Activity 1.2

Convert the following denary numbers into binary (using both methods):

a 41	d 100	g 144	j 255	m 4095
b 67	e 111	h 189	k 33000	n 16400
c 86	f 127	i 200	l 888	o 62307

The hexadecimal system

The **hexadecimal number system** is very closely related to the binary system. Hexadecimal (sometimes referred to as simply 'hex') is a base 16 system and therefore needs to use 16 different 'digits' to represent each value.

Because it is a system based on 16 different digits, the numbers 0 to 9 and the letters A to F are used to represent each hexadecimal (hex) digit. A in hex = 10 in denary, B = 11, C = 12, D = 13, E = 14 and F = 15.

Using the same method as for denary and binary, this gives the headings 16^0, 16^1, 16^2, 16^3, and so on. The typical headings for a hexadecimal number with five digits would be:

(16^4)	(16^3)	(16^2)	(16^1)	(16^0)
65 536	**4096**	**256**	**16**	**1**
2	1	F	3	A

A typical example of hex is 2 1 F 3 A.

Since $16 = 2^4$ this means that FOUR binary digits are equivalent to each hexadecimal digit. The following table summarises the link between binary, hexadecimal and denary:

▼ **Table 1.1**

Binary value	Hexadecimal value	Denary value
0 0 0 0	0	0
0 0 0 1	1	1
0 0 1 0	2	2
0 0 1 1	3	3
0 1 0 0	4	4
0 1 0 1	5	5
0 1 1 0	6	6
0 1 1 1	7	7
1 0 0 0	8	8
1 0 0 1	9	9
1 0 1 0	A	10
1 0 1 1	B	11
1 1 0 0	C	12
1 1 0 1	D	13
1 1 1 0	E	14
1 1 1 1	F	15

Converting from binary to hexadecimal and from hexadecimal to binary

Converting from binary to hexadecimal is a fairly easy process. Starting from the right and moving left, split the binary number into groups of 4 bits. If the last group has less than 4 bits, then simply fill in with 0s from the left. Take each group of 4 bits and convert it into the equivalent hexadecimal digit using Table 1.1. Look at the following two examples to see how this works.

Example 1

`101111100001`

First split this up into groups of 4 bits:

`1011 1110 0001`

Then, using Table 1.1, find the equivalent hexadecimal digits:

`B E 1`

Example 2

`100001111111101`

First split this up into groups of 4 bits:

`10 0001 1111 1101`

The left group only contains 2 bits, so add in two 0s:

`0010 0001 1111 1101`

Now use Table 1.1 to find the equivalent hexadecimal digits:

`2 1 F D`