a)

print 'Root\_1 and Root\_2 :',r1,r2

else:

re=-b/(2\*a)

img=math.sqrt(-d)/(2\*a)

print 'roots are imaginary'

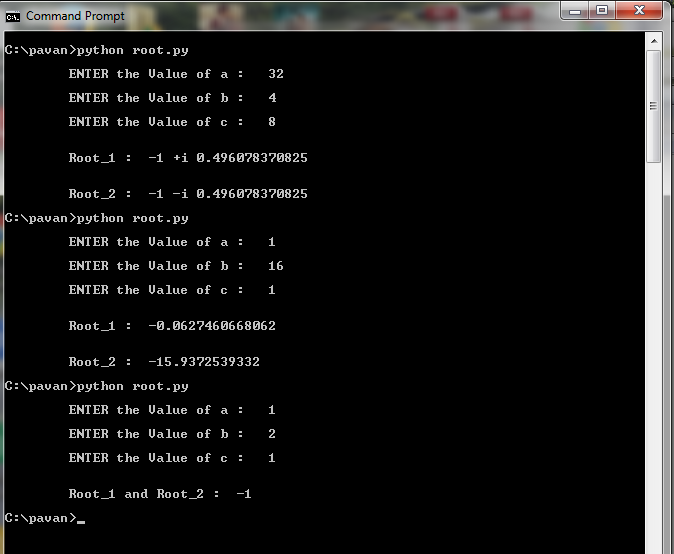
print "Root\_1:",re,"+i",img

print "Root \_2:",re,"-i",img

except:

print 'Enter numbers'

**OUTPUT :**



**Program :6. a) /\*Develop a Python program to check if the number provided by the user is an Armstrong number or not.\*/**

n=int(raw\_input("ENTER a NUMBER"))

sum=0

m=n

while m>0:

digit=m%10

m=m/10

sum=sum+(digit\*digit\*digit)

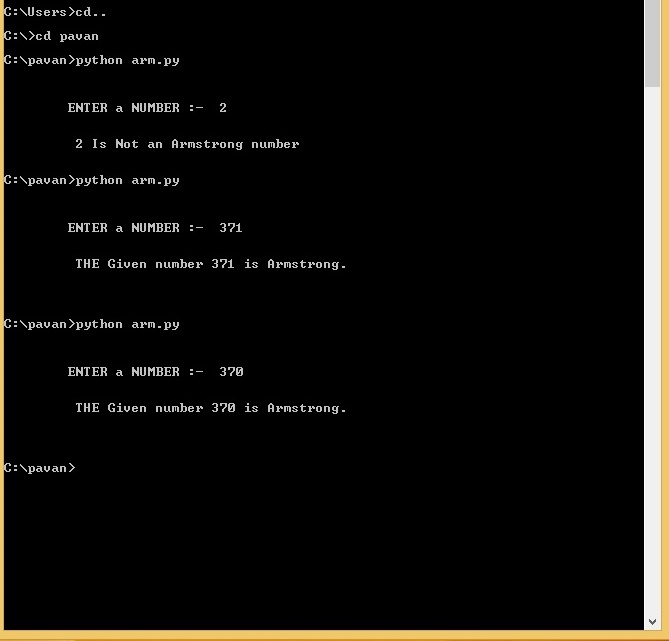
if(sum==n):

print “THE Given number",n,”is Armstrong"

else:

print n,”is Not an Armstrong number"

**OUTPUT :**

****

**Program :6. b)/\* Implement Python script to add two matrices.\*/**

x=[[1,2,3],[4,5,6],[7,8,9]]

y=[[1,2,3],[4,5,6],[7,8,9]]

result=[[0,0,0],[0,0,0],[0,0,0]]

for i in range(len(x)):

for j in range(len(y[0])):

result[i][j]=x[i][j]+y[i][j]

for r in result:

print r

**OUTPUT :**

****

**Program :7. a) /\*Develop a Python Program to multiply two matrices.\*/**

x=[[12,7,3],[4,5,6],[7,8,9]]

y=[[5,8,1,2],[6,7,3,0],[4,5,9,1]]

result=[[0,0,0],[0,0,0],[0,0,0]]

