CIS450 Computer Architecture
Lab 2 quiz
Dr. Daniel Andresen

Notes: This quiz is closed book, closed notes, closed neighbor. Bring a blank sheet of paper to write down your answers. I will select several (but not all) of the questions below for you to answer in class.

1. **Briefly** define the following and discuss their major tradeoffs (advantages/costs). What problem are they addressing?
    a. Word
    b. Hexadecimal
    c. Big-endian/little-endian
    d. C Pointer
    e. Sign extension
2. Discuss why we use binary rather than decimal number representations in hardware.
3. Why are ‘words’ always an integral number of bytes?
4. Discuss the relative merits of C (null-terminated) vs. Pascal (leading length count) style string representations.
5. In C, if you mix unsigned and signed variables in a single expression, signed values are implicitly cast to unsigned. Give an example where this might lead to an unexpected result.
6. Since standard numeric representations such as ‘int’ have implicit limits on their range, why not use standard libraries which can handle effectively infinite precision through representing numbers as strings or linked lists?
7. How would “p && *p” avoid a null pointer access?
8. How would going from 32 bits to 64 bits as the word size affect program performance, assuming cache sizes remained constant?
9. Why do we use typically “nearest even” of the four rounding methods?
10. Why would casting an ‘int’ to a ‘double’ in C change the numeric value?
11. Why do we normalize floating point values as they get further away from 0, but use denormalized values “near” 0?
12. Why are unsigned ints more natural for C programmers to use as pointers rather than signed ints?