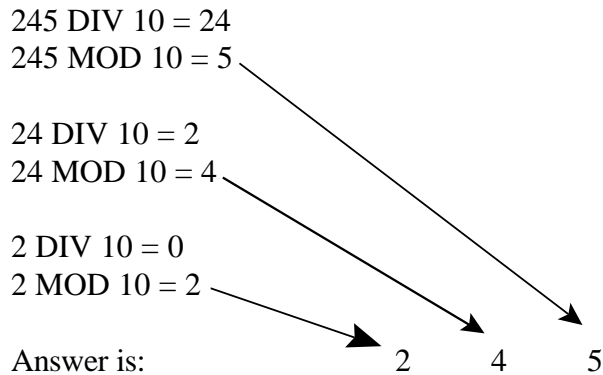


Base Conversion for CS223

When converting between bases for CS223, we're primarily concerned with four bases used for numbers: 2 (binary), 8 (octal), 10 (decimal), and 16 (hex). As we'll show in a minute, converting between bases 2, 8, and 16 is fairly simple and painless. The only real work involves converting to or from decimal. Here's how that's done:

Suppose you have a decimal number, say, 245_{10} . To convert to any other base from decimal (*and only from decimal!*) we use the Mod of Base technique. In this technique, you are determining what the coefficients are for the powers of the base – these coefficients are the digits of the number for that base. Just to show how it works, let's "convert" from decimal to decimal:

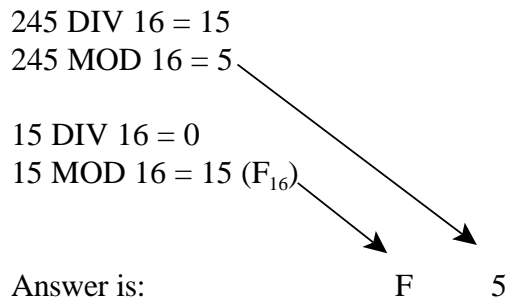
245_{10} :



Answer: 245_{10}

Similarly, convert from decimal to hex:

245_{10} :



Answer: $F5_{16}$

Once you have a number in hex, octal, or binary, converting between the three is a simple matter of arranging bits:

If you have a hex number, convert it to binary by changing each digit into four bits:

$$F5_{16} = 1111\ 0101 \quad (\text{because } F_{16} = 15_{10} = 1111_2 \text{ and } 5_{16} = 5_{10} = 0101_2)$$

So the binary equivalent is 11110101_2 .

If you have an octal number, convert it to binary by changing each digit into **3 bits**:

$$\text{e.g. } 756_8 = 111\ 101\ 110_2 \quad (\text{because } 7_8 = 111_2 \text{ and } 5_8 = 101_2 \text{ and } 6_8 = 110_2)$$

And if you have a binary number, just regroup (starting from the **RIGHT**) into sets of 3 to get octal, or sets of 4 to get hex:

$$11001111000101100_2 =$$

$$11\ 001\ 111\ 000\ 101\ 100 = 317054_8$$

$$1\ 1001\ 1110\ 0010\ 1100 = 19E2C_{16}$$