This homework will do two things: it will review a lot of the concepts we have worked on in this course so far; and it will move us towards a more applied cognitive science and information programming, by encapsulating all the work within the object-oriented paradigm. You will be creating one class - an agent of sorts.

The class that you be will be implementing will be of a bunny. The bunny class will have the following attributes: an age (in months), a weight (in pounds), a name (as a string), and an x and y coordinate that denotes the bunny’s location in your free-range farm (in miles). The bunny also has two boolean attributes: whether it’s hungry, and whether it’s alive (I know, sad).

The bunny will have the following methods:

1. The bunny should have an initialization method: __init__(self, name). This method gives birth to a new bunny. So the age of the bunny should always start at 0. The weight of a newborn bunny is always 1. Baby bunnies are always born hungry (as opposed to full). And baby bunnies are always born in the x=0, y=0 location - which is the location of your home. The user is, however, required to give the bunny a name when it’s born. So the initializing method should receive only one parameter - the name, all other attributes should be set to their default.

2. The bunny should a method that allows it to calculate the distance that it is away from home: distance_from_home(self). This method returns the Euclidean distance the current location of the bunny and home, which is at x=0, y=0. (You will use this method within the next two methods).

3. The bunny should also have a string method that allows us to print bunny to see it’s current state: __str__(self).
   a. If the bunny is alive, this method should return a string like this: “BunnyName is N months old, weights M pounds, it’s and is well-fed/starving.” For example, if I create a bunny from scratch, and then print it, it would look like this:

   ```
   >>> pecan = bunny("Pecan")
   >>> print pecan
   Pecan is 0 months old, weights 1.0 pounds, it is well-fed, and it is 0 miles away from home.
   ```

   b. If the bunny is dead, then this method should print: “BunnyName died peacefully at the age of N months, D miles away from home. Rest in peace.”
4. The bunny has one main method which is called `live(self)`. If the bunny is dead, then this method simply prints out a statement reminding the owner of the bunny that their beloved bunny is dead. However, if the bunny is alive, this live method takes the bunny through one month of it’s life. Several exciting things happen during this month:
   a. First, the bunny moves some random amount in the $x$ and $y$ dimension. The amount in each direction is chosen from a float value drawn at random uniformly between [-1, 1].
   b. Second, from all this movement and overall fun activities, the bunny’s weight decreases naturally by 0.45 pounds.
   c. Third, during it’s travels, the bunny has some probability of encountering different amounts of grass. Therefore, the weight of the bunny increases by a random float variable chosen at random between [0.0, 1.0).
   d. Fourth, the bunny becomes undernourished if its weight falls below 1 pound. Therefore, if the weight is less than or equal to 1.0, the boolean attribute hungry turns to True. If the weight is above 1.0, then hungry is False.
   e. Fifth, unfortunately, bunnies only live to be 8 years old (96 months), and with all the dangers of the world, the bunny always has some probability of dying. The probability of the bunny dying increases inversely proportional to the bunny’s age from 0.0 to 1.0 by the following function: `probability_of_death = self.age / 96.0`. To determine whether the bunny dies in this month of its life, you have to draw a random number from 0.0 to 1.0. If the number is less than the `probability_of_death`, then the bunny dies. In this case, the bunny’s alive attribute becomes False. And a print statement announces that: “Your bunny just died. Sorry.” (Otherwise, the bunny keeps living).
   f. Finally, the bunny’s age increases by one.

5. Create one final method inside your bunny class that is called `livefulllife(self)` that iterates through each month of the life of the bunny (calling the previous method `live`, iteratively) until the bunny dies (hint: `while self.alive:`), then stops and returns the age and distance.
Test your bunny class by creating one bunny, making it live until it dies, and reporting on the age in which it died, and how far it was from home. Here are examples when I run my own bunny. On the left, step by step, printing in each step. On the right, when I run my bunny in one go.

```python
>>> p = bunny("Peanut")
>>> print(p)
Peanut is 0 months old, weighs 1.000000 pounds; it is starving, and it is 0.000000 miles away from home.
>>> plot()
>>> print(p)
Peanut is 1 months old, weighs 1.240386 pounds; it is well-fed, and it is 0.469868 miles away from home.
>>> plot()
>>> print(p)
Peanut is 2 months old, weighs 1.119686 pounds; it is well-fed, and it is 0.879387 miles away from home.
>>> plot()
>>> print(p)
Peanut is 3 months old, weighs 0.804754 pounds; it is starving, and it is 1.536419 miles away from home.
>>> plot()
>>> print(p)
Peanut is 4 months old, weighs 0.536512 pounds; it is starving, and it is 1.448029 miles away from home.
>>> plot()
>>> print(p)
Peanut is 5 months old, weighs 0.243062 pounds; it is starving, and it is 1.852362 miles away from home.
>>> plot()
>>> print(p)
Peanut is 6 months old, weighs 0.225759 pounds; it is starving, and it is 1.628736 miles away from home.
>>> plot()
>>> print(p)
Peanut is 7 months old, weighs 0.254109 pounds; it is starving, and it is 1.276354 miles away from home.
>>> plot()
>>> print(p)
Peanut is 8 months old, weighs 0.1975 pounds; it is starving, and it is 1.224389 miles away from home.
>>> plot()
>>> print(p)
Peanut is 9 months old, weighs 0.351961 pounds; it is starving, and it is 1.515134 miles away from home.
>>> plot()
>>> print(p)
Peanut is 10 months old, weighs 0.499042 pounds; it is starving, and it is 1.145380 miles away from home.
>>> plot()
>>> print(p)
Peanut is 11 months old, weighs 0.178433 pounds; it is starving, and it is 0.38215 miles away from home.
>>> plot()
Your bunny just died. Sorry.
>>> page
[12
>>> p.distance_from_home()
1.58406167725312966
>>> print(p)
Peanut died peacefully at the age of 12 months, 1.584062 miles away from home.
>>> plot()
Remember: Peanut died.
```