Fall ’18 CIS 314 Midterm Review

You may bring one page of notes, front and back.
Questions will be in short-answer format with partial credit for partial answers.
Questions will require you to read and write C and x86-64 code.

Topics:

- Number formats: decimal, hex, binary
- Integer representations: unsigned, 2s compliment – to/from decimal
- Bitwise operators in C: and, or, not, xor, logical/arithmetic shifts
- Logical operators in C: and, or, not
- IEEE 754 single-precision normalized floating-point representation: to/from decimal
- Data-transfer instructions: movq, pushq, popq
- Arithmetic instructions: addq, shlq, shrq, sarq, andq, orq, xorq, leaq
- Control instructions: cmpq, j* instructions
- Labels, with j* instruction in x86-64, goto statement in C
- Procedures: %rip (program counter), %rsp, return address, callq, retq
- Register conventions: %rdi, %rsi, %rax only
- Arrays: storage in memory, lookup calculation (<= 2 dimensions)

Sample questions:
1. [5] What is the decimal value of the byte 0xA1, interpreted as an 8-bit (signed) int? Show your work:

2. [15] What is the bit value of 3.25 encoded as an IEEE 754 single-precision floating-point number? Show your work:

3. [10] Write x86-64 code which uses a single *leaq* instruction to calculate the result of *x = 4*x + y + 12* assuming that *x* is in %rdi and *y* is in %rsi. Comment your code:

4. [25] Consider the following C code:

```
long sum(long start) {
    long result = 0;
    do {
        result += start;
        --start;
    } while (start);
    return result;
}
```

a. (10) Rewrite the code (in C) to use a label and goto statement rather than a loop. Comment your code:

b. (15) Convert your C code from part *a* above into x86-64. Use the appropriate registers for the arguments and return value (as specified by the x86-64 register conventions). Comment your code: