

CS286 Computer Architecture & Organization

Fall 2026

Suggested Exercise Questions for the Midterm Exam

NOTE: This question list is NOT the list of the questions that are guaranteed to appear in the midterm exam. This question list is not a complete list of the questions (i.e., other questions can appear in the midterm exam).

QUESTION #1

Suppose that you wrote the following statements in C/C++;

```
int my_int_01;  
unsigned int my_int_02;  
  
void main (void)  
{  
  
    my_int_02 = 63489;  
  
    my_int_01 = my_int_02;  
  
    printf("my integer: %d\n", my_int_01);  
}
```

Assuming that the computers that run this program is a 16-bit architecture computer (i.e., the registers in the processors are all 16 bits), what number will you see when you print the value of "my_int_01" in the program? Show all your work to find the number (i.e., please explain how you got the number). Note that "%d" in the *printf* statement means "print it as a decimal number".

QUESTION #2

For 16-bit architecture computers, what is the largest positive integer the processors can handle? What is the smallest negative integer the processors can handle? Show each of the numbers (the largest positive integer and the smallest negative integer) in both the decimal and the binary formats.

QUESTION #3

```
.data
STR_FIRSTNUM: .asciiz "Please enter the first number: "
STR_SECONDNUM: .asciiz "Please enter the second number: "
STR_EQUAL: .asciiz "The two numbers are equal."
STR_NOT_EQUAL: .asciiz "The two numbers are not equal."

.text
.globl main
main:
FIRSTNUM:  li $v0, 4                # system call for print string
            la $a0, STR_FIRSTNUM    # specify string
            syscall                 # display message

            li $v0, 5                # system call for read_int
            syscall                 # get int from user
            move $t0, $v0           # move user int into $t0

            li $t1, 257              # load 257 into t1
            slt $t3, $t0, $t1        # check if t0 < 256
            beq $t2, $zero, FIRSTNUM # if t2 = 0, ask to reenter number

            slt $t2, $t0, $zero      # check if t0 < 0
            bne $t2, $zero, FIRSTNUM # if t2 = 0, ask to reenter number

            move $s0, $t0            # move int from t1 to s0

SECONDNUM:  li $v0, 4                # system call for print string
            la $a0, STR_SECONDNUM    # specify string
            syscall                 # display message

            li $v0, 5                # system call for read_int
            syscall                 # get int from user
            move $t0, $v0           # move user int into $t0

            slt $t2, $t0, $t1        # check if t0 < 256
            bne $t2, $zero, SECONDNUM # if t2 = 0, ask to reenter number

            slt $t2, $t0, $zero      # check if t0 < 0
            bne $t2, $zero, SECONDNUM # if t2 = 0, ask to reenter number

            move $s1, $t0            # move int from t1 to s0

            bne $s0, $s1, ELSE        # if s0 = s1, continue, or goto ELSE
            li $v0, 4                # system call for print string
            la $a0, STR_EQUAL        # specify string
            syscall                 # display string
            j EXIT

ELSE:       li $v0, 4                # system call for print string
            la $a0, STR_EQUAL        # specify string
            syscall                 # display string
            j EXIT

EXIT:       jr $31                  # return to main
```

The program above requests a user to enter two numbers, one at a time. Each number is prompted a user with a message. After that, the program tests if the entered number is within a range 0 – 256 (inclusive), for one at a time. Then, the program compares the two numbers and displays the result. If the two numbers are equal, it displays so. Otherwise it displays a message that indicates the two numbers are not equal. At the end, the program should properly terminate.

Question: There are three things that are not correct. What are they? Mark the three problems in the program above.

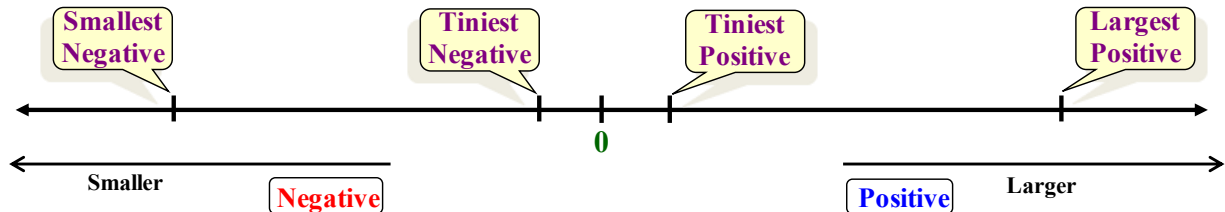
Hints: The problems are neither syntax error nor illegal instructions.

Note: “slt” means ‘set if less than’. “slt \$t0, \$s3, \$s4 # \$t0 = 1 if \$s3 < \$s4 otherwise \$t0 = 0.

QUESTION #4

- What is the difference between the 1264th tiniest and the 1265th tiniest positive numbers (we called the difference “interval”) in IEEE-754 floating point number system (i.e., “single precision IEEE-754 floating point numbers)?
- How many other positive numbers in IEEE-754 floating point number(s) have the same interval to the next tiniest number? (c) Briefly justify your solution.

Note: the grading weight for this question: 40% to (a) and 60% to (b).



QUESTION #5

IEEE-754 floating-point standard has another sub-standard, called “double precision floating-point numbers”, which has 11 bits for exponent and 54 bits for significand. The basic idea is not changed at all, except that the two fields (exponent and significand) are extended and that “bias of 1023” is used for the exponent bits (0: 01111111111).

Show the bit pattern for -823.4375 (negative 823.4375) using the double-precision floating-point number in IEEE-754.
Show all your work for full credit.

QUESTION #6

- (a) Derive a formula for how much faster a super-pipeline processor is, compared to its scalar counterpart (for executing infinitely large number of instructions) when the super-pipeline processors does not experience any super-scalar or pipeline hazard. Assume that the scalar processor has K internal stages, running at the same clock cycle rate of R Hz and that the super-pipeline processor has S parallel data-paths. Show your work.
- (b) Find how much a super-pipeline processor is faster than its scalar counterpart (for the above).

QUESTION #7

We have a processor, called Processor A. The processor clock cycle rate and CPI of processor A are:

- Processor A: Clock cycle rate = 900 MHz ($1M = 10^6$)
- CPI = 1.8

Now a new processor is being developed, called Processor B. The processor clock cycle rate and CPI of processor B are:

- Processor B: Clock cycle rate = 1.2GHz ($1G = 10^9$)
- CPI = 0.85

How can Processor A perform as well as Processor B? Show all your work for full credit

QUESTION #8

Our favorite program runs in 10 seconds on computer A, which has a 400 MHz clock (= clock cycle rate = 400MHz). We are trying to help a computer designer build a computer B, which will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing computer B to require 1.35 times as many clock cycles as computer A for this program. What clock rate should we tell the designer target? Show your work.