# Guide to using HW1 PL/0 P-Register Machine Emulator:

* To begin, first you must compile the source code labeled “final.c”.
  + To compile, use the GCC compiler and open up your OS’s terminal or command prompt.
  + Change directory to the one where “final.c” is located and run a GCC command to compile it.
  + Additionally, you may also renamed the compiled version of the file.
  + By default, however, it should compile into an “a.exe” or “a.out” file, dependent upon the OS you are using.
* Next, you must already have a text file holding the lines of instructions you would like to run. Each instruction should have an Opcode, R, L, and M variable, represented simply by 4 integers, all separated by a space. This is very important as the space will be counted as the delimiter for each variable.
  + For example: one line in the file may look like this:
    - 7 0 0 10
    - As you can see, there are 4 digits, each representing in order: OP, R, L, M; all separated by a space. This should conclude one line. After this line, you may include additional instructions you would like to run in this emulator, all following the same format.
* To finally run said instructions, open up your terminal or prompt, if you already have not done so, and change directory to where you compiled file is located.
  + Note: This program takes in the text input as a command line argument, the way this program was written allows for two options of doing so:
    - 1. If your text file is within the same directory as your compiled file, you may pass in just the title of your text file as an argument.
      * Example: a.exe input.txt
      * Example: a.out input.txt
    - 2. Else, you must specify the location of your file and pass it in, instead, as your argument.
      * Example: a.exe C:\Users\Abdool\Desktop\Project1\input.txt
      * Example: a.out C:\Users\Abdool\Desktop\Project1\input.txt
* If done properly, you should now see an 2 outputs:
  + 1. At the top, you should see a brief rundown of all your instructions that are going to be run, including the line number, operation name, register, lexicographical level, and modifier.
  + 2. Immediately after that, you should start seeing each line of code after its execution, first line being the initial values and every line after that being the actual execution. You can verify this by checking each variable and the data stack being outputted after each run to see the simulations being taken placed.
* If you would, instead, prefer to have your output logged to a separate text file, run the same command line but add in “>output.txt” or however else you would like to name your output file.
  + Example: a.exe input.txt>output.txt
  + Example: a.out input.txt>output.txt