CS 220 Spring 22 Exam 3 **Name:**_____

For partial credit make sure to **show all work** where appropriate. Some answers are better than other answers. Full credit for the best answer.

1. [5] What floating-point number is represented by **`0xC2FEA000`**. You must show work to receive credit.

2. [5] Assume we are multiplying the unsigned integers **1100 X 1011**. Trace the values of the multiplicand, multiplier, and result at every step.

Multiplicand	**Multiplier**	**Result**

3. [2] When multiplying two **n**-bit numbers the result can have as many as _____ bits.

4. [2] What is the purpose of the **`-g`** flag on gcc.

5. [2] What is the purpose of the **`-c`** flag on gcc.

6. [2] In gdb, a location where code execution will be temporarily halted is called a …

7. Consider a logic function with three inputs **A**, **B**, **C** and one output **Out**. The output should be a one when exactly one (and only one) of the inputs is a one.
    a. [5] Draw the truth table for the function.

    b. [5] Write out the sum-of-products logic equation for the function.

    c. [5] Minimize the equation as much as possible

    d. [5] Draw the circuit diagram for the logic equation.

8. Consider the following (rather ridiculous) C function.

```
void f(int vec[], int n) {
    int x = 99;
    int *y = malloc(sizeof(int));
    static double pi = 3.14159;
    printf("%d %X %d %f\n", x, y, vec[3], pi + n);
}
```

   a. [2] Memory for **x** is allocated on/in the _____

   b. [2] Memory for **y** is allocated on/in the _____

   c. [2] Memory for what **y** points to is allocated on/in the _____

   d. [2] How many bytes did the call to **malloc** allocate? _____

   e. [2] Memory for **pi** is allocated on/in _____

   f. [5] To avoid creating a memory leak, can the function that called **f** free the memory that was allocated by **malloc**? Briefly explain why or why not.

9. [5] Assume **x** is an integer variable, write a C statement that will set the 8th bit in **x** to a 1 (where the 0th bit is the least significant bit).

10. Assume a direct mapped cache with four rows and two instructions per row. The function below computes $\mathbf{x^y}$ by multiplying **x** by itself **y** times.

```
104ac:      e3a02001        mov     r2, #1
104b0:      e3510000        cmp     r1, #0
104b4:      da000002        ble     104c4
104b8:      e0020092        mul     r2, r2, r0
104bc:      e2411001        sub     r1, r1, #1
104c0:      eafffffa        b       104b0
104c4:      e1a00002        mov     r0, r2
104c8:      e12fff1e        bx      lr
```

a. [5] What address ranges constitute the loop body?

b. [5] Which row and column in the cache does the **bx** instruction map to? Express answer in decimal.

c. [5] When computing 2^{10} what would the cache hit ratio be for the code?

d. [2] Does the code contain any ***compulsory*** misses? Briefly explain why or why not.

e. [2] Does the code contain any ***conflict*** misses? Briefly Explain why or why not.

f. [2] Does the code contain any ***capacity*** misses? Explain why or why not.

11. Answer questions about the recursive function below. **`u_int32_t`** is just an **`unsigned int`**.

```
u_int32_t what(u_int32_t n, u_int32_t r) {
    if (n == 0)
        return r;
    else
        return what(n >> 1, (r << 1) | (n & 1));
}
```

a. [10] Neatly draw the runtime stack of a call to **`what(13, 0)`** up to the point where we hit the base case. Show the values of **`n`** and **`r`** on each call.	**b.** [2] Briefly describe what the function computes. That is, what is the net effect of the function.