# Homework #6

In this exercise, you will implement class Country, which represents a country with a name, a population, and an area.

## a. Here is a sample interaction from the Python shell:

```
>>> canada = Country('Canada', 34482779, 9984670)
>>> canada.name
'Canada'
>>> canada.population
34482779
>>> canada.area
9984670
```

The code above cannot be executed yet because class Country does not exist. Define Country with a constructor (method __init__) that has four parameters: a country, its name, its population, and its area.

## b. Consider this code:

```
>>> canada = Country('Canada', 34482779, 9984670)
>>> usa = Country('United States of America', 313914040, 9826675)
>>> canada.is_larger(usa)
True
```

In class Country, define a method named is_larger that takes two Country objects and returns True if and only if the first has a larger area than the second.

## c. Consider this code:

```
>>> canada.population_density()
3.4535722262227995
```

In class Country, define a method named population_density that returns the population density of the country (people per square km).

## d. Consider this code:

```
>>> usa = Country('United States of America', 313914040, 9826675)
>>> print(usa)
United States of America has a population of 313914040 and is 9826675 square km.
```

In class Country, define a method named __str__ that returns a string
representation of the country in the format shown above.

## e. After you have written __str__, this session shows that a __repr__ method

## would be useful:

```
>>> canada = Country('Canada', 34482779, 9984670)
>>> canada
<exercise_country.Country object at 0x7f2aba30b550>
>>> print(canada)
Canada has population 34482779 and is 9984670 square km.
>>> [canada]
[<exercise_country.Country object at 0x7f2aba30b550>]
>>> print([canada])
[<exercise_country.Country object at 0x7f2aba30b550>]
```

Define the __repr__ method in Country to produce a string that behaves like this:

```
>>> canada = Country('Canada', 34482779, 9984670)
>>> canada
Country('Canada', 34482779, 9984670)
>>> [canada]
[Country('Canada', 34482779, 9984670)]
```