Introduction:
The purpose of this exercise is to familiarize you with torch. This exercise will focus on lua and simple torch usage. You are required to hand in question 3, the rest of the questions are for self-learning. It is important to go over them, since we won’t cover basic lua/torch syntax during the course.

You have 2 options for using torch7:
I. If you have a ubuntu(12+) or Mac OS-X machine, you can install it from here: http://torch.ch/. The installation process is simple and straightforward.
II. You can access zebra server: http://compute.zebra-med.com:3001/login

For this exercise and future once, you should use torch7 documentation: http://torch.ch/docs/package-docs.html. The docs aren’t perfect and you will be required to look at the code: https://github.com/torch.

Question 0:
It’s time to learn lua, enter the following link: http://tylerneylon.com/a/learn-lua/
It contains several code snippets that explain basic features of the lua language, go over it.
Much more about lua: http://www.lua.org/manual/5.1/

Question 1:
Go over the torch tutorial here: http://torch.ch/docs/five-simple-examples.html
Notes: I. Everytime you see “luarocks install …” skip it - all the packages were installed.
II. On zebra, outputs of gnuplot can be seen in the “Reports” tab. There, open the folder containing your script.

Question 2:
Notes: I. On zebra, add the following lines to the start of your scripts:
   os.remove('*.png')
   itorch = require('zebra.report')
II. As before, you can view the output of your script (itorch.image) in the “Reports” tab.
Question 3 (for submission):
Write torch code to normalize MNIST dataset globally, i.e. has zero mean and a variance of 1.

If you are on zebra system, you can find the train and test files at: /data/TAU/mnist/mnist.t7/
On other systems, you will have to download the files, from shell run:
wget https://s3.amazonaws.com/torch7/data/mnist.t7.tgz
tar xvf mnist.t7.tgz

You can load the data like this:

```python
train = torch.load('mnist.t7/train_32x32.t7', 'ascii')
test = torch.load('mnist.t7/test_32x32.t7', 'ascii')
```

```ini
[train  
  [data] = ByteTensor - size: 60000x1x32x32
  [labels] = ByteTensor - size: 60000]
```

You will need to use mean/std method of torch and Tensor slicing. Look here for reference:
https://raw.githubusercontent.com/torch/tutorials/master/2_supervised/A_slicing.lua