CMPEN 331 – Computer Organization and Design,
Chapter 2 Review Questions

1. For the following MIPS assembly instructions above, what is a corresponding C statement? `f`, `g`, `h` and `i` are assigned to registers `$s0`, `$s1`, `$s2` and `$s3`, respectively

```
add  f, g, h
add  f, i, f
```

2. For the following C statement, what is the corresponding MIPS assembly code? Assume that the variables `i`, and `j` are assigned to registers `$s3`, and `$s4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

```
B[8] = A[i – j];
```

5 Translate the following MIPS code to C. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

```
addi $t0, $s6, 4
add  $t1, $s6, $0
sw   $t1, 0($t0)
lw   $t0, 0($t0)
add  $s0, $t1, $t0
```

6 Assume that registers $s0 and $s1 hold the values 0x80000000 and 0xD0000000, respectively.

a. What is the value of $t0 for the following assembly code?

```
add $t0, $s0, $s1
```

b. Is the result in $t0 the desired result, or has there been overflow?

c. For the contents of registers $s0 and $s1 as specified above, what is the value of $t0 for the following assembly code?

```
sub $t0, $s0, $s1
```

d. Is the result in $t0 the desired result, or has there been overflow?

7 Provide the type and hexadecimal representation of following instruction:

```
sw $t1, 32($t2)
```

8 Find the MIPS instructions that extracts bits 16 down to 11 from register $t0 and uses the value of this field to replace bits 31 down to 26 in register $t1 without changing the other 26 bits of register $t1.

9 For the following C statement, write a sequence of MIPS assembly instructions that does the identical operation. Assume $t1 = A, $t2 = B, and $s1 is the base address of C.

```
A = C[0] << 4;
```

10. Assume $t0 holds the value 0x00101000. What is the value of $t2 after the following instructions?

```
 slt  $t2, $0, $t0
 bne  $t2, $0, ELSE
 j    DONE
ELSE: addi $t2, $t2, 2
DONE:
```

11. Write the MIPS assembly code that creates the 32-bit constant

$0010\ 0000\ 0000\ 0001\ 0100\ 1001\ 0010\ 0100_{two}$

and stores that value to register $t1.

Solutions

1. For the following MIPS assembly instructions above, what is a corresponding C statement? `f`, `g`, `h` and `i` are assigned to registers `$s0`, `$s1`, `$s2` and `$s3`, respectively

```
add  f, g, h
add  f, i, f
```

```
f = g + h + i
```

2. For the following C statement, what is the corresponding MIPS assembly code? Assume that the variables `i`, and `j` are assigned to registers `$s3`, and `$s4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

```
B[8] = A[i - j];
```

```
sub $t0, $s3, $s4
sll    $t0, $t0, 2
add $t1, $s6, $t0
lw $t2, 0($t1)
sw $t2, 32($s7)
```

3. Assume that registers $s0 and $s1 hold the values 0x80000000 and 0xD0000000, respectively.
   a. What is the value of $t0 for the following assembly code?

```
add $t0, $s0, $s1
```

```
0x50000000
```

   b. Is the result in $t0 the desired result, or has there been overflow?

```
Overflow
```

c. For the contents of registers $s0 and $s1 as specified above, what is the value of $t0 for the following assembly code?

```
sub $t0, $s0, $s1
```

B0000000

d. Is the result in $t0 the desired result, or has there been overflow?

no overflow

4. Provide the type and hexadecimal representation of following instruction:

```
sw $t1, 32($t2)
```

i-type, 0xAD490020

5. Translate the following MIPS code to C. Assume that the variables `f`, `g`, `h`, `i`, and `j` are assigned to registers `$s0`, `$s1`, `$s2`, `$s3`, and `$s4`, respectively. Assume that the base address of the arrays `A` and `B` are in registers `$s6` and `$s7`, respectively.

```
addi $t0, $s6, 4
add  $t1, $s6, $0
sw   $t1, 0($t0)
lw   $t0, 0($t0)
add  $s0, $t1, $t0
```

```
f = 2*(&A);
```

6. Find a sequence of MIPS instructions that extracts bits 16 down to 11 from register $t0 and uses the value of this field to replace bits 31 down to 26 in register $t1 without changing the other 26 bits of register $t1.

```
srl $t0, $t0, 11
sll $t0, $t0, 26
ori $t2, $0, 0x03ff
sll $t2, $t2, 16
ori $t2, $t2, 0xffff
```

```
and $t1, $t1, $t2
or $t1, $t1, $t0
```

7. For the following C statement, write a minimal sequence of MIPS assembly instructions that does the identical operation. Assume $t1 = A, $t2 = B, and $s1 is the base address of C.

```
A = C[0] << 4;
```

```
lw $t3, 0($s1)
sll $t1, $t3, 4
```

8. Assume $t0 holds the value 0x00101000. What is the value of $t2 after the following instructions?

```
 slt  $t2, $0, $t0
 bne  $t2, $0, ELSE
 j    DONE
ELSE: addi $t2, $t2, 2
DONE:
```

```
$t2 = 3
```

9. Write the MIPS assembly code that creates the 32-bit constant

$0010\ 0000\ 0000\ 0001\ 0100\ 1001\ 0010\ 0100_{two}$

and stores that value to register `$t1`.

```
Generally, all solutions are similar:
lui $t1, top_16_bits
ori $t1, $t1, bottom_16_bits
```