

Problem 1: Compare the coding of the DLX instructions:

```
add  r1, r2, r3
addi r4, r5, #6
```

to the corresponding Sun SPARC V8 instructions:

```
add %g3, %g2, %g1    ! g1 = g2 + g3
add %g5, 6, %g4      ! g4 = g5 + 6
```

The definition of the SPARC V8 architecture is available via <http://www.ee.lsu.edu/ee4720/sam.pdf> or <http://www.sparc.com/standards/V8.pdf>. *Hint: The information needed to solve the problem is in Appendix B.*

How are the approaches used to code immediate variants of the add instructions different in the two ISAs?

Problem 2: DLX does not have indexed addressing nor does it have autoincrement addressing. Suppose one wanted to include those addressing modes in an extended version of DLX, call it DLX-BAM (better addressing modes). The addressing modes would be used in load and store instructions. Show how they would best be coded, where the fewer changes to the coding structure the better. (For example, adding a fourth instruction type [say Type-A], would be a big change and so would be bad.) Sample mnemonics for these instructions appear below:

```
! Indexed addressing.
lw r1, (r2+r3) ! r1 = MEM[ r2 + r3 ];
sw (r2+r3), r4 ! MEM[ r2 + r3 ] + r4;

! Autoincrement addressing.
lb r1, 3(+r2) ! r1 = MEM[ r2 + 3 ];   r2 = r2 + 1;
lw r4, 8(+r5) ! r4 = MEM[ r5 + 8 ];   r5 = r5 + 4;
sw 4(+r7), r8 ! MEM[ r7 + 4 ] = r8;   r7 = r7 + 4;
```

Problem 3: Write a C program that does the same thing as the DLX program below.

```
! r2: Start of table of indices, used to retrieve elements
!   from the character table.
! r4: Start of table of characters.
! r6: Location to copy characters to.
! r8: Address of end of index table.
```

LOOP:

```
lw r1, 0(r2)
add r3, r1, r4
lb  r5, 0(r3)
sb  0(r6), r5
addi r2, r2, #4
addi r6, r6, #1
slt r7, r2, r8
bneq r7, L00P
```

Solution template available via: <http://www.ee.lsu.edu/ee4720/2000f/hw02.c>

```
void
untangle(int *r2, char *r4, char *r6, int *r8)
{
    /* Put solution here. */
}
}
```

Problem 4: Re-code the DLX program above using DLX-BAM, taking advantage of the new instructions.