QUESTION ONE:
A 32 bit word can represent a MIPS instruction. How can we tell if that instruction is of type R, J, or I?

QUESTION TWO:
Below is a pretty neat function (subroutine) written in MIPS that's *almost* ready for prime time – but eight key lines of code are missing. Get busy!

```
sum_tree:
        ______________________
        ______________________
        ______________________
        ______________________
        add $v0, $0, $0
        beq $a0, $0, done
        lw $s0, 0($a0)
        add $s1, $a0, $0
        lw $a0, 4($a0)
        jal sum_tree
        add $s0, $s0, $v0
        lw $a0, 8($s1)
        jal sum_tree
        add $s0, $s0, $v0
done:
        ______________________
        ______________________
        ______________________
        ______________________
```

What's this code up to, anyway?

QUESTION THREE:
Assuming the first instruction of each snippet is located at memory address 0x10001000, fill in this table.

MIPS Snippet	***Instruction as binary***
`addu $t0, $t0, $t0`	
`Self: j Self`	
	0b 01000 00010 00000 10000 00000 000000

`bne $t0, $t1, Skip`
`addiu $v0, $zero, 1`
`Skip:`	

`li $t0, 70000`	
	0b 00000 11111 00000 00000 00000 001000