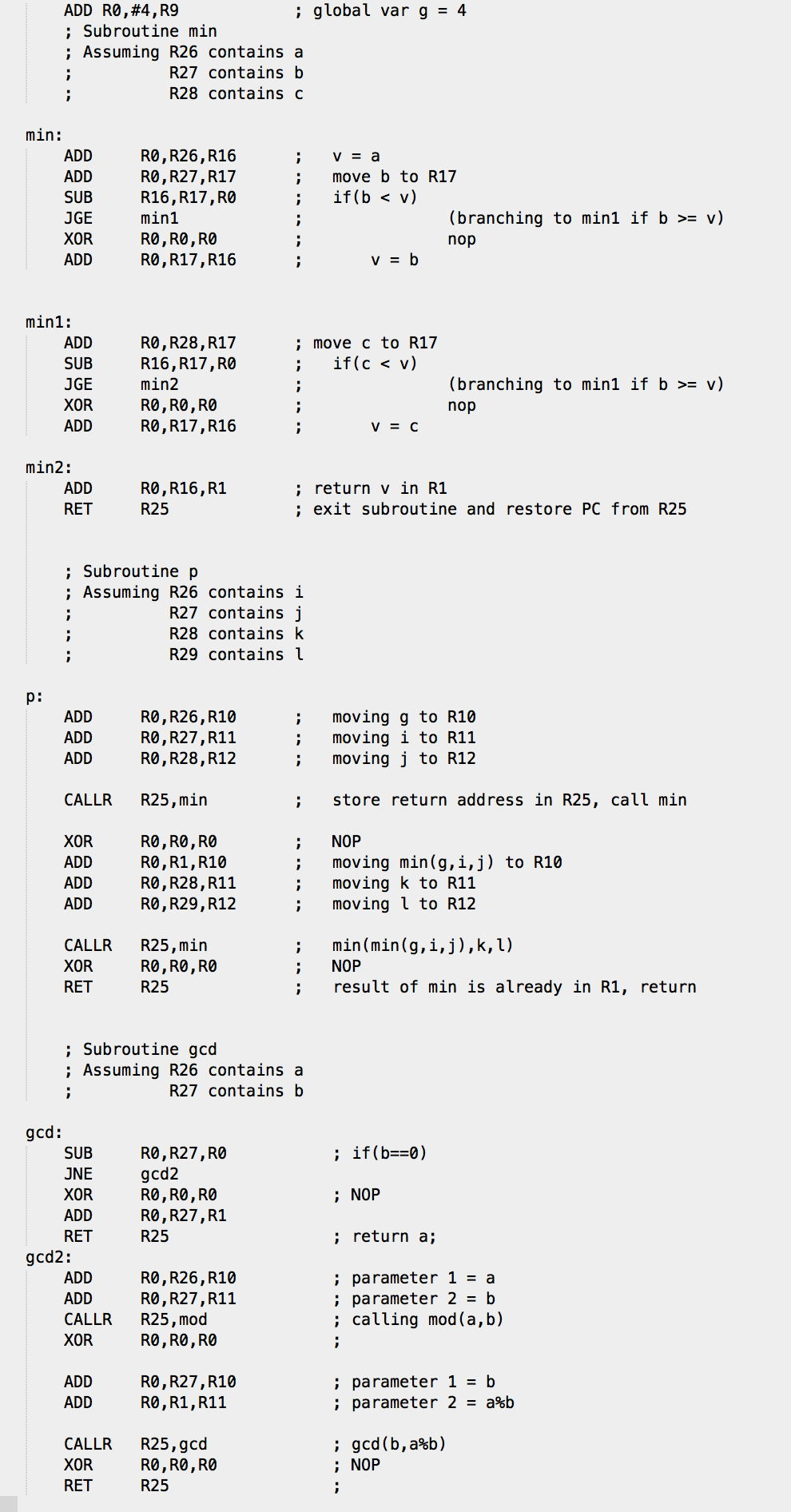
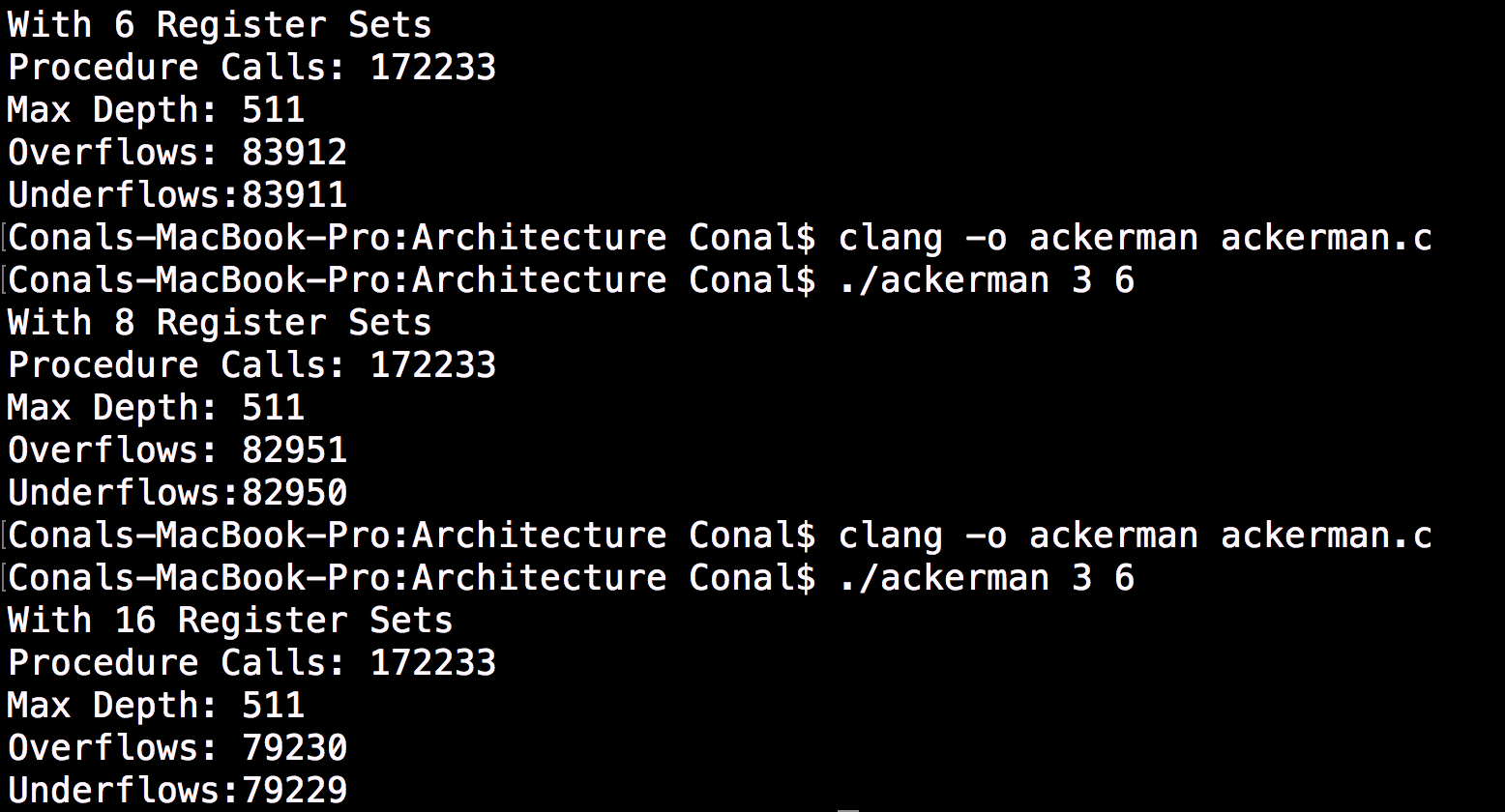
Computer Architecture II

Tutorial 3

**Q1. Translate the C/C++ code segment into RISC-1 assembly language**



**Q2. Ackerman Function**

Below are the results of my tests running this function with 6,8 and 16 register sets

As would be expected, the number of procedure calls and maximum depth does not change, however the amount of times that the register sets over/underflow is reduced as the number of register sets increases.

**Q3. Determine how long it takes to calculate ackermann(3, 6) on your computer. Make sure you time the release version of your code. Briefly describe your approach and comment on its accuracy.**

To calculate how long it takes to run ackermann(3,6) on my computer I used the “time.h” library provided by c. This tells you exactly how much time the program is using the cpu for and can therefore give an accurate representation of how long the program takes to run.

As with all good experiments I decided to repeat multiple times and take the average time. The average of 1000 calls of ackerman(3,6) was 0.001533 seconds.

