

SRD Work

Year 11 – Computer Science

Number Base revision taken from the Nicola Wilkin Workbook

What are number bases?

Objective: Understand the following number bases: decimal (base 10), binary (base 2) and hexadecimal (base 16).

Decimal (base 10)

Since you first learnt to recognise numbers you have been taught to use a base 10 number system, this is known as a decimal (or denary) number base. It has 10 different digits,

0, 1, 2, 3, 4, 5, 6, 7, 8 and 9

**A decimal number system
uses 10 digits (0 –
9) to represent the value.**

There is no single digit for the number ten and we use two digits (a 1 and a 0) to represent the place value we know as 10. This stands for "1 ten and 0 ones".

To represent any number above 9, we use different values each worth ten times more than the previous column (starting from the right). Take for example the number two thousand, nine hundred and thirty-five (2,935). This can be split into separate columns, each representing a different value.

Thousands	Hundreds	Tens	Ones
2	9	3	5

Each of the columns are worth ten times more than the one to the right. Using the example above the 3 is representing the value 30, the 9 is representing the value 900 and the 2 is representing the value 2000. Adding those values together we get the following:

$$\begin{array}{r}
 2000 \\
 900 \\
 30 \\
 + \quad 5 \\
 \hline
 2935
 \end{array}$$

Task 1: Split the following numbers into their correct columns and then write them in the final column as a decimal number.

Explanation	Thousands	Hundreds	Tens	Ones	Decimal number
One thousand, four hundred and fifty-six					
Nineteen ninety-nine					
Six thousand and twenty					

Binary (base 2)

Computers use electrical pulses and these can either be on or off. As there are only two states a binary number system is used by most computer systems. This on or off state is represented by a 1 for on and 0 for off.

A binary number system uses 2 digits (0 and 1) to represent the value.

Whereas a base 10 number system uses 10 digits 0 – 9, a

base 2 number system uses 2 digits 0 – 1. The base 10 number system has columns which were worth ten times the amount of the previous column and a base 2 number system has columns worth twice as much as the previous column.

Eight	Four	Two	One
1	0	1	1

Using the example above, the first 1 represents 8, there is nothing in the four-place column, 1 in the two column and 1 in the one column. If we add those column values together we get 11.

$$\begin{array}{r}
 8 \\
 2 \\
 + \quad \underline{1} \\
 \hline
 11
 \end{array}$$

Therefore, the binary number 1011 is the same value as 11 in decimal.



Task 2: Place the following numbers into their correct columns and then write them as a binary number by combining them together and finally work out the decimal total by adding together the column headings which contain a 1, for example in the first row the decimal value will be 11 (8 + 2 + 1) to therefore the binary value 1011 is the same as 11 in decimal.

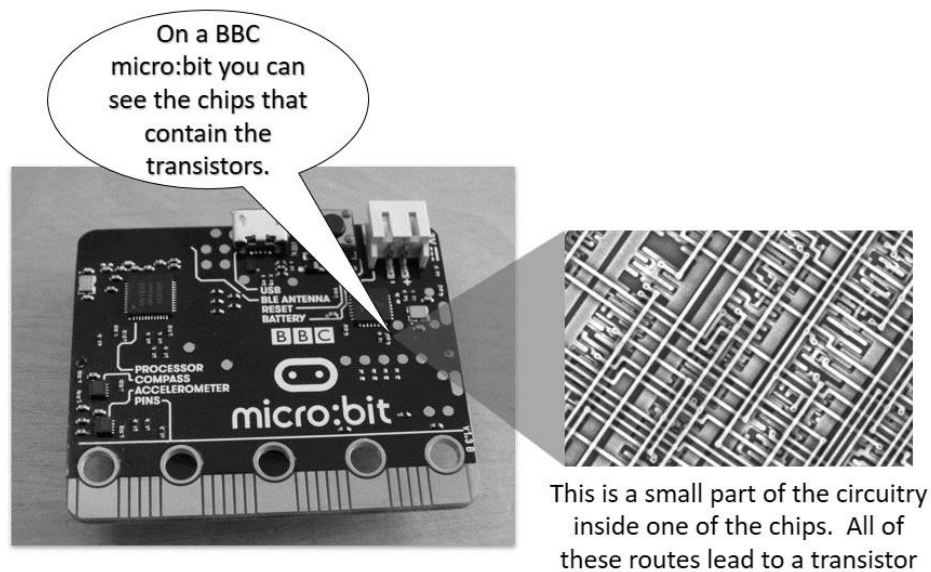
Explanation	Eight	Four	Two	One	Binary number	Decimal value
1 x 8, 0 x 4, 1 x 2 and 1 x 1						
1 x 8, 1 x 4, 0 x 2 and 0 x 1						
1 x 2 and 1 x 1						

