CS2100 Tutorial 3

AY 24/25 Sem 2 — github/omgeta

Q1. (a.) MIPS:

```
        add   $s0,   $zero,  $zero #init vars
        addi  $s1,   $s5,    -1
        addi  $s3,   $zero,  -1

Loop:   slt   $t0,   $s0,    $s1
        and   $t0,   $zero,  Exit
        beq   $s3,   $zero,  Exit

        add   $t1,   $s4,    $s0   #if statement
        add   $t2,   $s4,    $s1
        lb    $t3,   0($t1)
        lb    $t4,   0($t2)
        beq   $t3,   $t4,    Else
        add   $s3,   $zero,  $zero
        j     End
Else:   addi  $s0,   $s0,    1     #else
        addi  $s1,   $s1,    -1
End:    j     Loop

Exit:
```

(b.) MIPS:

```
        add   $s0,   $zero,  $zero #init vars
        addi  $s1,   $s5,    -1
        addi  $s3,   $zero,  -1
        add   $t1,   $s4,    $s0   #str[lo]
        add   $t2,   $s4,    $s1   #str[hi]

Loop:   slt   $t0,   $t1,    $t2
        beq   $t0,   $zero,  Exit
        beq   $s3,   $zero,  Exit

        lb    $t3,   0($t1)
        lb    $t4,   0($t2)
        beq   $t3,   $t4,    Else
        add   $s3,   $zero,  $zero
        j     End
Else:   addi  $t1,   $t1,    1     #start++
        addi  $t2,   $t2,    -1    #end--
End:    j     Loop

Exit:
```

Q2. (a.) addi $s1, $zero, 0 = 001000 00000 10001 0000.. = 0x20110000
0x11000002 = 000100 01000 00000 0..010 = beg $t0, $zero, exit
0x22310001 = 001000 10001 10001 0..001 = addi $s1, $s1, 1
j loop = j 0x0040002c = 000010 0000 0100 0000... 1011 = 0x0810000b

Instruction Encoding	MIPS Code
	# **$s1** stores the result, **$t0** stores a non-negative number
0x20110000	**addi $s1, $zero, 0** #Inst. address is 0x00400028
0x00084042	**loop: srl $t0, $t0, 1**
0x11000002	**beq $t0, $zero, exit**
0x22310001	**addi $s1, $s1, 1**
0x0810000B	**j loop**
	exit:

(b.) $s1 = floor(log($t0))

Q3. (a.)
```
srl $s4, $s4, 1
lw $t1, 0($t0)
lw $t1, 0($t0)
slt $t9, $t1, $s1
beq $t9, $zero, equal
j end
j loop
```

(b.) 16

(c.) j loopEnd = j 0xFFFFFF44 = 0x0BFFFFD1

(d.) No. Jump takes the direct partial jump target address so two jumps of the same target give the same instruction.