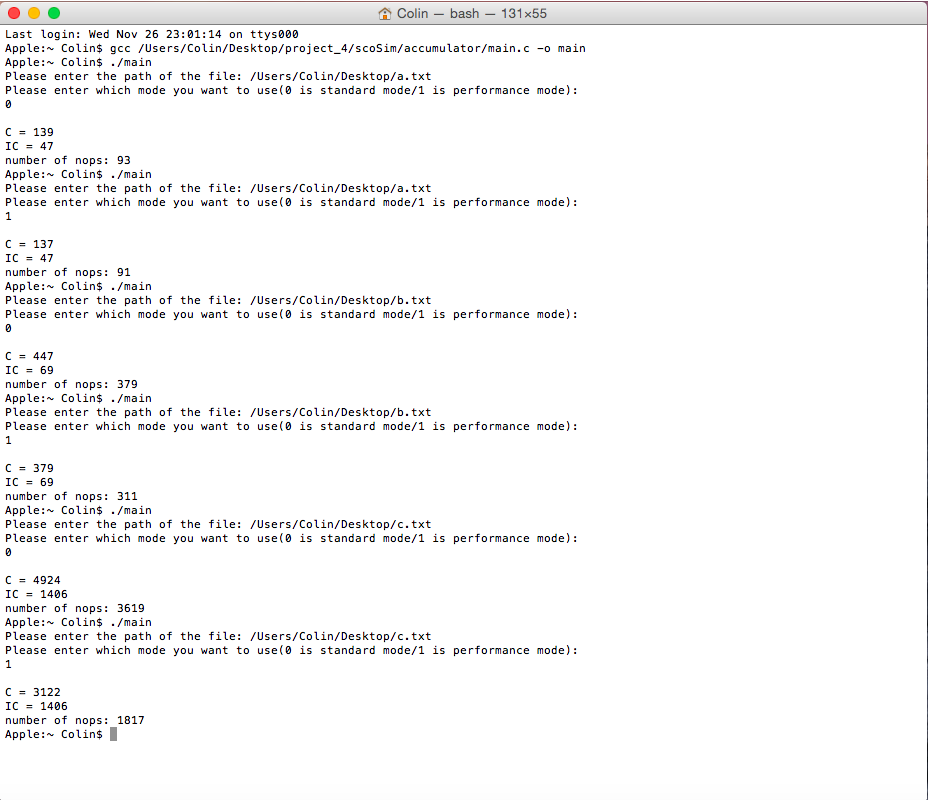
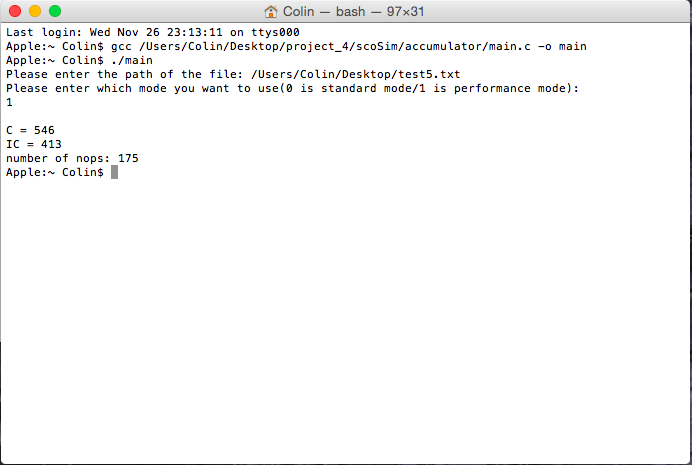
Name: Yitong Dai ID yzd0014

a. Please check “guidance ” to learn how to use my simulator.

