CMPUT 229 Computer Organization and Architecture I
Midterm Examination
(February 28, 2001)

Name: _____ SID: _____

Do all problems. Closed book and notes. No calculator. Instruction encoding sheet is okay.

1. Express (-25) in 16-bit signed 2's complement representation. Give the answer in hexadecimal.

2. If the ASCII code for character 'a' is 0x61, then that of character 'z' is:

3. If `0x41ac0000` represents an IEEE single precision floating point number, then its decimal value is:

4. Determine the encoding of the assembly instruction `sll $8, $9, 5`. Express your answer in hexadecimal.

5. What does the word `0x8e25fff0` represent as a MIPS assembly instruction?

6. Name any four of the five addressing modes in MIPS.

7. Assuming `$t0 = 0x15` and `$t1 = -4`, what will be the results in `$Hi` and `$Lo` when the instruction `mult $t0, $t1` is executed? Express your answer in hexadecimal.

8. Assuming `$t0 = 0x35` and `$t1 = -4`, what will be the results in `$Hi` and `$Lo` when the instruction `div $t0, $t1` is executed? Express your answer in hexadecimal.

9. Assuming `$t0 = 0x35`, what will be result in `$t1` when the instruction `sll $t1, $t0, 3` is executed. Express your answer in hexadecimal.

10. Use the real MIPS instructions of `beq`, `bne`, and `slt` to derive the minimum implementation of the pseudo-instruction `ble $t1, $t0, there`.

11. If you use the instruction `j there` in your program, and if this instruction is placed at `0x10010020` in memory, what is the largest address location that the label "`there`" can be?

12. Assume `array` is a global array of integers, `tmp` is a global integer, and variable `j` is represented by `$t0`. ("Global" means their storage is allocated with assembler directives `.word` or `.space`.) Translate the following C instruction into MIPS assembly language.

```
array[j] = array[j-1];
```

13. Implement the following two C functions in MIPS assembly language. In `swap()`, assume that input arguments `a` and `b` are passed in `$a0` and `$a1`, respectively, and use `$t0` as the local variable `i`.

```
int i = 4, j = 9;
main()
{
   swap(&i, &j);
}
```

```
swap(int *a, int *b)
{   int i;
    i = *a; *a = *b; *b = i;
}
```