import math

class vectors:

\_\_m\_vector\_1 = [0, 0, 0]

\_\_m\_vector\_2 = [0, 0, 0]

\_\_m\_result = [0, 0, 0]

def setVectors(self, vector\_1, vector\_2):

self.\_\_m\_vector\_1 = vector\_1;

self.\_\_m\_vector\_2 = vector\_2;

def getResult(self):

return self.\_\_m\_result

def sum(self):

self.\_\_m\_result[0] = self.\_\_m\_vector\_1[0] + self.\_\_m\_vector\_2[0]

self.\_\_m\_result[1] = self.\_\_m\_vector\_1[1] + self.\_\_m\_vector\_2[1]

self.\_\_m\_result[2] = self.\_\_m\_vector\_1[2] + self.\_\_m\_vector\_2[2]

def difference(self):

self.\_\_m\_result[0] = self.\_\_m\_vector\_1[0] - self.\_\_m\_vector\_2[0]

self.\_\_m\_result[1] = self.\_\_m\_vector\_1[1] - self.\_\_m\_vector\_2[1]

self.\_\_m\_result[2] = self.\_\_m\_vector\_1[2] - self.\_\_m\_vector\_2[2]

def dotProduct(self):

self.\_\_m\_result = (self.\_\_m\_vector\_1[0] \* self.\_\_m\_vector\_2[0]) \

+ (self.\_\_m\_vector\_1[1] \* self.\_\_m\_vector\_2[1]) \

+ (self.\_\_m\_vector\_1[2] \* self.\_\_m\_vector\_2[2])

def crossProduct(self):

self.\_\_m\_result[0] = (self.\_\_m\_vector\_1[1] \* self.\_\_m\_vector\_2[2]) \

- (self.\_\_m\_vector\_1[2] \* self.\_\_m\_vector\_2[1])

self.\_\_m\_result[1] = (self.\_\_m\_vector\_1[2] \* self.\_\_m\_vector\_2[0]) \

- (self.\_\_m\_vector\_1[0] \* self.\_\_m\_vector\_2[2])

self.\_\_m\_result[2] = (self.\_\_m\_vector\_1[0] \* self.\_\_m\_vector\_2[1]) \

- (self.\_\_m\_vector\_1[1] \* self.\_\_m\_vector\_2[0])

def lenght\_1(self):

self.\_\_m\_result = math.sqrt( (self.\_\_m\_vector\_1[0] \* self.\_\_m\_vector\_1[0]) \

+ (self.\_\_m\_vector\_1[1] \* self.\_\_m\_vector\_1[1]) \

+ (self.\_\_m\_vector\_1[2] \* self.\_\_m\_vector\_1[2]) )

def lenght\_2(self):

self.\_\_m\_result = math.sqrt( (self.\_\_m\_vector\_2[0] \* self.\_\_m\_vector\_2[0]) \

+ (self.\_\_m\_vector\_2[1] \* self.\_\_m\_vector\_2[1]) \

+ (self.\_\_m\_vector\_2[2] \* self.\_\_m\_vector\_2[2]) )

def lenght\_3(self):

self.\_\_m\_result = math.sqrt( (self.\_\_m\_vector\_1[0] - self.\_\_m\_vector\_2[0]) \

\* (self.\_\_m\_vector\_1[0] - self.\_\_m\_vector\_2[0]) \

+ (self.\_\_m\_vector\_1[1] - self.\_\_m\_vector\_2[1]) \

\* (self.\_\_m\_vector\_1[1] - self.\_\_m\_vector\_2[1]) \

+ (self.\_\_m\_vector\_1[2] - self.\_\_m\_vector\_2[2]) \

\* (self.\_\_m\_vector\_1[2] - self.\_\_m\_vector\_2[2]) )

vector = vectors()

def getValues():

vector\_1 = [0, 0, 0];

vector\_1[0] = input("Please enter first cordinate of X ")

vector\_1[1] = input("Please enter first cordinate of Y ")

vector\_1[2] = input("Please enter first cordinate of Z ")

vector\_2 = [0, 0, 0];

vector\_2[0] = input("Please enter second cordinate of X ")

vector\_2[1] = input("Please enter second cordinate of Y ")

vector\_2[2] = input("Please enter second cordinate of Z ")

vector.setVectors(vector\_1, vector\_2)

getValues()

operation = raw\_input("Please input the operation")

if operation == '+':

vector.sum()

elif operation == '-':

vector.difference()

elif operation == '\*':

vector.dotProduct()

elif operation == 'x':

vector.crossProduct()

elif operation == 'A':

vector.lenght\_1()

elif operation == 'B':

vector.lenght\_2()

elif operation == 'AB':

vector.lenght\_3()

print vector.getResult()