Complex number operations:

class Cmpl:

#Constructor

def \_\_init\_\_(self,x,y):

self.x=x

self.y=y

#Addition

def add(self,c2):

tempx=c2.x+self.x

tempy=c2.y+self.y

return (tempx,tempy)

#Subtraction

def sub(self,c2):

tempx=self.x-c2.x

tempy=self.y-c2.y

return (tempx,tempy) #returning as tuple

#multiplication

def mul(self,c2):

tempx=c2.x\*self.x+(-1\*self.y\*c2.y)

tempy=self.x\*c2.y+self.y\*c2.x

return (tempx,tempy)

#division

def div(self,c2):

temp=-1\*c2.y

num\_real=self.x\*c2.x+self.y\*temp\*-1 #getting real part in numerator after multiplying by conjugate

num\_imag=self.x\*temp+self.y\*c2.x #getting imag part in numerator

denom=(c2.x\*\*2+c2.y\*\*2) #denominator is magnitude of 2nd object

num\_real=num\_real/denom #dividing by the magnitude of denominator

num\_imag=num\_imag/denom

return (num\_real,num\_imag)

c1=Cmpl(1,2) #creating objects

c2=Cmpl(3,4)

print('Division ',c1.div(c2))

print('Addition ',c1.add(c2))

print('Product ',c1.mul(c2))

print('Subtraction ',c1.sub(c2))

2) MAP FILTER LAMBDA

def sq(n):

return n\*n

l=[1,2,3,4,5]

print(list(map(lambda x : x%2==0,l)))

print(list(filter(lambda x : x%2==0,l)))

def bubsort(n,l):

for i in range(0,n-1):

for j in range(0,n-1-i):

if l[j]>l[j+1]:

temp=l[j]

l[j]=l[j+1]

l[j+1]=temp

print(l)

n=int(input("enter number of elements\n"))

print("enter elements\n")

l=[]

for i in range(n):

item=int(input())

l.append(item)

bubsort(n,l)

///////////

2) split using in built function and regex

import re

def spl(st):

li=re.findall(r'[^,]+', st)

print(li)

st=input('enter a string\n')

spl(st)