CS 64 – MiniExam#1

Data representation
1. Convert the decimal number 27 into an 8-bit number. Express in:
   a) binary:

   b) hex:

2. Convert the following into decimal numbers, assuming it was stored as:
   a) an unsigned 8-bit number:
      1111 1101

   b) a two’s complement 8-bit number:
      1111 1101

Binary Arithmetic – Show all work as in class. Use 8-bit binary numbers.

3. $12 + 8$

4. $2 \times 7$

5. What is the maximum number of bits the answer can be when multiplying two
   13-bit numbers?
**Bitwise arithmetic** - remember that bits are numbered from right to left, and the bit all the way on the right is bit 0. For all of these questions, an 8-bit number is sufficient.

For these questions, I have written a line of code in C. Write a new line of code in C that performs the same operation using a bitwise operation.

6. \( x = y \times 32; \)

7. \( x = y \mod 8; \)

8. \( x = y / 4; \)

For these questions, write **efficient** code that performs the operation described.

9. Set bit 2 to 0 in variable a.

10. Set bit 6 to 1 in variable b.

11. Check to see whether bits 0, 2, and 5 in variable x are the values 1, 1, and 0, respectively. If so, print out "Hooray!".

12. There is an unsigned number stored in the bottom 7 bits of variable v. Assign variable x to be the value stored in variable v.

\[ x = v_\text{--------------------------}; \]