ISTA 350 lists vs. arrays Worksheet   Name:

Write a function called `make_list` that takes a filename and returns a list of lists. Each line of the file consists of an equal number of floats. The catch is, the list of lists is transposed. In other words, each line of the file is a column in the list of lists. For example, if the file contains:
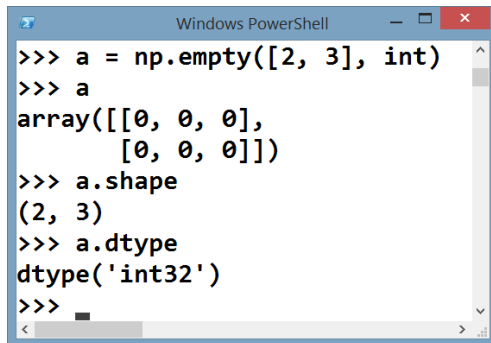
```
 0.78 14.3
-12.2 19
```

the function returns:

```
[[0.78, -12.2],
 [14.3,  19]]
```

If you are feeling brave, incorporate the transposition as you read in the data. If you are not feeling brave, make a list of lists and then use it to make a transposed list of lists. If you think you're done, call me over.

Another approach to the first problem is store the data in an np array and then use the `transpose` method, then turn the resulting array in a list of lists. To explore the difference between lists and arrays, write a function called `my_transpose`. It takes an np array as its sole argument and returns a new array that is the transpose of the argument without using the `transpose` method. You will discover that in some cases, arrays are much easier to work with than lists, even when I've barred you from using built-in methods. Everything you need to know:

```
>>> a = np.empty([2, 3], int)
>>> a
array([[0, 0, 0],
       [0, 0, 0]])
>>> a.shape
(2, 3)
>>> a.dtype
dtype('int32')
>>>
```