Lab Exercise #10 -- SPARC Subprograms

This exercise develops familiarity with SPARC assembly language subprograms.

A. Consider the SPARC assembly language functions given as source code in the file "~cse320/Labs/lab10.program.s", and the output from that program.

The program was assembled and linked using:

```
<prompt> gcc lab10.program.s ~cse320/lib/reglib.o
```

The program output is displayed below.

```
IU WINDOW          y: 00000000  psr: fe401000  icc: 4 (-Z--)
   g0: 00000000   o0: 00000000   l0: 00010b50   i0: 00011010
   g1: ffffffff   o1: 00000000   l1: 00010c14   i1: 00010c44
   g2: 00000000   o2: 00000000   l2: 00021a98   i2: ffbff72c
   g3: 00000000   o3: 00000000   l3: 00021a9c   i3: 00021af8
   g4: 00000000   o4: 00000000   l4: 00021abc   i4: 00000000
   g5: ffffffff   o5: 00000000   l5: 00010b10   i5: ff1f0140
   g6: 00000000   o6: ffbff660   l6: 00010b2c   i6: ffbff6c0
   g7: ff1f2a00   o7: 00010ba4   l7: 00010b44   i7: 000109a8

MEMORY CONTENTS [00010b10 to 00010c5f]
  00010b10: 9d e3 bf a0 40 00 01 3f 01 00 00 00 ae 10 00 00
  00010b20: a0 10 00 18 a2 10 00 19 a4 10 00 00 80 a4 80 11
  00010b30: 16 80 00 05 01 00 00 00 a4 04 a0 01 10 bf ff fc
  00010b40: 01 00 00 00 b0 10 00 17 81 c7 e0 08 81 e8 00 00
  00010b50: 9d e3 bf a0 21 00 00 42 a0 14 23 50 23 00 00 43
  00010b60: a2 14 60 14 25 00 00 86 a4 14 a2 98 27 00 00 86
  00010b70: a6 14 e2 9c 29 00 00 86 a8 15 22 bc 2b 00 00 42
  00010b80: aa 15 63 10 2d 00 00 42 ac 15 a3 2c 2f 00 00 42
  00010b90: ae 15 e3 44 31 00 00 44 b0 16 20 10 33 00 00 43
  00010ba0: b2 16 60 44 40 00 01 1b 01 00 00 00 11 00 00 42
  00010bb0: 90 12 23 10 92 02 21 50 40 00 00 23 01 00 00 00
  00010bc0: 21 00 00 86 a0 14 22 9c 23 00 00 86 a2 14 62 bc
  00010bd0: a4 24 40 10 a7 3c a0 02 90 10 00 10 92 10 00 13
  00010be0: 7f ff ff cc 01 00 00 00 29 00 00 86 a8 15 22 98
  00010bf0: d0 25 00 00 11 00 00 43 90 12 20 14 92 10 00 13
  00010c00: d4 05 00 00 40 00 43 4f 01 00 00 00 81 c7 e0 08
  00010c10: 81 e8 00 00 0a 54 68 65 20 73 75 6d 20 6f 66 20
  00010c20: 74 68 65 20 25 64 20 65 6c 65 6d 65 6e 74 73 20
  00010c30: 69 6e 20 74 68 65 20 61 72 72 61 79 20 69 73 20
  00010c40: 25 64 0a 00 9d e3 be 90 c1 3f bf 00 c5 3f bf 08
  00010c50: c9 3f bf 10 cd 3f bf 18 d1 3f bf 20 d5 3f bf 28

IU WINDOW          y: 00000000  psr: fe401000  icc: 4 (-Z--)
   g0: 00000000   o0: 00011010   l0: 00021a9c   i0: 00021a9c
   g1: ffffffff   o1: 00010c44   l1: 00021abc   i1: 00000008
   g2: 00000000   o2: ffbff72c   l2: 00000020   i2: 00000000
   g3: 00000000   o3: 00021af8   l3: 00000008   i3: 00000000
   g4: 00000000   o4: 00000000   l4: 00021abc   i4: 00000000
   g5: ffffffff   o5: ff1f0140   l5: 00010b10   i5: 00000000
   g6: 00000000   o6: ffbff600   l6: 00010b2c   i6: ffbff660
   g7: ff1f2a00   o7: 00010b14   l7: 00010b44   i7: 00010be0

   The sum of the 8 elements in the array is 0
```

1. Complete the following table to show the actual address for some of the symbolic addresses defined (or used) in functions "main" and "sum_list".

symbol	actual address	symbol	actual address
main	_____	sum_list	_____
format	_____	loop	_____
total	_____	endloop	_____
list	_____	iu_window	_____
endlist	_____	memory	_____

2. Give the relative order in which the four functions ("main", "sum_list", "iu_window", and "memory") appear in memory by listing the function names in order, from first to last.

3. In the output produced by the first call to "iu_window" (printed on the first page), locate and label each of the following items:

   a. The address (in a register) from which "iu_window" was called.
   b. The address (in a register) from which "main" was called.

4. In the output produced by "memory" (printed on the first page), locate and label each of the following items:

   a. The ".text" section of function "main".
   b. The ".text" section of function "sum_list".
   c. The first instruction in function "memory".
   d. Each branch or call instruction in function "main".
   e. Each branch or call instruction in function "sum_list".

5. In the output produced by the second call to "iu_window" (printed on the first page), locate and label each of the following items:

   a. Each argument (in a register) to function "sum_list".
   b. The address (in a register) from which "iu_window" was called.
   c. The address (in a register) from which "sum_list" was called.

B. Revise the source code to complete the program.

Copy the source code file into your directory and revise the program so that function "sum_list" correctly computes the sum of the array elements.

Assemble, link and execute your program to prove that it executes correctly.