Question 1: (0 points)
Bank of Questions

Performance Analysis (V01, V09, V0B, V11)

The following questions study the RISC-V assembly code for the `FindMax` procedure shown in Figure **??**. For simplicity, the RISC-V code for storing and restoring callee-saved registers to and from the stack are omitted.

```
1  # FindMax(Square, N, M)
2  # Input Parameters
3  #   a0: Square is the address of first element of 2D matrix
4  #   a1: N is the number of rows in Square
5  #   a2: M is the number of columns in Square
6  # Return Value:
7  #   a0: value of maximum element in Square
8  #
9  0x1FFF FFAC FindMax:    li      s0, -1              # max <- -1
10 0x1FFF FFB0             add     s1, zero, zero      # i <- 0
11 0x1FFF FFB4 NextRow:    slt     t6, s1, a1          # if i<N then t6 <- 1
12 0x1FFF FFB8             beq     t6, zero, Return    # if i>=N Return
13 0x1FFF FFBC             add     s2, zero, zero      # j <- 0
14 0x1FFF FFC0 NextColumn: slt     t6, s2, a2          # if j<M then t6 <- 1
15 0x1FFF FFC4             beq     t6, zero, RowDone   # if j>=M RowDone
16 0x1FFF FFC8             mul     t1, s1, a2          # t1 <- i*M
17 0x1FFF FFCC             add     t2, t1, s2          # t2 <- i*M+j
18 0x1FFF FFD0             slli    t3, t2, 2           # t3 <- 4*(i*M+j)
19 0x1FFF FFD4             add     t4, a0, t3          # t4 <- &(Square[i][j])
20 0x1FFF FFD8             lw      t5, 0(t4)           # t5 <- Square[i][j]
21 0x1FFF FFDC             slt     t6, s0, t5          # if(max < Square[i][j]) then t6 <- 1
22 0x1FFF FFE0             beq     t6, zero, NoChange
23 0x1FFF FFE4             add     s0, t5, zero        # max <- Square[i][j]
24 0x1FFF FFE8 NoChange:   addi    s2, s2, 1           # j <- j+1
25 0x1FFF FFEC             jal     zero, NextColumn
26 0x1FFF FFF0 RowDone:    addi    s1, s1, 1           # i <- i+1
27 0x1FFF FFF4             jal     zero, NextRow
28 0x1FFF FFF8 Return:     add     a0, s0, zero        # a0 <- max
29 0x1FFF FFFC             jalr    zero, ra, 0
```

Figure 1: RISC-V Assembly code for `FindMax` procedure.

Question 2: (4 points)
Consider the following invocation of the procedure `FindMax`

```
lui     a0, 0x002
li      a1, 0x1F4
li      a2, 0x3E8
call    FindMax
```

What are the values, expressed in decimal, of the parameters `N` and `M` for this call to `FindMax`?

Question 3: (4 points)
In a given invocation of `FindMax`, `N = 10000` and `M = 5000` and the condition for the

branch in line 22 is true 50% of the time. How many instructions are executed by this call?

Question 4: (4 points)

Several executions of programs that are similar to `FindMax` have been used to determine the number of clock cycles executed by each type of instructions in the RISC-V processor that is executing `FindMax`. It was determined that the following instructions take one cycle each: `li`, `slt`, `add`, `slli`, `addi`. The `mul` instruction takes five cycles. Branch instructions take four cycles each, the jump instructions `jal` and `jalr` take two cycles each, and a load-word instruction takes ten cycles. How many clock cycles are necessary to execute an invocation of `FindMax` with `N = 10000` and `M = 5000` described above?

Question 5: (4 points)

What is the average number of clocks per instruction (CPI) for the invocation of `FindMax` with `N = 10000` and `M = 5000` described above?

Question 6: (4 points)

If the invocation of `FindMax` with `N = 10000` and `M = 5000` described above is executing in a RISC-V processor running with a clock frequency of 4 GHz, how long does it take to execute `FindMax`?