What can be represented using binary?

Objective: Understand that computers use binary to represent all data and instructions.

Computers use binary to store data. A **binary digit**, more commonly known as a bit, is the smallest piece of data possible in a computer system, it is either a 1 or a 0.

The circuits in a computer's processor are made up of billions of transistors. A transistor is a microscopic device that open and close circuits to allow electrical signals to either flow or not flow through the circuit. The digits 1 and 0 used in binary reflect the on and off states of a transistor.



To give you an idea of size of these transistors, a human hair is approximately 80,000- 100,000 nanometres wide. In 2016, researchers at Berkeley Lab created a working transistor which is just 1 nanometre long.

Computer programs, as you know them, are known as a “high-level language” and although may be full of technical jargon are more human-friendly than the language used by computers. The high-level languages must be converted into machine code which are binary commands to tell the computer what to do, where to store the data and where they are in the program. Programmers write in a high-level language code and this is converted by a translator into binary instructions that the processor can execute.

All software, programs, images, music, documents, video and any other information that is processed by a computer is stored using these on/off electrical pulses in the computer and can be represented in binary. We will be looking at how this is done later in the book.

Hexadecimal (base 16)

We have learnt that a base 10 number system uses 10 digits and a base 2 number system uses 2 digits, so you will not be surprised that a base 16 number system uses 16 digits. "But wait," I hear you cry. "There are only 10 possible number digits, what do we use for the other 6?" Good question and I'm glad you are paying attention.

A hexadecimal number system uses 16 digits (0 - 9 and A-F) to represent the value.

They could have chosen anything to be the extra symbols for the additional digits, but to make things easier they decided to use letters of the alphabet as most keyboards have these already. In fact, specifically, they use the first 6 uppercase characters of the alphabet so the digits available for a hexadecimal system are:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F.

The individual digits in a hexadecimal number are worth 16 times the amount of the previous column value.

4096	256	16	1
6	B	2	8

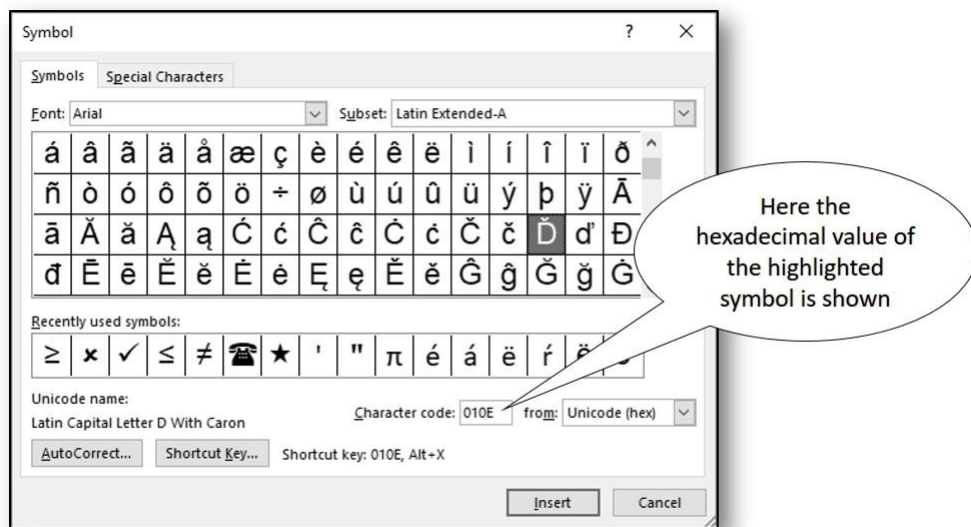
Therefore, the hexadecimal value 6B28 is worth 27,432 in a decimal number system. Luckily for a GCSE in Computer Science, you only need to be able to convert a number up to the decimal value of 255 which is two hexadecimal positions, which makes calculations much easier to work with.

