

CSCI 2330 – Bitwise Operator Exercises

1. Compute **0x68 | 0x55** and write your answer in hex.
2. Compute **0x68 & 0x55** and write your answer in hex.
3. Compute **0x68 ^ 0x55** and write your answer in hex.
4. Compute **0x68 || 0x55** and write your answer in hex.
5. Assuming 8-bit numbers, compute **~!25** and write your answer in hex.
6. Assuming 8-bit numbers, compute (a) **5 << 1**, (b) **5 << 2**, and (c) **5 << 3** and write your answers in decimal. What is notable about these values?
7. C does not provide a logical XOR operator (which you might reasonably expect to be **^^**). How could you compute the logical XOR of two ints **x** and **y** using existing logical operators (**==**, **!=**, **||**, **&&**, and **!**)? **Hint:** The logical NOT operator (**!**) is a useful way to transform any numeric value into only the values 0 (false) or 1 (true).