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

**OUTPUT :**

****

**Program :7. b)/\* Retrieve and display image using appropriate Python library.\*/**

from PIL import Image

a = Image.open("someImage.png")

a.show()

**OUTPUT :**

**Program :8. a) /\*Develop a Python Program to check for a valid date.\*/**

import datetime

def validate(date\_text):

try:

datetime.datetime.strptime(date\_text, '%d-%m-%Y')

print 'Success. Entered Date is Validated'

except ValueError:

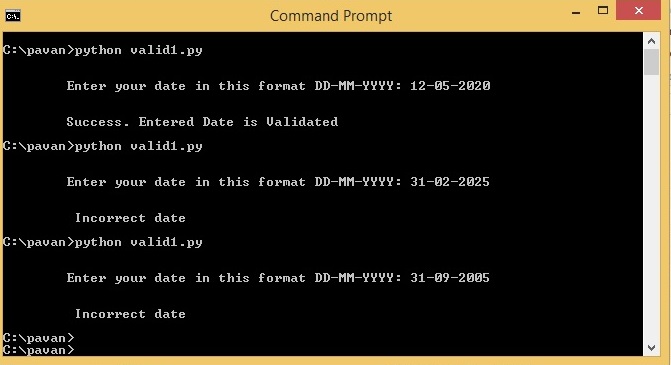
# raise ValueError("Incorrect date")

print 'Incorrect date'

print 'Enter your date in this format DD-MM-YYYY'

validate('31-02-2013')

validate('12-06-2015')

**OUTPUT :  
**

**Program :8.b)/\* Implement stack operations using Python Script.\*/**

class Stack :

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

self.items = []

def push(self, item):

self.items.append(item)

def pop(self):

return self.items.pop()

def is\_empty(self):

return (self.items == [])

s = Stack()

s.push(54)

s.push(45)

s.push(75)

s.pop()

while not s.is\_empty():

print s.pop()

**OUTPUT :**

****

**Program :9.a)/\* Develop Python Program to implement the union and intersection on a given list.\*/**

def intersect(a,b):

return list(set(a)&set(b))

def union(a,b):

return list(set(a)|set(b))

#if \_name\_=="\_main\_":

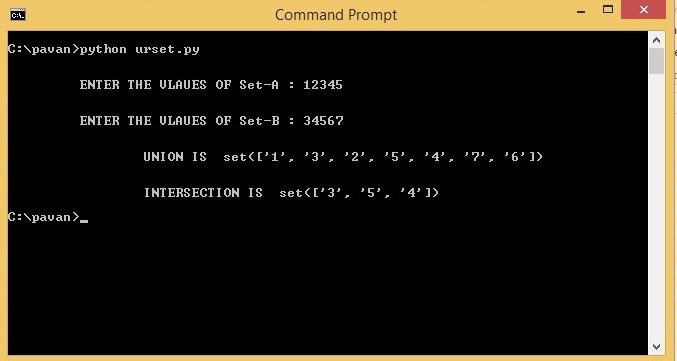
a=int(raw\_input(‘ENTER THE VALUES OF Set-A:’)

b= int(raw\_input(‘ENTER THE VALUES OF Set-B:’)

print intersect(a,b)

print union(a,b)

**OUTPUT :**

****

**Program :9. b)/\* Design Python program to find area of a triangle.\*/**

def area(a,b,c):

s=(a+b+c)/2

area=(s\*(s-a)\*(s-b)\*(s-c))\*\*0.5

return area

try:

a=float(raw\_input('ENTER FOR A:'))

b= float(raw\_input('ENTER FOR B:'))

c= float(raw\_input('ENTER FOR C:'))

