CSE 351 Summer 2025, Midterm Exam I
July 10, 2025

Full Name:	
UW NetID:	@uw.edu
Seat Number:	
I certify that all work is my own. I had no prior knowledge of exam contents nor will I share the contents with any student in CSE 351 who has not yet taken the exam. Violation of these terms may result in a failing grade. **Signature:**	

Instructions:

- **Do not turn the page until the time shown at the front of the room.**
- This exam is closed book (no smartphones, calculators, or other electronic devices). Turn off any mobile devices, remove hats, headphones, and smartwatches, and put them away.
- You are allowed one page (U.S. letter, double-sided) of *handwritten* notes. **Write your name and NetID on your notes page and turn it in with your exam.**
- This exam contains 4 problems and 1 bonus question spread across 6 pages. The last page is a reference sheet. You may detach it from the rest of the exam.
- When a box or line is provided, please write your answer in the box or on the line.
- If a question involves bubbling in a ◯, please fill the shape in completely.
- You have 60 minutes to complete this exam. **Please stop writing when the clock stops.**

Advice:

- Read questions carefully before starting. Make sure you understand what they're asking!
- Don't spend too much time on any one problem. If you find yourself getting stuck, skip around. Make sure you get to all the questions.
- **Relax! You are here to learn :)**

Question	**1**	**2**	**3**	**4**	**5**	**Total**
Points	13	10	8	20	1	**52**
Page	2	3	4	5	6	

Name: _____ UW NetID: _____

1. (13 points) Integers and Binary

Suppose **c** is a C **char** (8-bit Two's Complement) with the bits **0b1100 1001**.

(a) (2 points) Give the 8-bit Two's Complement value of the bits held in **c**. Write your answer in decimal.

(b) (2 points) Give the unsigned value of the bits held in **c** (i.e., the result of casting **c** to an **unsigned char**). Write your answer in decimal.

(c) (2 points) How would you represent the 8-bit Two's Complement value of **c** (i.e., the decimal value from (a) above) in 32-bit Two's Complement? Write your answer in hex, including the prefix.

(d) (2 points) Assuming a Two's Complement encoding, what is the minimum number of bits needed to represent the 8-bit Two's Complement value of **c** (i.e., the decimal value from (a) above)?

(e) (5 points) Let **x** be a C **int** with bytes 0xFFFE 1001. Assume **ty** is a C **short** and **uy** is a C **unsigned short**, both with unknown values. For each of the following C expressions, indicate whether the result is always, sometimes, or never true:

(i) `x < ty` ◯ Always ◯ Sometimes ◯ Never

(ii) `(x + 4*ty) < 0` ◯ Always ◯ Sometimes ◯ Never

(iii) `x < (unsigned int)uy` ◯ Always ◯ Sometimes ◯ Never

(iv) `(int)(~x | uy) < 0` ◯ Always ◯ Sometimes ◯ Never

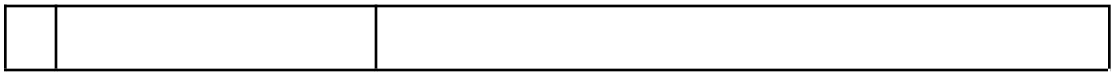
(v) `(int)(x ^ (uy << 8)) < 0`

◯ Always ◯ Sometimes ◯ Never

Name: _____ UW NetID: _____

2. (10 points) Floating Point

(a) (2 points) Give the single-precision floating point representation of the most negative C **`int`**. Write the representation in binary (no prefixes required). You may abbreviate a large number of 1's or 0's by writing the bit times the number of occurrences, e.g., **`0(x5)`** to indicate five 0's in a row.



S **E** **M**

The company you work for is developing a new floating-point representation, even smaller than the single-precision **`float`**. Their requirements are as follows:

- The new representation should take up as few bits as possible;
- It should be able to represent numbers whose magnitude is at least as large as the largest numbers representable as an **`int`** (32-bit Two's Complement); and
- It should be at most 1/16 less precise than a single-precision **`float`**. That is, the minimum quantity of numbers we can represent (per exponent) should be 1/16 less than that of a **`float`**.

(b) (4 points) What is the minimum number of bits required for **E** and **M** in this new floating-point representation?

E: _____ bits **M**: _____ bits

(c) (4 points) Assuming you use the minimum number of bits for **E** and **M**, are there any values an **`int`** can represent that this new, smaller floating-point representation cannot? If so, give an example. Explain your answer in 1-2 sentences.

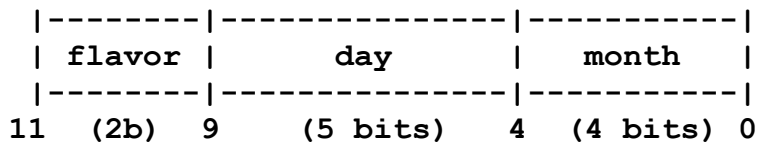
◯ Yes ◯ No

Explain:

Name: _____ UW NetID: _____

3. (8 points) Number Representation & Bitwise Operations

You've been asked to design a numerical representation of ice cream batches for an ice cream store's database. Each batch of ice cream has a flavor (one of four) and a sell-by date, consisting of a month and day. Your coworker proposes the following encoding:



Where **`day`** is a 5-bit number in the inclusive range [1, 31], and **`month`** is a 4-bit number in the inclusive range [1, 12].

For the following questions, assume **`x`** and **`y`** are C **`short`**s containing an ice cream batch encoding in their least significant 11 bits. Make no assumptions about the contents of **`x`** and **`y`**'s most significant 5 bits. Write all constants in hex, including the prefix.

(a) (2 points) Write a C expression to extract the sell-by date from **`x`** (the least significant bits of the result should contain the sell-by day and month, with all other bits as 0).

(b) (2 points) Write a C expression that evaluates to true if **`x`** and **`y`** have the same flavor, and false otherwise.

(c) (4 points) You propose an alternative date encoding: Instead of encoding the day and month separately, you will instead store a single unsigned value representing the number of days since January 1, in the range [0, 366]. For example, a sell-by date of January 1 would have the value 0; January 2 would be 1; and so on. The flavor encoding remains unchanged.

Give one advantage your encoding has over your coworker's proposal. Explain your answer in 1 - 2 sentences.

Name: _____ UW NetID: _____

4. (20 points) Pointers & Memory

For the following questions, assume a 64-bit, little-endian machine. Suppose the state of memory begins as follows (all values in hex):

Addr:

0x100	F8	F9	FA	FB	FC	FD	FE	FF
0x108	01	02	03	04	05	06	07	08
0x110	09	0A	0B	0C	0D	0E	0F	10
0x118	11	12	13	14	15	16	17	18

We then define the following variables:

```
char c[10]; // Assume c begins at 0x108
short* s = (short*)c;
```

(a) (2 points) How much space, in bytes, is allocated by the variable declarations above?

◯ 10 bytes ◯ 12 bytes ◯ 18 bytes ◯ 26 bytes

(b) (12 points) For each of the following C expressions, give its type and value. Write values in hex, including the prefix, and be sure to use the correct bitwidth.

(i) `s[1]`

Type: ◯ `char` ◯ `char*` ◯ `short` ◯ `short*`
Value:

(ii) `&c[5]`

Type: ◯ `char` ◯ `char*` ◯ `short` ◯ `short*`
Value:

(Question continues on next page.)

(b) (continued)

(iii) `*((int*)c + 1)`

Type: ◯ `char` ◯ `char*` ◯ `int` ◯ `int*`
Value:

(iv) `(long)*(c - 1)`

Type: ◯ `char` ◯ `char*` ◯ `long` ◯ `long*`
Value:

(c) (6 points) Fill in the table below with the new contents of memory after we run the following C code (some values have been provided for you):

```
for (int i = 0; i < 4; i++) {
    s[i] = 0xA0 + i;
}
```

Addr:

0x100	F8	F9	FA	FB	FC	FD	FE	FF
0x108								
0x110								
0x118	11	12	13	14	15	16	17	18

5. (1 point) What should Alexandra's orange tabby cat be named? Be creative!