Why is hexadecimal used?

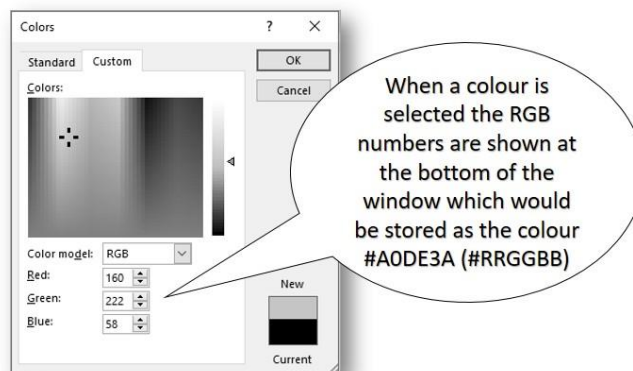
Objective: Explain why hexadecimal is often used in computer science.

The reason people use hexadecimal numbers is as a shorthand notation of the long binary numbers you may have to work with. It shortens them considerably and therefore you are less likely to include typing errors or note down the value incorrectly if there are fewer digits to be working with.

For instance, symbols in word processors often show their hexadecimal number and you can see this in some applications.



They are also found when referring to colours. Colours are usually made up from three numbers known as the RGB code (which stands for red, green, blue) and these are displayed as hexadecimal numbers.



As we have seen, binary is used by computers and one hexadecimal digit can be used to represent 4 binary digits so makes converting between binary and hexadecimal neat and easier than using, say, a base 20 number system.

Here is a table showing the decimal values from 0 to 15 in both binary and hexadecimal.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Here is the same number represented in the three number bases:

Decimal	2,890
Binary	1011 0100 1010
Hexadecimal	B4A

In the next chapter, you will be learning how to convert between these different number bases for yourself but for now you only need to understand the differences between the three number bases.

End of chapter recap

Task 3: Using the table on the previous page, convert these binary numbers into hexadecimal to find the hidden words.	
Binary number	Hidden Word
1011 1110 1110 1111	
1011 1110 1101	
1101 1110 1010 1111	
1010 1101 1101	
1010 1100 1110	
1011 1110 1010 1101	
1111 1110 1110 1101	
1011 1010 1101	
1100 1010 1011	

Converting between number bases

Objective: Be able to convert in both directions between binary and decimal, binary and hexadecimal and decimal and hexadecimal.

Convert from binary to decimal

The easiest to convert is from binary to decimal. We have already seen that each column is worth double the previous column, you just need to remember three things:

- The columns start from the right (the least significant place)
- The first column is worth 1
- Each column (moving to the left) is worth double the previous column.

Here is an example of the column place value:

128	64	32	16	8	4	2	1

As you can see it starts from the right with the value 1 and they double in value as they move to the left.


To work out the decimal value for a binary number you simply insert each individual digit into the columns starting from the right. For instance the binary number 1011 would appear in the columns as follows:

128	64	32	16	8	4	2	1
				1	0	1	1

Now you need to add up the column value for any columns that contain a 1.

$$\begin{array}{r}
 8 \\
 2 \\
 + \quad 1 \\
 \hline
 11
 \end{array}$$

Therefore, the binary value of 1011 is equivalent to 11 in decimal.



