**HW#4 (CSC390); Turn in your code on the Blackboard by** 03/11/2016 by 5:00PM

**#Q1.** Consider the following two C procedures (swap and sort) and their corresponding MIPS assembly codes as shown in the figures 1 and 2, respectively. Using these two C procedures (swap and sort) write a MIPS assembly program that will sort the following array elements in the **ascending order**.

Array= [100 50 75 -1 -50 500 20 40 40 17 19 23 5 7 -20]

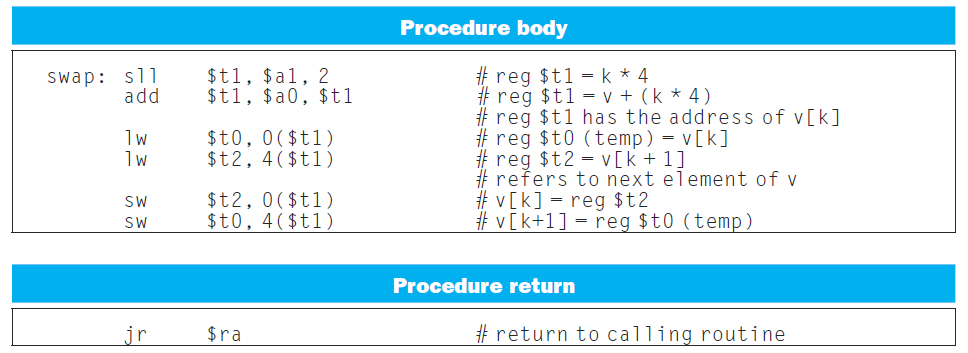
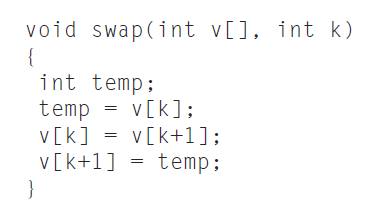
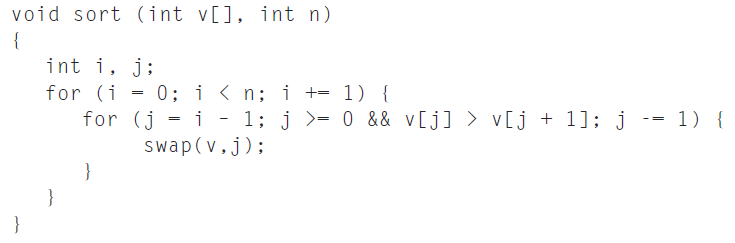


Figure MIPS assembly code for swap procedure



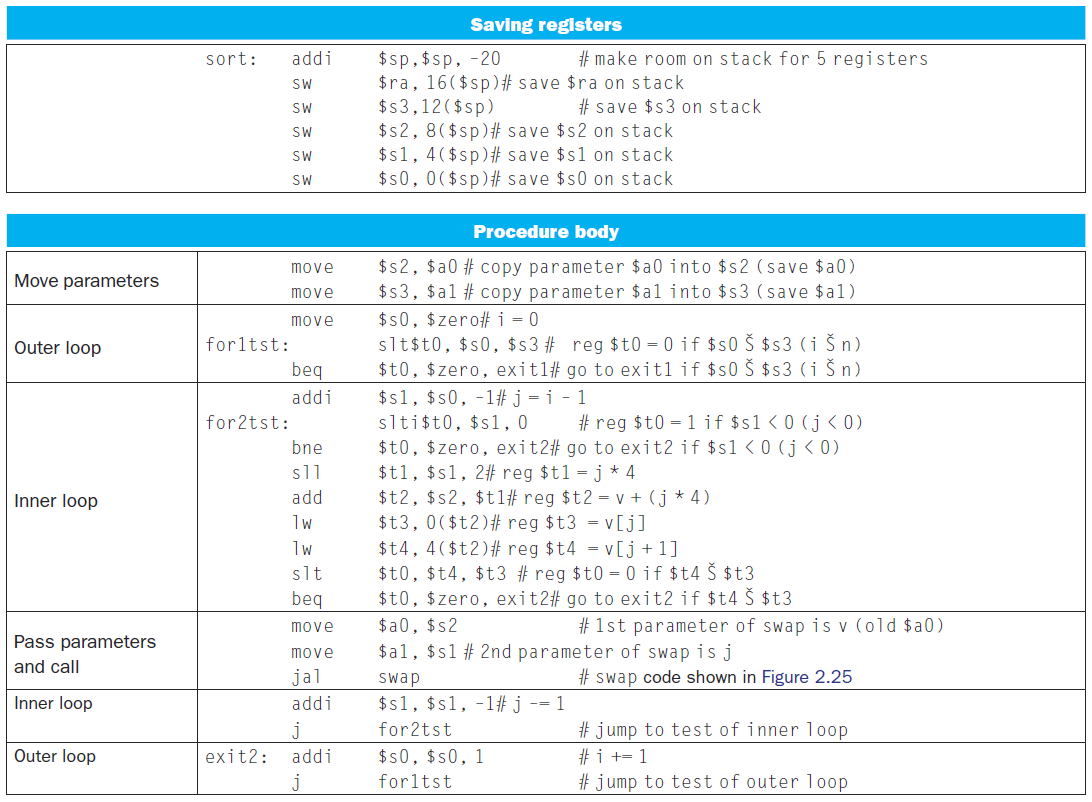
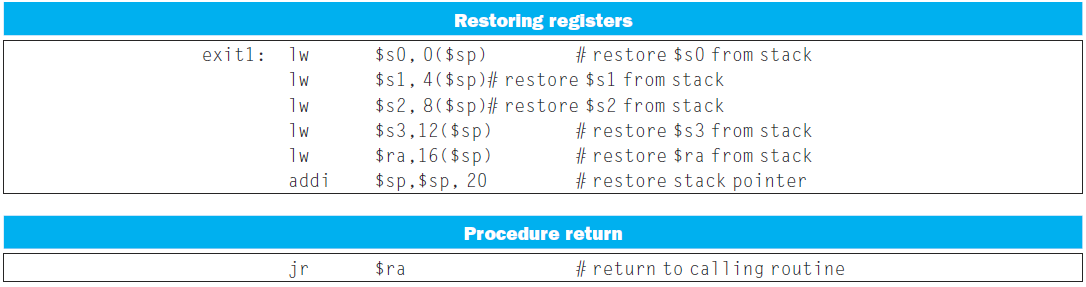
 

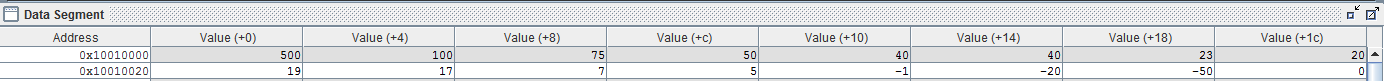
Figure MIPS assembly version of sort procedure

# What chages you would make to sort the array elements in **descending order**.

Results (ascending order) :



Results (descending order) :



**Q2.** (Read chapter#3 (pages 178-182) of your text book before you solve the following questions)

1. MIPS assembler generates an exception error when overflow occurs in mathematical operation of signed numbers. Write a MIPS assembly language program that will check overflows during the element by element addition between the array elements (signed numbers) of two vectors A and B and store the results in C vector. If overflow occurs the code will store 0xFFFFFFFFhex in the corresponding memory location of the C vector, otherwise store the actual addition results.

Consider the following vectors A and B and store the results C vectors in your coding:

A = [1000000000, 2000000000, 2000000000, -1000000000, -2000000000];

B= [1000000000, -1000000000, 1000000000, -1000000000, -1000000000];

C=[. . . …. …. … ];

1. Write a MIPS assembly language program that will check overflows during the element by element addition between the array elements (unsigned numbers) of two vectors A and B and store the results in C vector. If overflow occurs the code will store 0x00000000 in the corresponding memory location of the C vector, otherwise store the actual addition results.

Consider the following vectors A and B and store the results C vectors in your coding:

A = [1000000000, 2000000000, 2000000000, 1000000000, 2000000000];

B = [3000000000, 3000000000, 1000000000, 4000000000, 2000000000];

C=[. . . …. …. … ];

**Note: Please turn in your Code on the Blackboard by 03/11/2016 by 5:00PM**