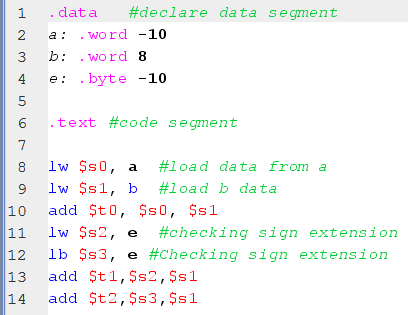
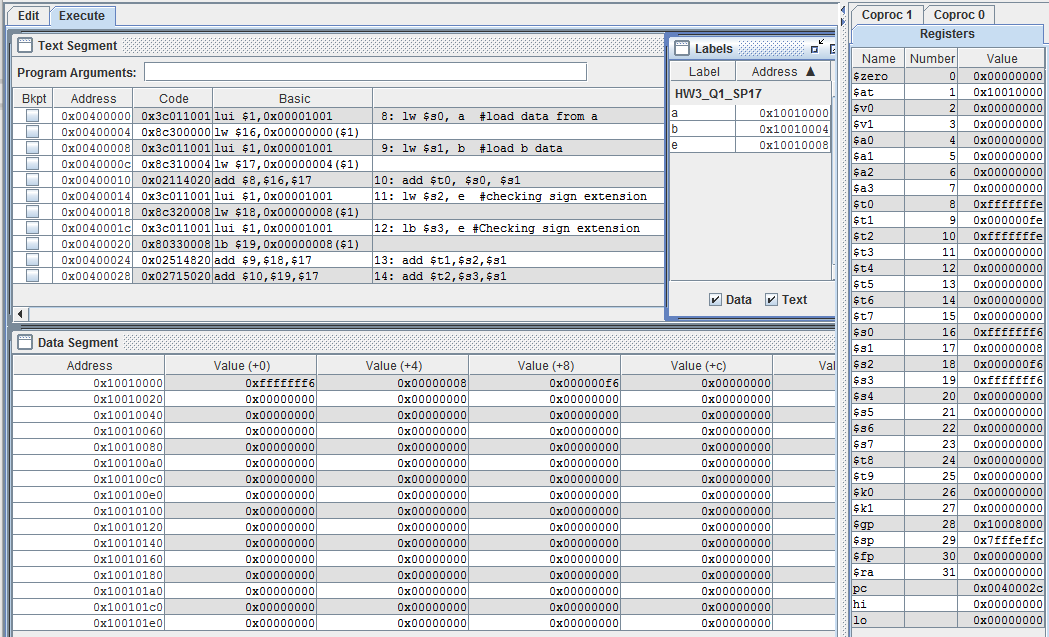
**HW#4 (CSC390-Spring 2018)**

**Due: 02/16/2018 by 11:00PM (Blackboard Submission)**

Q1. Consider the following MIPS assembly codes which perform a simple mathematical operation (-10+8) using the variables a, b, and e. The value -10 is assigned to variables a and e, as shown in lines 2 and 4, respectively.



When we assembled the program, the first three locations of the Data-Segment are initialized with the values of a, b, and e, as shown in the following figure. Explain the questions on the next page.



1. Observe that for the same value of -10 the first location of the Data-Segment has 0xfffffff6 and the third location has 0x000000f6. Explain why?
2. Line 10 of the code is performing (-10+8) operation. Check whether you are getting the correct result in $t0.
3. Lines 11 and 12 are loading the value of the variable “e” into the registers $s2 and $s3 respectively. Observe that after executing the program $s2 has 0x000000f6 and $s3 has 0xfffffff6. Explain Why?
4. Observe that lines 13 and 14 are performing the same mathematical operation, (-10+8), and storing the results in $t1 and $t2, respectively. Check which one has the correct result. Explain why?

**Q2.** Write a MIPS assembly code to transfer data from register $s0 to $t0 without using Load and Store instructions.

**Q3.** Write down the **machine code** of the following R-format instruction showing every instruction fields.

**add $t3, $S3, $S4**

Verify your machine code using the MARS simulator.

**Q4.** Write a MIPS assembly language program that calls a procedure, Add\_Sub\_Mul, which   
 accept four parameters (g,h,i,j) and returns,

f = (g+h) if i >j; f= (g-h) if i < j; and f = g\*h if i == j; the equivalent C function is shown below:

int Add\_Sum\_Mul (int g, int h, int i, int j) {

int f;

if (i > j) {

f=(g+h);}

else if (i < j) {

f=(g-h);}

else if (i==j){

f=g\*h}

}

Consider the variables g, h, i, j and f are initialized with some initial values in the data segment. Use $s0 as f in the function and also use $s0 to store the base address of f in the memory location. Clearly comment on the every instruction you use in your program. Specially, clearly show and describe the stack operation. Remember, resisters ($a0-$a2) are used for passing arguments in to the function and $v resisters are used to store the results in the function.