```
#Lab 12 Solution
class vector(object):
    '''A class that represents a simple vector in 0 or more dimensions.'''
    def __init__(self, comp=[]):
        self.components = comp
        self.size = len(comp)

    def __repr__(self): #Q2 modified
        vector_info = "vector("
        if len(self.components) > 0:
            vector_info += str(self.components) #create a string from list
        return (vector_info + ')'+' of size '+str(len(self.components)))


    def __eq__(self,other): #Q1
        """returns True if 2 vectors are equal"""
        return(self.size==other.size and self.components==other.components)

    def magnitude(self): #Q3
        total=0
        for i in self.components:
            total += i*i
        return(total **0.5)

    def anyZero(self): #Q4
        for i in self.components:
            if (i==0):
                return True
        return False

    def __add__(self,other): #Q5
        """ adds 2 vectors and returns result in a new vector"""
        sum_vector = [0] * max(other.size, self.size)
        for i in range(other.size): #copy other into sum_vector
            sum_vector[i] += other.components[i]
        for i in range(self.size):  #add self to sum_vector
            sum_vector[i] += self.components[i]
        return vector(sum_vector)

#5 Writing code to use the class vector
#1
v0 = vector() #vector([]) also ok
print(v0)
#2 Write the code to create 2 vectors v1 with components 3,5 and v2 with components 1,2,0
v1=vector([3,5])
v2=vector([1,2,0])
v3=vector([2,4,5,6])
#3
print(v0)
print(v1)
print(v2)
print(v3)
#4 write the code to print the magnitude for v3.
print(v3.magnitude())
#5 Using the method in 1 above, write the code to check if the 2 vectors v1 and v2 are equal.
print(v1==v2)
#6 Write the code to check if v2 and v3 have the same magnitude.
print(v2.magnitude() == v3.magnitude())
#7 Write the code to check if v2 has any zero components.
print(v2.anyZero())
#8 Write code to add the 2 vectors v2 and v3.
print(v2+v3)
```