EECS 470
Lab 5: Scripting

Goal: To build a script that will automatically run every testbench and validate its output is correct (essentially a P3 autograder). This should run for your P3.

You are free to build this script however you like, but we have outlined a rough path that we would suggest you follow as it will make life easier.

We have also provided several example scripts. Try skimming through them before starting to generate your own code, just to get a feel for different styles / ideas.

- Firstly you will need a "ground truth", that is a set of correct outputs. In theory you could have you script generate this every time, but that's just kind of wasteful.
  - Instead create a "solution" or "correct_output" or some other directory
  - Run the original, non-pipelined simulator for every testbench and save it's output
  - (Gee, this sounds a lot like a task you could automate...)
    - Try this command:

      -
        ```
        for file in `ls test_progs`; do echo "assemble $file"; file=$(echo $file | cut -d'.' -f1); echo "run $file"; echo "copy $file.writeback.out"; echo "copy $file.program.out"; echo; done
        ```
    - See where we're going with this...?

- Great! Now it's time to build the actual testing script

- The first thing this script should do is make sure it's running an up to date copy of your project
  - I find all of that output really verbose / annoying, but you can do some cool tricks
    ```
    make_out=$(make simv 2>&1)  # Redirect stdout->stderr and capture
                                # both in the variable make_out
    if [ $? -ne 0 ]; then       # if make fails...
       echo "Building simv failed!"
       echo "$make_out"         # These quotes are important to
                                # preserve whitespace in the output
    fi
    ```
  - At this level it's all style; maybe you like having make's output print every time

- Now we want to run every testbench
  - But you should already know how to do this...

- For each testbench, you should validate writeback.out and program.out
  - Do writeback.out first, it's easier as they must match exactly
  - For program.out, you only want to match lines starting with '@@@'

- Congratulations! You've just built the P3 autograder
  - (Do you see how this could be really helpful for your final project...?)

- Nifty [Optional] Extras
    - Read up on "commit hooks". Your script can run every time someone commits and email the results to your group
    - Check out the "PASSED / FAILED" printing section of `examples/run_tests.sh` to see how to colorize the output (because who doesn't like the pretty?)
    - You may want to figure out some way for your script to detect and kill runaway simulations (infinite loops):
        - This will work, but is a little annoying as it will always wait 5 seconds for each testcase

          ```
          ./simv &> test_output/$prog.log &
          sleep 5
          killall -9 simv &> /dev/null
          if grep -q '$finish' test_output/$prog.log; then
             echo "test case finished
          else
             echo "infinite loop :("
          fi
          ```

        - You can do better.