In [126…
```
class vector2:
    def __init__(self,x=0,y=0):
        self.x=x
        self.y=y
    def __str__(self):
        return "[{},{}]".format(str(self.x),str(self.y))
a=vector2(2,45)
print(a)
```
```
[2,4]
```
In [127…
```
j=vector2(5,2)
print(j)
```
```
[5,2]
```
In [128…
```
def add(self,j):
    c=vector2()
    c.x=self.x +j.x
    c.y=self.y+j.y
    return c
vector2.add=add
```
In [144…
```
c=a.add(j)
print(c)
```
```
[7,6]
```
In [145…
```
def mul(self,k):
    return vector2(k * self.x ,k * self.y)
vector2.mul =mul
```
In [146…
```
d=a.mul(6)
print(d)
```
```
[4,8]
```
In [147…
```
def sub(self,k):
    return self.add(k.mul(-1))
vector2.sub=sub
```
In [148…
```
d_min_b=d.sub(j)
print(d_min_b)
```
```
[-1,6]
```
In [134…
```
def dot(self, g: vector2):
    return self.x * g.x + self.y * g.y


vector2.dot = dot
```
In [135…
```
dota = a.dot(j)
print(dota)
```
```
18
```
In [149…
```
class vector3:
    def __init__(self,x=0,y=0,z=0):
        self.x=x
        self.y=y
        self.z=z
    def __str__(self):
        return "[{},{},{}]".format(str(self.x),str(self.y),str(self.z))
n=vector3(7,3,122)
print(n)
```
```
[7,3,122]
```
In [157…
```
def add_232(self,j):
    c=vector3()
    c.x=self.x +j.x
    c.y=self.y+j.y
    c.z=self.z+j.z
    return c
vector3.add_232=add_232
```
In [158…
```
c=n.add_232(n)
print(c)
```
```
[14,6,244]
```
In [ ]:
```
Part 7
User will insert a N elements of vectorN ,let it be dymanically in this case
Then we will use a loop to insert all elements into a list
```
In [ ]:
```

```