Question 4 (25 points): The current processor commercialized by *TinyProc Inc* is the TP500 that runs at a 500 MHz frequency (1MHz = 10^6 Hz). The most important application that runs in the TP500 is `ControLux` and it controls a luxury car dashboard. The average number of clock per instruction executed by TP500 when running `ControLux` is 1.2. A new model of this car is under design that will require a new version of this application, `ControLux 2.0`. `ControLux 2.0` will require the execution of twice as many instructions as `ControLux`. The hardware group at *TinyProc Inc.* has figured out a way to improve the clock frequency for the processor and is planning for a TP750 that will run at a frequency of 750 MHz without affecting the CPI if the same compiler is used. *TinyProc Inc.* would like the performance of the TP750 in the new car model to be the same as the TP500 in the previous car model. Now it falls to your compiler group to improve the code generation to either reduce the number of instructions executed by the new version of the application or to reduce its CPI.

1. (**8 points**) How does the performance of `ControLux 2.0` running on TP750 compares with the performance of `ControLux` running on TP500? (Which one is faster and my how much?)

2. (**8 points**) If the compiler group manages to reduce the CPI for `ControLux 2.0` on TP750 to 1.0, how would the performance of `ControLux 2.0` running on TP750 compare with the performance of `ControLux` running on TP500?

3. (**9 points**) Assuming that the compiler group manages to reduce the CPI for `ControLux 2.0` on TP750 to 1.0, by what percentage the number of instructions executed by `ControLux 2.0` on TP750 would have to be reduced in order for the performance of `ControLux 2.0` on TP750 to be the same as that of `ControLux` on TP500?