Homework 4

As always, use docstrings and good style, and don't forget to test your code! Write your solutions in a file named `yourname_homework4.py` and submit to m.spacek@lmu.de before class 8 (July 5).

Write one script to do all of 1-11. The script should be run from within IPython or Jupyter Notebook using `run yourname_homework4.py`. Try to ensure that all the plots pop up when the script is run, and that the command line remains interactive. If it doesn't, check the instructions on finding and modifying your `matplotlibrc` files near the end of the notes for class 6.

1. Define a function called `absdiff()` that returns the absolute value of the difference between two sequences (arrays, lists, or tuples). Assume that the two input sequences have the same length. It should return an array of the same length as both of the input sequences.

   Example:

   ```
   In [1]: a = [1, 2, 3]
           b = 10, 0, -10
           absdiff(a, b)
   Out[1]: array([ 9,  2, 13])
   ```

2. Load example time series data from `homework4.npz`. Extract all 3 arrays from the file. One of them is named `t` and respresents time points, but the names of the other two you need to figure out. Hint: inspect the object returned when loading the data. For the rest of this exercise, let's call the other two arrays `a` and `b`, but in your code and plots you should give them the name they have in the `.npz` file.

3. Plot `a` and `b` vs `t` in the same plot (i.e., on the same "axes")

4. Calculate the absolute value of the difference between `a` and `b` using your `absdiff()` function. Give the resulting array the name `absd`

5. Plot `absd` vs `t` on the same axes.

6. Label the x axis `Time (s)` and the y axis `Position (cm)` and give it a title `Time series`. Add a legend with appropriate labels for the three traces. You might have to modify your plot commands in 3. and 5. to add labels to the legend. Fortunately, you're writing a script that's easy to modify and re-run...

7. Save the figure to disk programmatically within the script. Name it `time_series.png`

8. Create a new figure and plot the *distributions* of `a`, `b` and `absd` on the same axes. Use the same set of bins for all 3 distributions. Choose bins that span the full range of data, and also have a reasonable amount of resolution. It might take you a few tries to figure out a decent set of bins. Fortunately, you're writing a script that's easy to modify and re-run...

9. Label the x axis `Position (cm)` and give it a title `Distributions`. Add a legend with appropriate labels for the three traces.

10. Save the figure to disk programmatically within the script. Name it `distributions.png`

11. Save `t` and `absd` to a single numpy binary file named `t_absd.npz`