(CS2.201a) Computer Systems Organization (Spring 2023) 01.03.2023

Mid-Semester Examination

Alloted time: 90 minutes Total marks: 40

Instructions:

- There are two questions printed on two sides of a sheet.
- Discussions amongst the students are not allowed. No other material either in printed, written, or electronic format can be accessed. Any dishonesty shall be penalized heavily.
- Partial marking is available for every question. Make sure that your answers are legible and most importantly disambiguous.
- Be clear in your arguments. Vague arguments shall not be given full credit.

1. Floating point encoding: Consider the following 5-bit floating point representation based on the IEEE floating point format. This format does not have a sign bit – it can only represent nonnegative numbers.

   - There are $k = 3$ exponent bits. The exponent bias is 3.
   - There are $n = 2$ fraction bits.

   Recall that numeric values are encoded as a value of the form $V = M \times 2^E$, where E is the exponent after biasing, and M is the significand value. The fraction bits encode the significand value M using either a denormalized (exponent field 0) or a normalized representation (exponent field nonzero). The exponent E is given by $E = 1 - \text{Bias}$ for denormalized values and $E = e - \text{Bias}$ for normalized values, where e is the value of the exponent field exp interpreted as an unsigned number.

   Below, you are given some decimal values, and your task it to encode them in floating point format. In addition, you should give the rounded value of the encoded floating point number. To get credit, you must give these as whole numbers (e.g., 17) or as fractions in reduced form (e.g., 3/4). Any rounding of the significand is based on round-to-even, which rounds an unrepresentable value that lies halfway between two representable values to the nearest even representable value.

Value	Floating Point Bits	Rounded Value
9/32	001 00	1/4
3	10010	3
9	11000	8
3/16	00011	3/16
15/2		~~15~~ 8

Please fill the missing entries in the table. [16 marks]

2. Examine the following block of code and answer the questions that follow, based on it.

```
.SomeBlock:
    push ebp
    mov ebp, esp

    cmp dword [ebp + 12], 0 ; Compare the lower 32 bits of data at location ebp + 12, with zero
    je .BaseCase

.recurse:
    mov eax, [ebp + 8]
    xor edx, edx
    div dword [ebp + 12] ; 32 bit version of division
    push edx
    mov eax, [ebp + 12]
    push eax
    call .SomeBlock
    leave  ; leave instruction combines two instructions: (1) mov esp, ebp, and (2) pop ebp
    ret

.BaseCase:
    mov eax, [ebp + 8]

.done:
    leave
    ret
```

(a) Please write down the explanation corresponding to each line of code, and explain what mathematical computation this block of code does.

(b) What changes would one need to make, to make it work for

i. 64 bit machine, and

ii. 64 bit machine and 64 bit data?

[24 marks]

END OF QUESTIONS!