**Penn State University**

**School of Electrical Engineering and Computer Science**

**CMPEN331 – Computer Organization and Design**

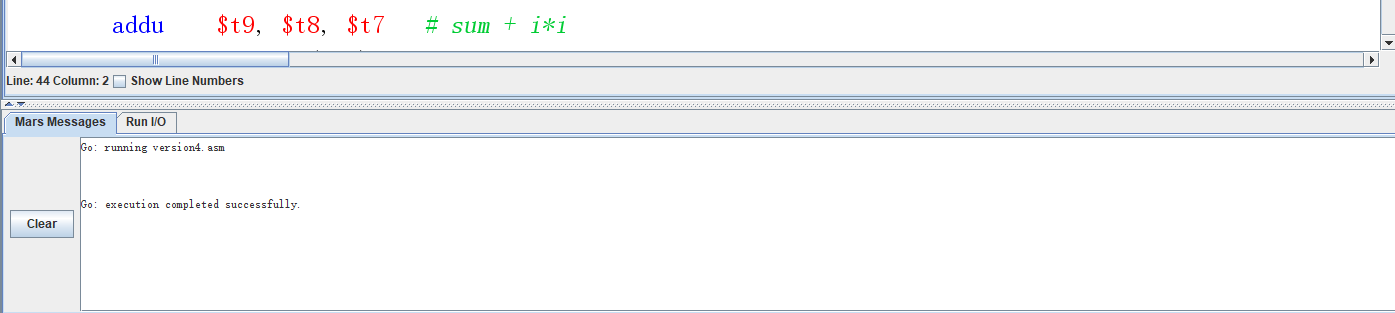
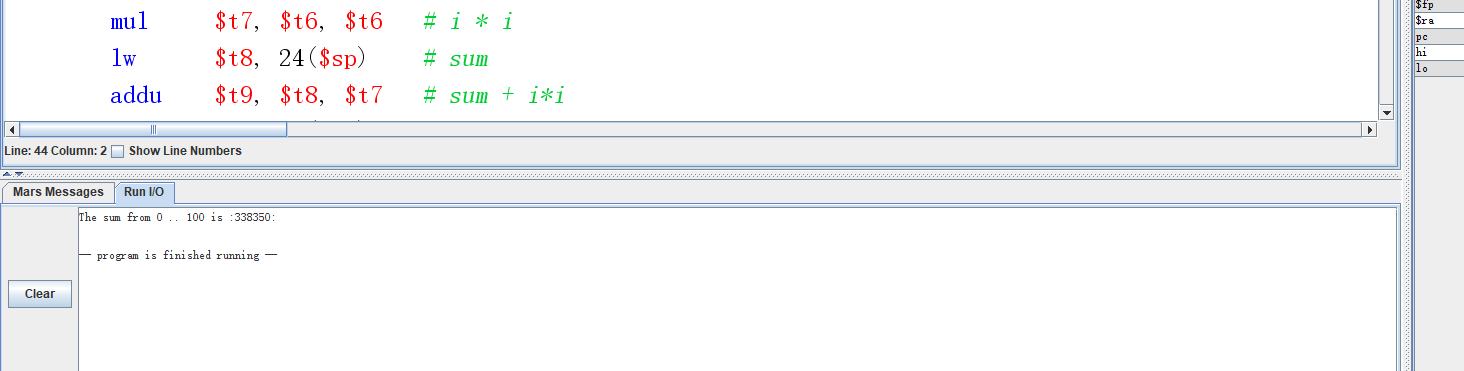
**Section 002**

**Instructor: Dr. Mohamed Almekkawy**

**Lab 2**

**Produced by Hongshuo Wang**

1. **When you run the previous program, what is printed?**

The sum from 0 .. 100 is :338350:

1. **What is the Value in register $t7 (in decimal) when the program ends?**

0X00002710 (Hex)

 = 10000 (decimal)

1. **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 register 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.)**

**$t6** begins at 0x00000000 and each loop it increases by 1. According to comment, we set $t6 to be index i which denote the number of looping. Hence after each loop it increase one.

Also, by the comment **$t0** which is also start at 0x00000000, increase 1 each loop before $t1 increasing because we assign it to be i + 1.

According to the formula, **$t7** is decided by the $t6, which equals to ($t6) ^ 2

Also, **$t8** is the sum of all the previous $t7, which lw previous loop’s $t9.

**$t9** is the sum of all the $t7 and is calculated by $t8 + $t7, including the current $t7 which we just figure out.

Moreover, **$at** is also changing when we use pseudo code. That is what supposed to do.

**$pc** changes start at 0x0040002C, increase 4 after each instruction executed and finally goes back to 0x004002C. That is his duty which is always points next instruction needs to be executed.

**$lo** equals to $7 and always change after 13#line because of pseudo code mul, put the mult result in lo register.