Write a python program that defines a type **Vector** to represent a vector in three dimensions. Later, create following functions and also code in main function to demonstrate the usage of **Vector3** type.

*createVector(x, y, z) # returns a vector with x,y,z as components*

*reverseVector(v) # returns a vector with components: -vx -vy -vz*

*addVectors(v1, v2) # returns a vector v1+v2*

*dotProd(v1, v2) # returns a number v1x·v2x + v1y·v2y + v1z·v2z*

*crossVector(v1, v2) # returns a vector as cross product: v1 X v2*

*magnitute(v) # returns a number as magnitude of v*

class vector3:

pass

def createVector(x, y, z): # returns a vector with x,y,z as components

v = vector3()

v.x = x

v.y = y

v.z = z

return v

def reverseVector(v): # returns a vector with components: -vx -vy -vz

v = vector3()

v.x = -v.x

v.y = -v.y

v.z = -v.z

return v

def addVectors(v1, v2): # returns a vector v1+v2

v = vector3()

v.x = v1.x + v2.x

v.y = v1.y + v2.y

v.z = v1.z + v2.z

return v

def dotProd(v1, v2): # returns a number v1x·v2x + v1y·v2y + v1z·v2z

d = v1.x \* v2.x + v1.y \* v2.y + v1.z \* v2.z

return d

def crossProd(v1, v2): # returns a vector as cross product: v1 X v2

v = vector3()

v.x = v2.y \* v3.z – v2.z \* v3.y

v.y = v3.z \* v1.x – v3.x \* v1.z

v.z = v1.x \* v2.y – v1.y \* v2.x

return v

def magnitute(v): # returns a number as magnitude of v

m = (v.x \*\* 2 + v.y \*\* 2 + v.z \*\* 2) \*\* (0.5)

return m

def vector3AsStr(v): # returns string of a vector v

s = "(" + str(v.x) + "," + str(v.y) + "," + str(v.z) + ")"

return s

def main1(): # returns a number as magnitude of v

vp = createVector(1,1,1)

vq = createVector(2,2,2)

vr = createVector(-2,-1,-3)

vs = addVectors(vp, vq)

vs2 = vector3AsStr(vs)

print(vs2)

**main1()**