CmpSci 201
Lab 1

This lab will introduce you to the rolecks ARM simulator, which is the recommended platform for testing your code.

Step 1 Download the rolecks system off of the course web page (http://www.cs.umass.edu/ trekp/cs201). Rolecks is a java program packaged as an executable jar file. As long as the system you're on has a recent version of Java installed you should be able to run it. On Mac OSX and Windows machines, you can just double click on the rolecks.jar icon. On Unix boxes (other than OSX) you may have to run java directly. Simply enter the directory where rolecks.jar is stored and type 'java -jar rolecks.jar'.

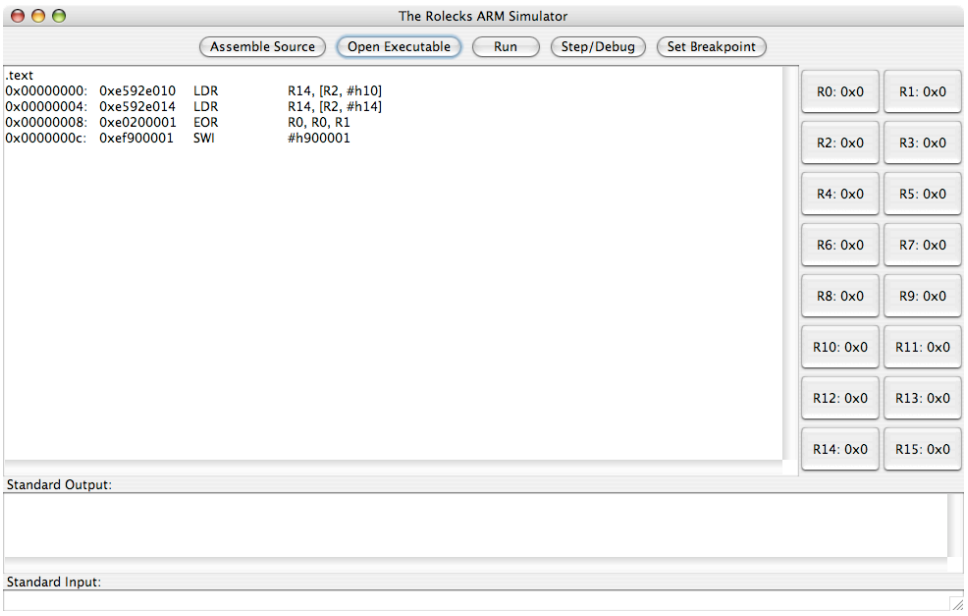
Step 2 Download the test code from the web page. (It's the code link under where you got this). Open it up in your favorite text editor. Go to the bottom of the code and change the line

```
mystudentid:    .int    #hFFFFFFFF
```

So that `#hFFFFFFFF` is replaced with your student number (no dashes or anything please). Remember that decimal constants need to have the # symbol in front of them!

Step 3 Fire up the rolecks system. Click on the button labeled 'Assemble Source'. This will bring up a file selection widget. Locate the modified test code, and select it. This will assemble the test code into the appropriate binary format for the ARM simulator to understand it. If there's an error, rolecks will display an alert message that will hopefully lead you straight to the problem.

Step 4 Click on the 'Open Executable' button and select the binary file. It should be in the same directory, and have the same name as the source file, but with a different extension (.bin). So if the source file was named "foo.s" the binary file will be named "foo.bin". After loading the file, a disassembled version of the code should appear in the main window and look something like this:



`.text` tells you you're in the text or code section of the binary. Each disassembled instruction generates 4 columns. The first column shows the address of that instruction. The second column shows the encoded value of the instruction. The third column has the instruction mnemonic in it. And the last column shows the arguments to the instruction. If there were any labels in the code, they would show up on their own line.

Step 5 Run the binary by clicking the 'Run' button. Record the value in R0, and email it to trekp@cs.umass.edu