1 Part A

In this assignment you will rewrite Chapter 11 Exercises 1 and 2 from the Maxeler compiler tutorial such that you will use a 1024x901 green border for exercise 1 that your kernel will generate on the fly (as opposed to reading in a blue image as given in the example.) The width and height dimensions must be passed to the manager from your CPU code.

You must include a C implementation of your kernel to verify that the output matches.

Hints:

- Recall that the total number of ticks that your kernel has to run is equal to the number of pixels in the output image. You will have to set the value of $N$ accordingly.
- Copy `lena256.ppm` from the Chapter 11 sample solution:
  `tutorial-chap11-solution1-mergesolution/EngineCode/src/mergesolution`
- To view a `.ppm` image in your VM, use the `eog` command like this:
  ```
  eog someImageFile.ppm
  ```

2 Part B

In Exercise 2, you will remove every other vertical row of pixels (shrink its vertical dimension rather than shrink it in two dimensions.)

You must include a C implementation of your kernel and verify that the output matches.

3 Compiling and Testing Your Programs

Build and test your program using the VM as discussed in class. Name your projects as discussed in class: `z88888-a06a` and `z88888-a06b`.

4 How to Hand in Your Program

Hand in your program using the SVN repo as discussed in class and in the appendix of the lecture notes. Don’t forget to create the repo and commit your files!!!