Name:

Student ID:

Chapter 2-2

Exercise 1:

[5] <§2.4> Why doesn't MIPS have a subtract immediate instruction?

Exercise 2:

[10] <§2.6> Construct a control flow graph (like the one shown in Fig. 2.11) for the following section of C or Java code:

```
for (i=0; i<x; i=i+1)
    y = y + i;
```

Exercise 3:

[5] <§2.9> Show the single MIPS instruction or minimal sequence of instructions for this C statement:

```
b = 25 | a;
```

Assume that `a` corresponds to register `$t0` and `b` corresponds to register `$t1`.

Exercise 4:

In the following problems, the data table contains the values for registers `$t0` and `$t1`. You will be asked to perform several MIPS logical operations on these registers.

a.	`$t0 = 0xAAAAAAAA, $t1 = 0x12345678`
b.	`$t0 = 0xF00DD00D, $t1 = 0x11111111`

4.1

For the lines above, what is the value of `$t2` for the following sequence of instructions?

sll $t2, $t0,4

or $t2, $t2, $t1

4.2

For the values in the table above, what is the value of `$t2` for the following sequence of instructions?

```
sll $t2, $t0, 4
andi $t2, $t2, -1
```

-1 is replaced by 0x FFFF and ignore the overflow

4.3

For the lines above, what is the value of `$t2` for the following sequence of instructions?

```
srl $t2, $t0, 3
andi $t2, $t2, 0xFFEF
```

In the following exercise, the data table contains various MIPS logical operations. You will be asked to find the result of these operations given values for registers `$t0` and `$t1`.

a.	`sll $t2, $t0, 1` `andi $t2, $t2, -1`
b.	`andi $t2, $t1, 0x00FC` `srl $t2, 2`

-1 is replaced by 0x FFFF and ignore the overflow

4.4

Assume that `$t0` = 0x0000A5A5 and `$t1` = 00005A5A. What is the value of `$t2` after the two instructions in the table?

4.5

Assume that `$t0` = 0xA5A50000 and `$t1` = A5A50000. What is the value of `$t2` after the two instructions in the table?

4.6

Assume that `$t0` = 0xA5A5FFFF and `$t1` = A5A5FFFF. What is the value of `$t2` after the two instructions in the table?

Exercise 5:

The following program tries to copy words from the address in register `$a0` to the address in register `$a1`, counting the number of words copied in register `$v0`. The program stops copying when it finds a word equal to 0. You do not have to preserve the contents of registers `$v1`, `$a0`, and `$a1`. This terminating word should be copied but not counted.

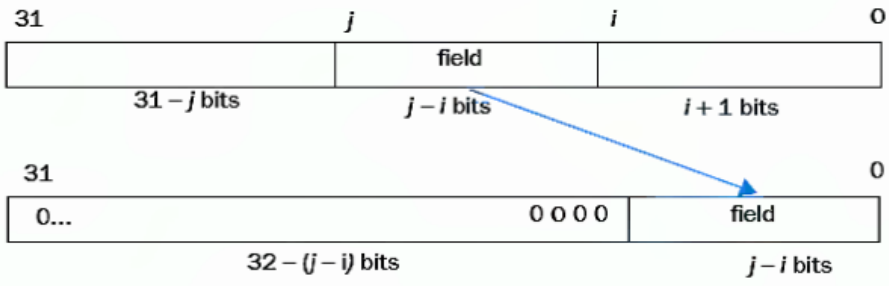
```
      addi $v0, $zero, 0 # Initialize count
loop: lw   $v1, 0($a0)   # Read next word from source
      sw   $v1, 0($a1)   # Write to destination
      addi $a0, $a0, 4   # Advance pointer to next source
      addi $a1, $a1, 4   # Advance pointer to next destination

      beq  $v1, $zero, loop  # Loop if word copied != zero
```

There are multiple bugs in this MIPS program; fix them and turn in a bug-free version.

Exercise 6:

Some computers have explicit instructions to extract an arbitrary field from a 32-bit register and to place it in the least significant bits of a register. The figure below shows the desired operation:



Find the shortest sequence of MIPS instructions that extracts a field for the constant values $i = 5$ and $j = 22$ from register `$t3` and places it in register `$t0`. (Hint: It can be done in two instructions.)