Lab Exercise #12 -- SPARC Floating Point Operations

A. The files named "lab12.A.main.c" and "lab12.A.sub.s" contain the source code modules for a program which performs some floating point calculations.

1. Examine the contents of "lab12.A.main.c", then answer the questions below.

a) How many bytes are allocated for array "vec"? ____________

b) How many bytes are allocated for one element in array "vec"? ____________

c) What is the value of "N" when the program executes? ____________

2. Examine the contents of "lab12.A.sub.s", then answer the questions below.

a) Describe the general purpose of function "sum_vector".

____

____

b) Describe the information which is passed into function "sum_vector".

____

____

____

____

c) Describe the information which is returned by function "sum_vector".

____

____

d) For each of the indicated assembly language instructions in the listing of function "sum_vector" on the next page, provide a meaningful comment.

3. Translate and execute the program using the commands:

```
   <prompt> gcc ~cse320/Labs/lab12.A.main.c ~cse320/Labs/lab12.A.sub.s
   <prompt> a.out
```

Revise any of your responses above that are incorrect.

B. The files "lab12.B.main.c" and "lab12.B.sub.s" contain the outline of an incomplete program which uses floating point operations. Copy the assembly language source file into your account and complete the program, using the comments as a guide to your modifications.

Test your program using the following input values:

```
   1, -5, 6          (roots are  3.000000 and  2.000000)
   1, 0, -4          (roots are  2.000000 and -2.000000)
   3.7, 16.5, 1.7    (roots are -0.105527 and -4.353932)
```

```
! Function sum_vector

        .global  sum_vector

        .section ".text"
        .align   4
sum_vector:
        save     %sp, -96, %sp     ! __________________________________________

        set      zero, %l7         ! __________________________________________

        ldd      [%l7], %f0        ! __________________________________________

        ldd      [%l7], %f30       ! __________________________________________

        clr      %l1               ! __________________________________________

        clr      %l2               ! __________________________________________

        clr      %l3               ! __________________________________________
loop:
        cmp      %l1, %i1          ! __________________________________________

        bge      endloop           ! __________________________________________
        nop

        smul     %l1, 8, %l0       ! __________________________________________

        ldd      [%i0+%l0], %f4    ! __________________________________________

        faddd    %f0, %f4, %f0     ! __________________________________________
if:
        fcmpd    %f4, %f30         ! __________________________________________
        nop
        fbge     else              ! __________________________________________
        nop
then:
        inc      %l2               ! __________________________________________

        ba       endif             ! __________________________________________
        nop
else:
        inc      %l3               ! __________________________________________
endif:
        inc      %l1               ! __________________________________________

        ba       loop              ! __________________________________________
        nop
endloop:
        st       %l2, [%i2]        ! __________________________________________

        st       %l3, [%i3]        ! __________________________________________

        ret
        restore                    ! __________________________________________

        .align   8                 ! __________________________________________
zero:
        .double  0r0.0             ! __________________________________________
```