EE 4720 **Homework 1** Solution **Due: 1 September 2000**

Problem 1: Find the SPECint2000 results for the API UP2000 750 MHz processor, it can be found at the `http://www.spec.org` web site. This processor has a SPECint2000 rating of 456. Find another processor with a slower rating but for which individual benchmarks are faster. (Look for different CPU families.) How many of the benchmarks are faster on the slower processor?

Problem 2: Write a DLX assembly language program to convert a string of characters to lower case. The string is `NULL`-terminated (the character following the end of the string is a zero). Register `r1` contains the address of the start of the string. Any register can be modified. The code for an upper-case A is 65 and the code for a lower-case a is 97. Modify the string, do not create a new one.

```
! ** Solution **
!
! Register r1 contains address of first character of string.
LOOP:
lbu r2, 0(r1)
beqz r2, DONE
slti r3, r2, #65
bneq r3, CONTINUE
sgti r3, r2, #90
bneq r3, CONTINUE
addi r3, r3, #32
sb 0(r1), r3
CONTINUE:
addi r1, r1, #1
j LOOP
DONE:
```

Problem 3: Write a DLX assembly language program that loads an element of a two-dimensional array to a register.

Register `r1` holds address of the start of the array, register `r2` holds the row of the element to retrieve, and register `r3` holds the column of the element to retrieve. Put the retrieved element in f0. The array dimensions are 256 rows $\times$ 1024 columns. Each element of the array is a double precision floating point number.

Elements are arranged in memory in the following order:

$$a_{0,0}\, a_{0,1}\, a_{0,2} \cdots a_{1,0}\, a_{1,1}\, a_{1,2} \cdots a_{2,0} \cdots$$

where $a_{i,j}$ is the element at row i, column j.

```
! ** Solution **
!
! r1: address of the start of the array.
! r2: row of element to retrieve.
! r3: column of element to retrieve.
! Put element in f0.
! Array dimensions are 256 rows x 1024 columns
! Each element of the array is a double precision floating point number.
! Elements are arranged in memory in the following order
 a_{0,0} a_{0,1} a_{0,2} ... a_{1,0} a_{1,1} a_{1,2} ... a_{2,0} ...
 where a_{i,j} is the element at row i and column j.

slli r4, r2, #10
or   r4, r4, r3
slli r4, r4, #3
add  r4, r4, r1
ld   f0, 0(r4)
```