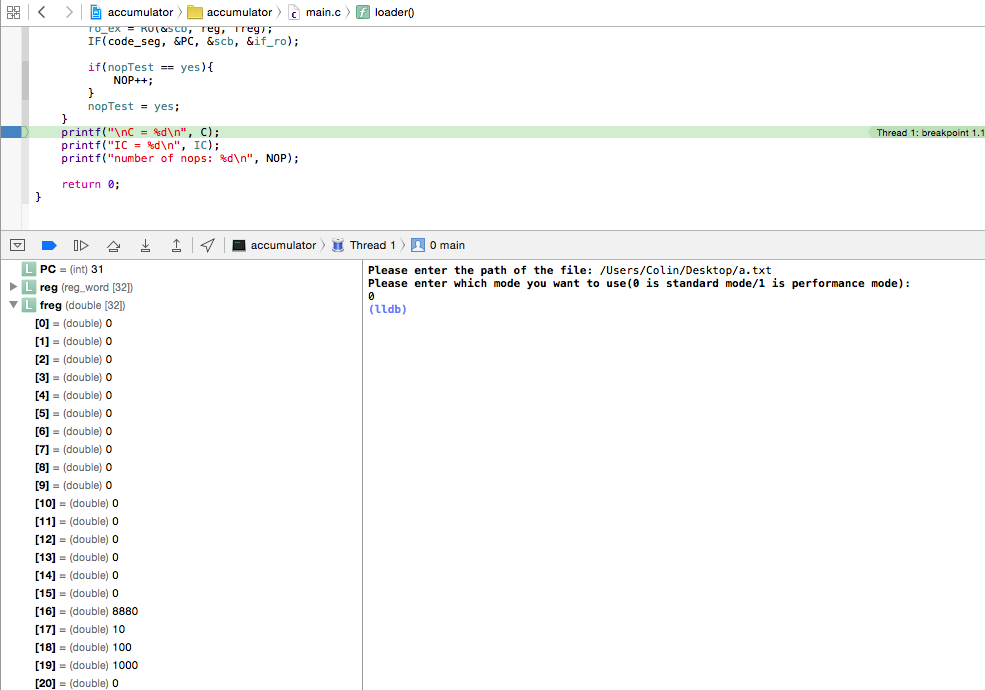
b. Total clock cycles(C), total instructions executed(IC) and number of nops are listed below.



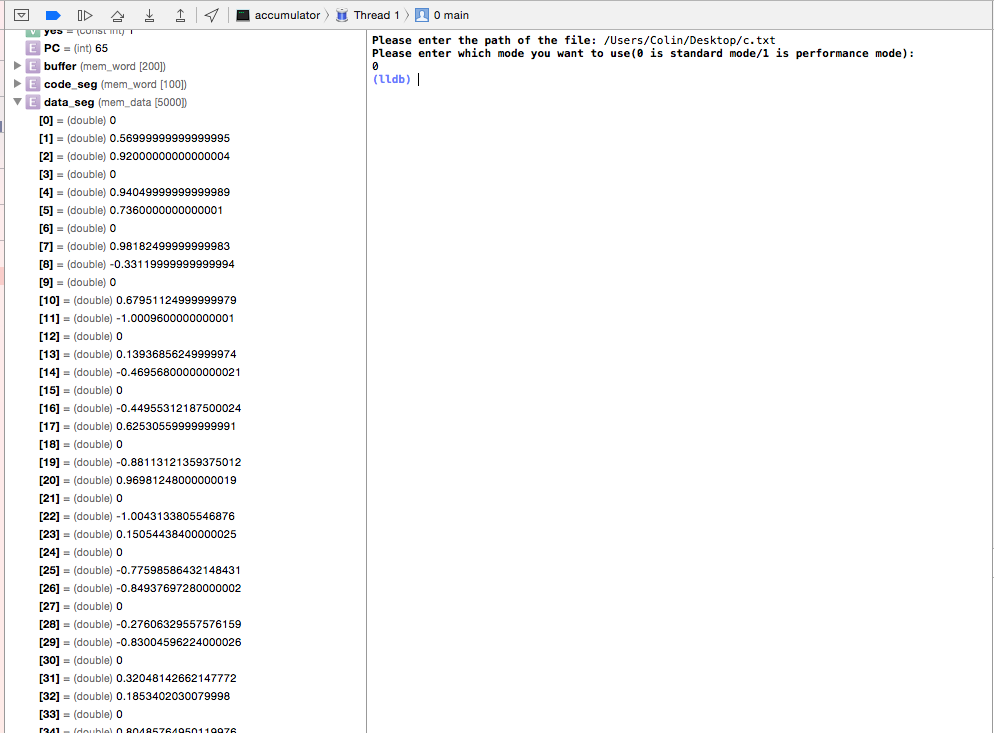
c. Meaning of number of nops I track: how many clock cycles are there that don’t lead completion of any instruction. For example, in the last benchmark(c.txt) number of nops is 1817, what it means is at the end of every single one of these 1817 cycles no instruction have been finished. What’s more if you add IC and number of nops, then result roughly equals C which will conform what I explain. The reason they don’t equal exactly is in some clock cycles multiple instructions can be finished due to parallel computing.

d. Because above three benchmarks have a bunch of hazard it can’t take full advantage of pipeline. I write another benchmark which is called test5.txt in my package. The result is shown below.

In this benchmark every instruction almost only takes one clock cycle to finish it.

e. To check the execution correctness of a.txt and c.txt you need do something special. Because this time the assembly code doesn’t have the capability to print the result, also for c.txt the result is too long to print. You need a programming circumstance in order to insert a breakpoint, then you can check any variable’s vale.

freg[16] keeps the final result of a.txt benchmark like what my snapshot shows above.



data\_seg[] array keeps the final result of c.txt benchmark. There are one hundred pair of numbers which represent the result of one hundred iterations.