```java
package Version1;

import java.util.ArrayList;

/**
* This is the Vector class, being the first version, the methods and constructors
* are not implemented. The purpose of this version is to test it and to show the tests
* created will 100% fail. Allowing us to slowly create the vector class and ensure the
* class is sound.
*/

public class Vector {

    private int N = 0;

    private ArrayList<Double> data;

    //empty
    public Vector() {
        throw new UnsupportedOperationException();
    }

    //a vector is created of size - size, with elements initalized to D
    public Vector(int size, double D) {
        throw new UnsupportedOperationException();
    }

    //a vector is created to be initialized to the array D
    public Vector(double [] D) {
        throw new UnsupportedOperationException();
    }

    //the vector is initalized to Int I
    public Vector(int [] I) {
        throw new UnsupportedOperationException();
    }

    public void append(double[] doubleArray) {
        throw new UnsupportedOperationException();
    }

    public void append(int[] intArray) {
        throw new UnsupportedOperationException();
    }

    public Vector append(Vector V) {
        throw new UnsupportedOperationException();
    }

    public Vector append(double aDouble) {
        throw new UnsupportedOperationException();
    }

    //this will be equivalent to the vector V
    public Boolean equal(Vector V) {
        throw new UnsupportedOperationException();
    }

    //returns the # of elements
    public int getLength() {
        throw new UnsupportedOperationException();
    }

    //returns the value  this[i]
    public double getValue(int i) {
        throw new UnsupportedOperationException();
    }

         //add this to V, returning a Vector the same size as this
    public Vector add(Vector V) {
        throw new UnsupportedOperationException();
    }

    //add aDouble to every element of this
    public Vector add(double aDouble) {
        throw new UnsupportedOperationException();
    }

     //sub this - V
    public Vector sub(Vector V) {
        throw new UnsupportedOperationException();
    }

     //will return a sub vector between the
     //indices l and r inclusive
    public Vector subV(int l, int r) {
        throw new UnsupportedOperationException();
    }

     //Multiple every element of this by corresponding element in V
    public Vector Mult(Vector V) {
        throw new UnsupportedOperationException();
    }

     //Multiply every element of this by aDouble
    public Vector Mult(double aDouble) {
        throw new UnsupportedOperationException();
    }

     //returns this as a normalized vector
    public Vector Normalize() {
        throw new UnsupportedOperationException();
    }

    //Returns the euclidean distance between this and V
    public double EuclidianDistance(Vector V){
        throw new UnsupportedOperationException();
    }

} // end Vector
```