Problem 1: RectangularRoom Class

You will need to design two classes to keep track of which parts of the room have been cleaned as well as the position and direction of each robot.

In `ps2.py`, we've provided skeletons for the following two classes, which you will fill in in Problem 1:

RectangularRoom

Represents the space to be cleaned and keeps track of which tiles have been cleaned.

Robot

Stores the position and direction of a robot.

We've also provided a complete implementation of the following class:

Position

Stores the *x*- and *y*-coordinates of a robot in a room.

Read `ps2.py` carefully before starting, so that you understand the provided code and its capabilities.

Problem 1

In this problem you will implement two classes, `RectangularRoom` on this page and `Robot` on the next.

For the `RectangularRoom` class, decide what fields you will use and decide how the following operations are to be performed:

- Initializing the object
- Marking an appropriate tile as cleaned when a robot moves to a given position (casting floats to ints - and/or the function `math.floor` - may be useful to you here)
- Determining if a given tile has been cleaned
- Determining how many tiles there are in the room
- Determining how many cleaned tiles there are in the room
- Getting a random position in the room
- Determining if a given position is in the room

Complete the `RectangularRoom` class by implementing its methods in `ps2.py`.

Although this problem has many parts, it should not take long once you have chosen how you wish to represent your data. For reasonable representations, *a majority of the methods will require only a couple of lines of code*.)

Hint:

During debugging, you might want to use `random.seed(0)` so that your results are reproducible.

Enter your code for `RectangularRoom` below.

Test: 1 class creation

```
room = RectangularRoom(5, 5)
room.getNumTiles()
```

Output:

```
Successfully created a room of size 25
```

Test: 2 test getNumTiles

```
This test creates a number of randomly sized rooms and
checks the number of tiles by calling room.getNumTiles().
```

Output:

```
room = RectangularRoom(6, 9)
room = RectangularRoom(13, 7)
room = RectangularRoom(20, 2)
room = RectangularRoom(11, 7)
room = RectangularRoom(4, 5)
Successfully created a room of size 20
```

Test: 3 unclean tiles

```
room = RectangularRoom(4, 6)
This tests that all squares are properly marked as unclean
by calling the isTileCleaned() and getNumCleanedTiles() methods.
```

Output:

```
Successfully created a room of size 24
Number of cleaned tiles:
```

Test: 4 test cleaningTiles

```
This test creates a randomly sized room and checks the
number of clean tiles by calling room.getNumCleanTiles(). Then, the
tiles in the room are cleaned and the check is performed again.
```

Output:

```
room = RectangularRoom(1, 2)
Successfully created a room of size 2
Number of clean tiles: 0
After cleaning, number of clean tiles: 2
```

Test: 5 unclean tiles

```
        room = RectangularRoom(6, 8)
        This test cleans a subset of the tiles by calling the cleanTileAtPosition
method,
        then checks that those tiles are marked as cleaned by calling the
isTileCleaned method.
```

Output:

```
    Successfully created a room of size 48
    After cleaning, number of cleaned tiles:
```

Test: 6 cleaning tiles repeatedly

```
        This test cleans the same tiles repeatedly, which should result in
        the tiles continuing to be marked clean.
```

Output:

```
    room = RectangularRoom(11, 14)
    Done with iteration 0, number of cleaned tiles is 25
    Done with iteration 1, number of cleaned tiles is 25
    Done with iteration 2, number of cleaned tiles is 25
    Done with iteration 3, number of cleaned tiles is 25
    Done with iteration 4, number of cleaned tiles is 25
```

Test: 7 getRandomPosition

```
         Test getRandomPosition by calling it 100 times and ensuring the outputs are
sensible.
```

Output:

Test: 8 isPositionInRoom

```
        Test isPositionInRoom by testing random positions inside and outside of a
room.
```

Output:

Test: 9 isPositionInRoom

```
        Test getRandomPosition and isPositionInRoom by generating random positions
        then testing if they are within the room.
```

Output: