Lab Exercise #11 -- SPARC Data Movement and Organization

A. Consider the following SPARC assembly language program.

1. In the spaces provided, give the hexadecimal contents (each byte) of the requested memory locations and registers after the instructions are executed.

```
================================================================================
        .global  main

        .section ".data"
        .align   4
nums:
        .byte    0x93, 0x2e
        .half    0x3a67, 0xba98, 0x7654
store:
        .word    0xffffffff, 0xeeeeeeee, 0xdddddddd

        .section ".text"
        .align   4
main:
        save     %sp, -96, %sp

        set      0x12345678, %l7         ! 0x12345678 --> %l7
        set      store, %l6              ! address of store --> %l6

        st       %l7, [%l6+0]            ! store word
        sth      %l7, [%l6+4]            ! store halfword
        stb      %l7, [%l6+8]            ! store byte

!  store+0:  _______  _______  _______  _______
!
!  store+4:  _______  _______  _______  _______
!
!  store+8:  _______  _______  _______  _______

        set      nums, %l5               ! address of nums --> %l5

        ld       [%l5+4], %l0            ! load word
        ldsh     [%l5+4], %l1            ! load signed halfword
        lduh     [%l5+4], %l2            ! load unsigned halfword
        ldsb     [%l5+4], %l3            ! load signed byte
        ldub     [%l5+4], %l4            ! load unsigned byte

!  l0:  _______  _______  _______  _______
!
!  l1:  _______  _______  _______  _______
!
!  l2:  _______  _______  _______  _______
!
!  l3:  _______  _______  _______  _______
!
!  l4:  _______  _______  _______  _______

        ret
        restore
================================================================================
```

2. Check your answers by executing the assembly language program. The source file for this program is named "~cse320/Labs/lab11.partA.s". Copy the source code into your account, then add the appropriate statements to call "memory" and "iu_window" in order to display the contents of a selected area of memory and the IU registers.

The following commands will assemble, link and execute the program:

```
<prompt> gcc lab11.partA.s ~cse320/lib/reglib.o
<prompt> a.out
```

If any of your answers are incorrect, re-do your work to make sure that you understand the operations.

B. Working with arrays of characters.

The source file named "~cse320/Labs/lab11.partB.s" contains a program to read a sequence of characters from standard input and store the characters in an array. The program uses the C library functions "getchar" and "printf" to handle communication with the user.

1. Copy the source file into your account and examine the program. Assemble and execute the program, then answer each of the following questions.

a) Give the hexadecimal value of each of the following symbolic constants:

EOL _____ SIZE _____

b) Why is space for the formatting strings ("fmt1" and "fmt2") reserved in the ".text" section, while space for the character sequences ("original" and "copy") is reserved in the ".data" section?

c) The loop to read characters from standard input may terminate under two different circumstances. What are those two circumstances, and why are both comparisons necessary?

2. Modify the program as per the comments in the source code.