Array Access and Logic Instructions

Question 1:   (5 points)

(V03, V04) Assume that register `a0` contains the binary representation of an `lbu` instruction. Write the minimum sequence of RISC-V instructions that produces a value in `a0` such that only the bit corresponding to the number of the register `rd` of the `lbu` instruction is 1. All other bits of `a0` must be zero. An instruction `lbu rd, offset(rs1)` uses the I-format:

```
15    # lumiptr:
16    # parameters:
17    #       a0: screen address
18    #       a1: R (number of rows)
19    #       a2: C (number of columns)
20    lumiptr:
21        mul   t1, a1, a2     # t1 <- R*C
22        add   t1, a0, t1     # t1 <- screen + R*C
23        add   t6, x0, x0     # t6(luminosity) <- 0
24    next_p:
25        lbu   t2, 0(a0)      # t2 <- pixel
26        add   t6, t6, t2     # lumens <- lumens + pixel
27        addi  a0, a0, 1      # p++
28        bne   a0, t1, next_p
29        addi  a0, t6, 0      # a0 <- lumens
30        jalr  zero, ra, 0
```

Figure 1: The code for `lumiptr`.

Question 2:   (5 points)

Assume that the register `s1` contains the memory address of the first instruction of the function `lumiptr` shown above (RISC-V programs are stored in memory in their binary representation). Write the shortest sequence of RISC-V instructions that loads the binary representation of the instruction `bne` that appears in line 28 into `a0`.

31	20 19	15 14	12 11	7 6	0	
imm[11:0]	rs1	funct3	rd	opcode		I-type
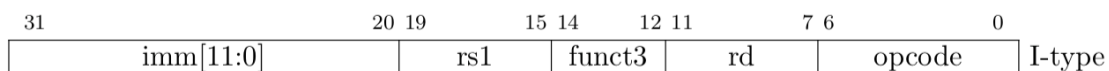
For instance, if the the `rd` register in the instruction whose binary code is in `a0` is register `x3`, then your program has to write the value 0000 0000 0000 0000 0000 0000 0000 1000 in `a0`.

In another example, if the `rd` register is register `x27`, then your program has to write the value 0000 1000 0000 0000 0000 0000 0000 0000 in `a0`.