Task 4: Convert these binary numbers into decimal. First, insert the digits into the correct columns, starting from the right, and then add up the columns containing a 1 to find the decimal equivalent.

Binary number	128	64	32	16	8	4	2	1	Decimal value
110									
11001									
10101									
100111									
1011000									
1100001									
10101100									
11111111									

More often you will not have a neat table in which to lay out your binary numbers and people tend to simply note down the numbers above each of the binary digits to help them work out the column values.

64 32 16 8 4 2 1
1011010

In the example shown the decimal value is 90 (64 + 16 + 8 + 2).

Task 5: Convert these binary numbers into decimal.	
Binary number	Decimal value
101	
1101	
10111	
110010	
1111100	
1001001	
10101110	
11011001	

Convert from decimal to binary

To convert from decimal to binary you need to perform a little more maths.

Lets familiarise ourselves with the column headings again.

128	64	32	16	8	4	2	1

Step 1: Decide on the column to start with. This should be lower than or equal to the value you are looking for so if we wanted to convert 50 to binary we would start with the column 32. Enter a 1 in that column.

Step 2: Find out the remainder ($50 - 32 = 18$)

Step 3: Repeat steps 1 and 2 until there is no more remainder (in this case we would also put a 1 in the 16 and the 2 columns).

Step 4: Fill in the other columns with 0's. Please note: you do not need to add 0s before your first 1 as these are unnecessary. Using the example of 50 our binary number would be 110010 ($32 + 16 + 2$).

Task 6: Convert decimal values into binary numbers.									
Decimal value	128	64	32	16	8	4	2	1	Binary number
27									
33									
52									
63									
85									
207									

Number base notation

As you may have worked out, the number “10” could mean ten in decimal or it could be two in binary. To make it clear as to which number base is being used it is common to include the number base as a subscript after the number. For instance, 100_2 would be a binary number (in this case it is equivalent to four in decimal) as opposed to 100_{10} which is one hundred in decimal.



Task 7: Convert the following numbers. If the number is currently a binary number convert it to a decimal number and if it is currently a decimal number, convert it to a binary number.

Original number	Converted number
1110_2	
10_{10}	
101_{10}	
11001110_2	
110_{10}	
11000011_2	

Convert from binary to hexadecimal

Each hexadecimal number is equivalent to one of 16 possible digits. (0 – 9 and A – F). Neatly, 4 binary digits (bits) can also be used to represent the values 0 to 15, which is 16 possible digits.

Each hexadecimal number, therefore, can be represented by 4 bits. Look back on task 3 (page 11) where you created words out of binary blocks. Each of those binary blocks contains 4 bits.

Step 1: To convert a binary number into hexadecimal you need to split the binary number into blocks of 4 but this needs to start from the right-hand number (the least significant bit). Therefore, 1101101 would become 110 1101.

Step 2: Convert each of these blocks of 4 into a decimal number. If the number is over 9 then use the following 10 = A, 11 = B, 12 = C, 13 = D, 14 = E and 15 = F. Using our example of 110 1101 the first block would make 6D.

Task 8: Convert the following binary numbers into hexadecimal	
Binary	Hexadecimal
1011 0011 ₂	
1101 0111 ₂	
11 0110 ₂	
11101111 ₂	
1110111 ₂	
11111000 ₂	
10111 ₂	
10111010 ₂	

Convert hexadecimal to binary

To convert back from hexadecimal to binary you simply need to reverse the process.

Step 1: Convert each single character into a decimal number (either the number is the number shown or if it is a letter use A = 10, B = 11, C = 12, D = 13, E = 14 and F = 15).

Step 2: Convert each of those decimal numbers into binary using the technique you learnt on page 15.