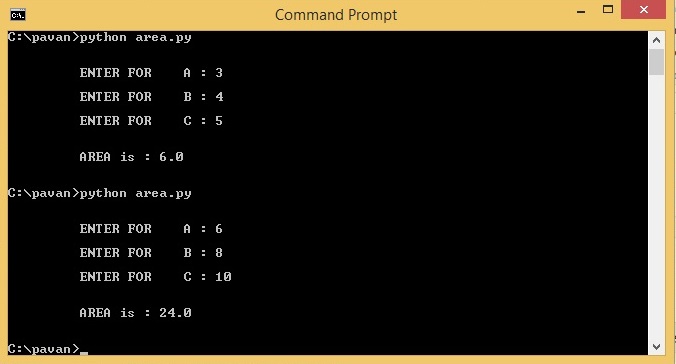
ar=area(a,b,c)

print 'AREA is',ar

except:

print 'Invalid input'

**OUTPUT :**

****

**Program :10. a) /\*Develop a Python Program to sum the elements of an array.\*/**

count = 0

for itervar in [3, 41, 12, 9, 74, 15]:

count = count + 1

print ' Count: ' , count

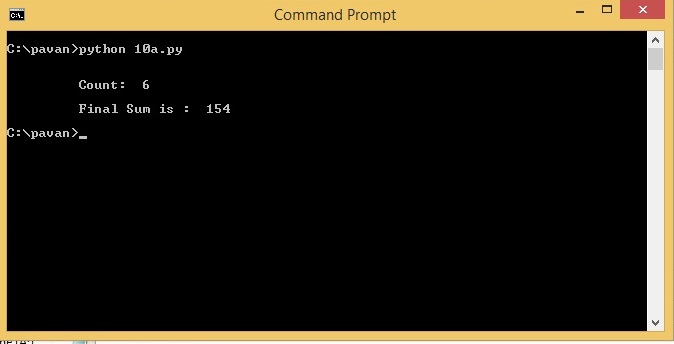
finalSum = 0

for itervar in [3, 41, 12, 9, 74, 15]:

finalSum = itervar + finalSum

print ' Final Sum is : ' , finalSum

**OUTPUT :**

****

**Program :10. b)/\* Write a Python program to display all files ending with .txt extension in the current directory.\*/**

import os

directorypath=raw\_input('ENTER THE DIRECTORY PATH:\n')

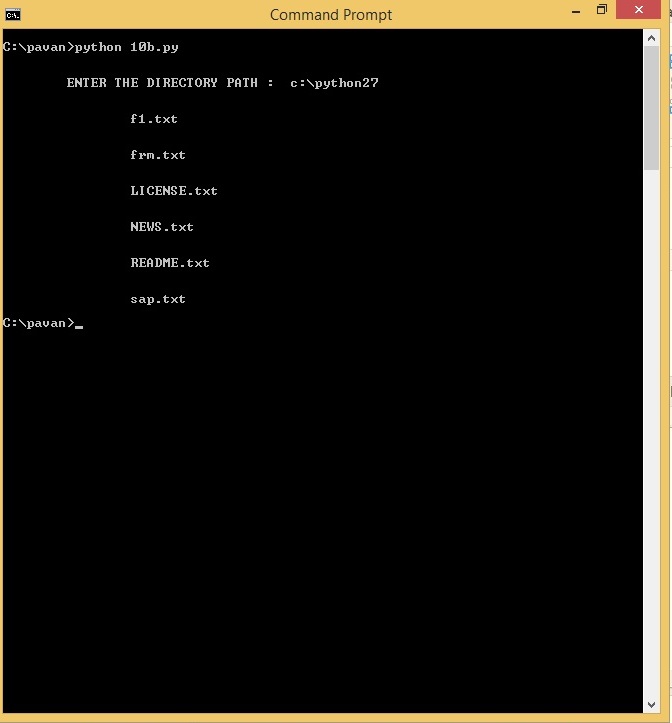
os.chdir(directorypath)

for file in os.listdir(directorypath):

if file.endswith('.txt'):

print file

**OUTPUT :**

****