The deadline for this exercise sheet is **Monday, 02.07.2018, 08:00.**

Introductory Words

Remember that you need proper documentation to pass the homework. The documentation doesn't need to be *perfect*, but everything that needs a docstring, should have a docstring.

This week, the homework will not have to do much with the lecture as we cannot really make a homework about neural networks that is suitable for this class. Therefore, consider this exercise a revision.

1 Floppy Ears, Fluffy Fur

A `BunnyFriend` is a creature that is born small and fluffy and has the dream of becoming a big, full-grown bunny!
We want to help its dream come `True` by feeding and playing with it, but be careful... if you don't play enough, it might grow to resent you...

Your task is to implement the `BunnyFriend`.

1.1 The Class

Each instance of `BunnyFriend` has a name and an age (in days). Its stomach has room for 3 meals, but it's almost full when it's born (with 2/3 meals).
Your `BunnyFriend` will live for 10 days until it's growing up. During those 10 days, each day, you have time for 3 interactions. You can either feed the bunny (filling the stomach by one meal), play with the bunny, or ignore it.
Your `BunnyFriend` will remember how many times you played with it in total as well as if it's been played with on the current day. This information will influence the `mode` of the `BunnyFriend` - it is always in one of these four modes:

- `happy` Cute, small, happy. Default mode.
- `sad` Cute, small, but sad. The `BunnyFriend` will be in this mode if it has not been played with on the previous day.
- `grown_happy` Fully grown and happy. The `BunnyFriend` will go into this mode after having lived for 10 days and if the user has played with it at least 15 times during this time.
- `grown_angry` Fully grown and angry and out for you. The `BunnyFriend` will go into this mode after having lived for 10 days and if the user has played with it less than 15 times during this time.

These modes mostly just influence the art that is going to be drawn when showing the bunny. We provided four text files with this task containing the bunny drawings as ASCII art. Read in all these files at instantiation of a `BunnyFriend` and create a dictionary mapping the modes to the ASCII art strings.

After each day, the bunny will

- get a little more hungry again (meaning stomach fullness is reduced by 1)
- get sad if it's not been played with during the day
- run away if it is starving (meaning if stomach fullness is at 0, the `BunnyFriend` cannot be interacted with any longer)
- age by 1 day and grow up if it's 10 days old (into either grown happy or grown angry mode, as explained above)

Furthermore, if someone calls `print` on a `BunnyFriend` object, the user will be shown the drawing corresponding to the current mode of the `BunnyFriend` and how full its stomach is. If the `BunnyFriend` is not interactable anymore, the user will only see a message stating that it has left.

1.2 The Test Class

Write a small test class that prompts the user to enter a name for the `BunnyFriend`, then instantiates it. Then, days will pass until the `BunnyFriend` is not interactable anymore:

```
while BunnyFriend is interactable:

    for 3 times:
        print BunnyFriend
        interact with BunnyFriend

    let the day pass
```

1.3 Bonus: A Familiar Game

When playing with the `BunnyFriend` you are actually playing a game of Hangman! Import the Hangman game from a couple of weeks prior and start it whenever you are playing with the `BunnyFriend`.
For this, you should use the `hangman.py` we provided with this sheet. It is a slightly different version from the solutions to the homework in week 6! In this version, the game will not start by itself, but can be started using the `game` function, to which you can hand over the path to the file with the Hangman words.

1.4 Bonus: A Familiar Game Reloaded

Now *of course* a bunny cannot actually speak and know any words to give to you to guess.
Therefore, you first have to teach it words!
Add another interaction in which the user can teach the `BunnyFriend` a word. The `BunnyFriend` will, from now on, only use the words in the Hangman game that it has been taught before. Implement this without changing the Hangman module.
If the user is trying to play with the `BunnyFriend`, but it doesn't yet know any words, the original playing functionality without the Hangman game will be executed.