Step 3: Combine those number together into a single binary number (remove the spaces)

Task 9: Convert the following hexadecimal numbers into binary.	
Hexadecimal	Binary
82 ₁₆	
C3 ₁₆	
F0 ₁₆	
2B ₁₆	
4A ₁₆	

Convert hexadecimal to decimal

There are several ways to do this, all of which involve complicated maths involving your 16 times table. Not many of us know our 16 times table (unless you are that delightful woman on Countdown of course) so the best advice I can give is to follow these much simpler steps, which, although may be more long winded, are less likely to make your brain hurt and make a mistake in your calculations:

Step 1: Convert the hexadecimal number to binary (see page 18)

Step 2: Convert the binary number to decimal (see page 12) Make sure you are using the whole binary number and working out the column headings as shown below, rather than working with the blocks of 4 individually.

64 32 16 8 4 2 1
1011010

Task 9: Convert the following hexadecimal numbers into decimal by first converting it to binary and then converting the whole binary number to decimal.		
Hexadecimal	Binary	Decimal
$1A_{16}$		
$B8_{16}$		
$F9_{16}$		
$4E_{16}$		
BC_{16}		
52_{16}		

Convert from decimal to hexadecimal

To reverse the process and convert from decimal to hexadecimal you need to do the following:

Step 1: Convert the decimal number to binary (see page 15)

Step 2: Convert from binary to hexadecimal (see page 17).

Task 10: Convert the following decimal numbers into hexadecimal by first converting it to binary and then converting the binary number to hexadecimal.		
Decimal	Binary	Hexadecimal
12_{10}		
49_{10}		
78_{10}		
95_{10}		
116_{10}		
255_{10}		

End of chapter recap

Task 11: Fill in the blanks on this conversion table.		
Decimal	Binary	Hexadecimal
5_{10}		
	10001001_2	
		$5B_{16}$
	10101101_2	
27_{10}		
		$E7_{16}$
	11100011_2	
163_{10}		
	11010001_2	
66_{10}		
		$7C_{16}$
	1011000_2	

Units of information

Objective: Know that a bit is the fundamental unit of information and a byte is a group of 8 bits.

Know that the quantities of bytes can be described using prefixes (kB for Kilobyte, etc.).

Bits

This is either a 0 or a 1. Bit is an abbreviation of the term **B**inary **D**igit (B-it) and is noted with a lowercase b.

A bit is the smallest piece of data stored on a computer

A byte is a group of 8 bits

Bytes

A byte is the name given to a group of 8 bit's (from 00000000 to 11111111). A byte can represent 256 different whole numbers and is noted with an uppercase B.

The table below shows the different units of information you should be aware of:

Name	Abbreviation	Multiple of...
Bit	b	-
Byte	B	8 bits
Kilobytes	kB	1000 bytes
Megabytes	MB	1000 kilobytes
Gigabytes	GB	1000 megabytes
Terabytes	TB	1000 gigabytes
Petabytes	PB	1000 terabytes
Exabyte	EB	1000 petabytes
Zettabyte	ZB	1000 exabytes
Yottabyte	YB	1000 zettabytes

Please note: The terms petabyte, exabyte, zettabyte and yottabyte are not needed for the AQA GCSE Computer Science examination (8520) and have only been included for your information.

