Task 1:

Define a `Vector` class. A `Vector` should have a `Point2D` as each of its endpoints. For now, just create a constructor and a toString method, and then create a few object instances to make sure that things work properly. Make sure that when you print the vector, its endpoints look reasonable too.

Task 2:

Overload the + and − operators in your `Vector` class so that you can add two `Vector`s, or subtract two `Vector`s. Like in our example, this should return a *new* `Vector`, rather than modifying either operand. Recall, the method that implements + is called `__add__`; the method that implements - is called `__sub__`. If you want a reminder on how to do so, take a look here.

Task 3:

Add a `normalise` method to your `Vector` class. This should return a new `Vector` that has the same direction as the original vector, a magnitude of 1, and starts at the origin (0, 0). This page shows instructions on how to normalise a vector; you will also need a way for your code to figure out the length of the X and Y components of the `Vector` from its endpoints. Think carefully about where to put this code – some of it might be better located somewhere other than in the `normalise` method.