Worksheet: Adding Binary Numbers

Adding Binary Numbers

1. Convert the denary value 86 to a binary value, writing the bits in the chart below.

bit value (denary)	128	64	32	16	8	4	2	1
denary 86 in binary	0	1	0	1	0	1	1	0

2. Convert the denary value 23 to a binary value, writing the bits in the chart below.

bit value (denary)	128	64	32	16	8	4	2	1
denary 23 in binary	0	0	0	1	0	1	1	1

3. Write the binary equivalent the denary numbers 86 and 23 into the appropriate row, then add the binary digits.

carry over			1		1	1		
denary 86 in binary	0	1	0	1	0	1	1	0
denary 23 in binary	0	0	0	1	0	1	1	1
sum	0	1	1	0	1	1	0	1

4. Write the binary sum from part (3) into row (a), then write the denary sum of bit values in row (b).

	bit value (denary)	128	64	32	16	8	4	2	1
a)	binary sum of 86 and 23	0	1	1	0	1	1	0	1
b)	sum of denary bit values	64 + 32 + 8 + 4 + 1 = 109							

Verify your work from questions 1 through 4. We should get the same result adding in binary as when we add in denary: 86 + 23 = 109. If you have a different answer, review your work to find your mistake.

Overflow

5. Given the binary numbers 147 and 119, add their binary digits.

carry over	1	1	1	1		1	1	1	
denary 147 in binary		1	0	0	1	0	0	1	1
denary 119 in binary		0	1	1	1	0	1	1	1
sum		0	0	0	0	1	0	1	0

You likely noticed a problem. Remember (or if you don't remember, you should try to remember) that 8 bits can store the values 0 through 255 (2^8 = 256 values). However, 147 + 119 = 266. When we sum two 8-bit values, it is not unlikely that the result may require 9 bits to store.

When an operation requires more bits than is allocated to store the resulting value, it is referred to as *overflow*.

There are many cases where overflow is actually used to make an algorithm work.