CS 5541 - Computer Systems

Floats and Puzzles

Problem 1:

Assume we are running code on a 6-bit machine using two's complement arithmetic for signed integers. A "short" integer is encoded using 3 bits. Fill in the empty boxes in the table below. The following definitions are used in the table:

```
short sy = -3;
int y = sy;
int x = -17;
unsigned ux = x;
```

Note: You need not fill in entries marked with "–".

Expression	Decimal Representation	Binary Representation
Zero	0	00 000
–	−6	11 1010
–	18	01 0010
ux	47	10 1111
y	-3	11 1101
$x >> 1$	-9	11 0111
TMax	31	01 1111
−TMin	-32	10 0000
TMin + TMin	0	00 0000

Problem 2:

Consider the following 8-bit floating point representation based on the IEEE floating point format:

- There is a sign bit in the most significant bit.
- The next 3 bits are the exponent. The exponent bias is $2^{3-1} - 1 = 3$.
- The last 4 bits are the fraction.
- The representation encodes numbers of the form: $V = (-1)^s \times M \times 2^E$, where M is the significand and E is the biased exponent.

The rules are like those in the IEEE standard(normalized, denormalized, representation of 0, infinity, and NAN). FILL in the table below. Here are the instructions for each field:

- **Binary:** The 8 bit binary representation.
- **M:** The value of the significand. This should be a number of the form x or $\frac{x}{y}$, where x is an integer, and y is an integral power of 2. Examples include 0, $\frac{3}{4}$.
- **E:** The integer value of the exponent.
- **Value:**The numeric value represented.

  Note: you need not fill in entries marked with "—".

Description	Binary	M	E	Value
Minus zero	1 000 0000	0	-2	-0.0
—	0 100 0101	21/16	1	21/8
Smallest denormalized (negative)	1 000 1111	15/16	-2	-15/64
Largest normalized (positive)	0 1101 1111	31/16	3	31/2
One	0 011 0000	1	0	1.0
—	0 101 0110	11/8	2	5.5
Positive infinity	0 111 0000	—	—	$+\infty$

Problem 3:

Consider the following 5-bit floating point representation based on the IEEE floating point format:

- There is a sign bit in the most significant bit.
- The next two bits are the exponent. The exponent bias is 1.
- The last two bits are the significand.

The rules are like those in the IEEE standard (normalized, denormalized, representation of 0, ∞, and NAN). As described in Class 10, the floating point format encodes numbers in a form:

$$(-1)^s \times m \times 2^E$$

where m is the *mantissa* and E is the exponent. The table below enumerates the entire non-negative range for this 5-bit floating point representation. Fill in the blank table entries using the following directions:

E**:** The integer value of the exponent.

m**:** The fractional value of the mantissa. **Your answer must be expressed as a fraction of the form** $x/4$**.**

Value: The numeric value represented. **Your answer must be expressed as a fraction of the form** $x/4$.

You need not fill in entries marked "—".

Bits	E	m	Value
0 00 00	—	—	0
0 00 01	0	1/4	1/4
0 00 10	0	2/4	2/4
0 00 11	0	3/4	3/4
0 01 00	0	4/4	4/4
0 01 01	0	5/4	5/4
0 01 10	0	6/4	6/4
0 01 11	0	7/4	7/4
0 10 00	1	4/4	8/4
0 10 01	1	5/4	10/4
0 10 10	1	6/4	12/4
0 10 11	1	7/4	14/4