Amount of storage space required

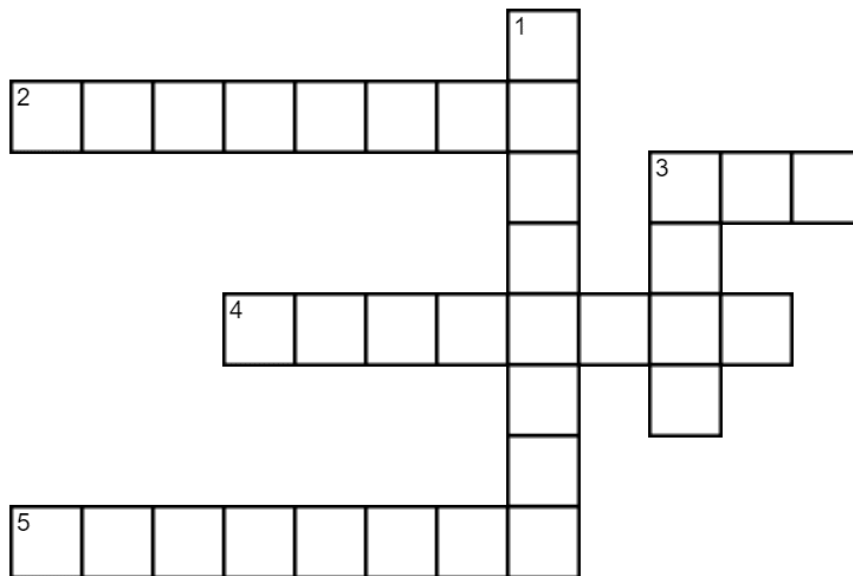
Different types of data require different amounts of storage space. Some examples of this follow:

Data	Approximate file size
A typical line of text from a book	90 bytes
One page of text from a standard novel	4 kB
MP3 song	3.5 MB
A Blu-ray movie	22 GB

End of chapter recap



Task 12: Complete the crossword



Across

- 2) 1000 bytes
- 3) A single 1 or 0
- 4) 1000 gigabytes
- 5) 1000 megabytes

Down

- 1) 1000 kilobytes
- 3) 8 bits

Binary Arithmetic

Objectives: To be able to add up to three binary numbers.

Before we look at adding with binary let's have a quick recap on the first few digits with binary using the first three digits

Binary	Decimal equivalent
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Now let's have a look at some very simple binary maths

Binary arithmetic	Binary Answer	Decimal equivalent
$0 + 0$	0	$0 + 0 = 0$
$0 + 1$	1	$0 + 1 = 1$
$1 + 0$	1	$1 + 0 = 1$
$1 + 1$	10	$1 + 1 = 2$
$1 + 1 + 1$	11	$1 + 1 + 1 = 3$
$1 + 1 + 1 + 1$	100	$1 + 1 + 1 + 1 = 4$

When you are adding two binary numbers together, for instance 1010 and 111, lay them out as follows:

$$\begin{array}{r}
 1\ 0\ 1\ 0 \\
 +\quad 1\ 1\ 1 \\
 \hline
 \hline
 \end{array}$$

Description	Example
<p>Take the first column on the right and add together the individual 1's. In this case $0 + 1 = 1$</p>	$ \begin{array}{r} 1\ 0\ 1\ 0 \\ +\quad 1\ 1\ 1 \\ \hline \qquad\qquad 1 \\ \hline \end{array} $
<p>Next take the next column and add together the 1's. In this case $1 + 1 = 10$. Don't forget we are adding in binary so the answer should also be in binary. Put the 0 in the same column and carry the 1. This is usually shown below the bottom line.</p>	$ \begin{array}{r} 1\ 0\ 1\ 0 \\ +\quad 1\ 1\ 1 \\ \hline \qquad\ 0\ 1 \\ \hline 1 \end{array} $
<p>In the next column don't forget to include the 1 that has been carried forward. In this example the calculation is $0 + 1 + 1$ which is 10. Again, put the 0 in the column and carry the 1.</p>	$ \begin{array}{r} 1\ 0\ 1\ 0 \\ +\quad 1\ 1\ 1 \\ \hline \qquad 0\ 0\ 1 \\ \hline 1\ 1 \end{array} $
<p>In the final column add together the digits, including the carried digit. In this case $1 + 1 = 10$. As there are no other columns instead of putting the 1 that is being carried below the line, move it up to the main answer row.</p>	$ \begin{array}{r} 1\ 0\ 1\ 0 \\ +\quad 1\ 1\ 1 \\ \hline 1\ 0\ 0\ 0\ 1 \\ \hline 1\ 1 \end{array} $

The answer to the sum $1010 + 111 = 10001$. You can check this is correct by converting it to decimal. 1010 (ten) + 111 (seven) = 10001 (seventeen).



