**Now we need to verify that you actually can use MARS successfully.**

* When you run the previous program, what is printed?
* What is the value in register $t7 (in decimal) when the program ends?
* Set a breakpoint for the instruction at line 13 of the assembler source code.  Run the program again; it should stop at the breakpoint.  Now execute that one instruction.  Which registers have changed as a result of executing that one instruction?  You might need to continue past the breakpoint several times to see what's going on.  Note that P&H COD Appendix A.10 has descriptions of all the instructions, but you can't just look up the answer.  (You should look up the instructions in App. A.10, but the answer requires you to pull together several different pieces of information, not just one.)

The sum from 0 .. 100 is :338350:

-- program is finished running –

value in register $t7 is 10000

$t6 have changed, it represents the current value of i, every time it will increase 1

$t7 will change according to $s6, since it will be the multiply of $s6\*$s6;

And load to lo, sum to $t8

Save the total sum to $t9

I=i+1 and save to i, which is $t0

While i < 100, jump to start of the loop.