Task 13: Complete the following binary addition

$$\begin{array}{r} 1\ 1\ 0\ 0 \\ +\ 0\ 1\ 1\ 0 \\ \hline \end{array}$$

$$\begin{array}{r} 1\ 0\ 0\ 1 \\ +\ 1\ 1\ 1\ 1 \\ \hline \end{array}$$

$$\begin{array}{r} 1\ 0\ 1\ 1\ 1\ 0 \\ +\ \ 1\ 1\ 0\ 1 \\ \hline \end{array}$$

$$\begin{array}{r} 1\ 1\ 0\ 0\ 0 \\ +\ 0\ 1\ 1\ 0\ 1 \\ \hline \end{array}$$

$$\begin{array}{r} 1\ 1\ 0\ 0\ 0\ 1 \\ \ 1\ 0\ 1\ 0 \\ +\ \ 1\ 1\ 1\ 0\ 0 \\ \hline \end{array}$$

$$\begin{array}{r} 1\ 1\ 0\ 0 \\ \ 1\ 1\ 0 \\ +\ 0\ 1\ 1\ 1 \\ \hline \end{array}$$

11110000 + 10011 =

1101011 + 10111
+ 1101001 =

1111 + 111111
+ 11111111 =

Use this space to convert the binary numbers into decimal to check if your answers are correct.

Binary Shifts

Objectives: Be able to apply a binary shift to a binary number. Describe situations where binary shifts can be used.

Lets take an 8-bit binary number, a byte:

0	0	1	0	1	0	0	0
---	---	---	---	---	---	---	---

We could shift those binary digits to the right →

0	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---

or to the left ←

0	1	0	1	0	0	0	0
---	---	---	---	---	---	---	---

Notice that as you move them left or right extra 0's are added to fill in the gaps. All the digits that fall off the end are lost and get deleted. But why would you bother shifting binary bits anyway?

Lets have a look if we keep moving the digits to the right.

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
0	0	0	0	1	0	1	0	10
0	0	0	0	0	1	0	1	5
0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	0	1	1

You will see that every time the number is shifted to the right it divides by 2. The only exception is the second to last row which shows 2 rather than 2.5 as it is only working with whole numbers so the 1 bit is deleted once it falls off the end of the byte.

If we shift the numbers to the left it performs a different calculation:

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
0	0	1	0	1	0	0	0	40
0	1	0	1	0	0	0	0	80
1	0	1	0	0	0	0	0	160

When you shift to the left it doubles the value.
This is because we are working with a base 2
number system and each column is worth
double the previous column.

End of chapter recap

**Sifting the binary digits to the left
doubles the value and sifting to
the right halves the value**





Task 14: Perform a binary shift on the decimal number by converting the number in binary and writing that in the Original binary column. Perform the binary shift and wrote the new binary number in the new binary number column. Finally, convert the new binary number into decimal. Remember is numbers don't fit in the 8-bit grid they get deleted.

Original decimal number	Original binary number	Instruction	New binary number	New decimal number
24		Double it		
34		Half it		
50		Multiply by 4		
60		Divide by 4		
38		Multiply by 8		
76		Divide by 8		
9		Multiply by 16		
168		Divide by 16		

As you can see the binary shift allows you to perform simple multiplication and division by the power of 2, (i.e. multiplying and dividing by 2, 4, 8 ,16, 32, 64 and 128). However, as it is used for 8 bits anything which shifts the numbers off the ends will mean you get a wrong answer. We have looked at something called a binary shift and as you have seen this leads to errors in some occasions when it goes outside of the 8 bits.

For the AQA GCSE Computer Science examination (8520) you will not need to understand negative numbers and do not needs to investigate how to resolve this problem, however you do need to be aware of the problem of shifting digits outside the 8-bits.