of five hundred and six knots in a string or five hundred and six notches on a stick. In binary arithmetic, the same number is represented as a clumsy 111111010.

Because the system used here is 'binary', the place value of each digit in each column is different. Instead of increasing in value in powers of 10, the columns go up in powers of 2.

The column on the right is still the 'ones' column, but because there are only two symbols (0 and 1), we run out of digits as soon as we add 1. In the decimal system, we only run out of symbols when we get to 9; the next column uses a digit which says: we have run out of symbols — we don't have anything for numbers bigger than 9 — so we'll use the 'tens' column and use a 1 to indicate that we now have one 'lot' of ten.

The binary system works in exactly the same way. Instead of grouping in tens and writing 10 for ten, binary groups in twos, so the binary digits 10 represent the decimal number 2.

Showing the number five hundred and six written in decimal and in binary illustrates the essential similarity clearly:

Hundreds	Tens	Ones	
5	0	6	
= 5x100 +	0x10 +	6x1	(= 506)

256s	128s	64s	32s	16s	8s	4s	2s	1s
1	1	1	1	1	1	0	1	0

= 1x256 + 1x128 + 1x64 + 1x32 + 1x16
+ 1x8 + 0x4 + 1x2 + 0x1 (= 506)

The rules of arithmetic in the binary system are exactly the same as the familiar rules of the decimal system — the only difference is that we run out of counting symbols after 1 instead of after 9. Let's try some additions to prove it. Decimal equivalents are printed in brackets.

```
 (3)    11
+(5)  +101
----------
 (8)  1000
```

(1 + 1 = 0 carry 1)
(1(carried) + 1 = 0 carry 1)
(1(carried) + 1 = 0 carry 1)
(1(carried) + 0 = 1)

In binary, as we have just seen, adding 1 to 1 means we have run out of symbols as only zeros and ones are allowed. So we say 'one and one equals nought carry one' (just as in decimal adding 1 to 9 means we have run out of symbols — there are no symbols larger than 9 — so again we say 'nine and one equals nought carry one'). Here's another addition, worked out for you, with two more to try for yourself.

```
  (4)   100      (7)  111    (3)    11
 +(6)  +110     +(2) + 10  +(12) +1100
----------     -----------------------
 (10)  1010      (?)    ?    (?)     ?
```

By now you will have noticed that binary numbers are much longer than their decimal equivalent. See if you can add 11010110 to 1101101 — remember to keep the rightmost columns lined up just as you would when adding a longer decimal number to a shorter one!

The History Of Numbers

Babylonian **Roman** **Hindu** **Binary**

The ancient Babylonians had an advanced number system, based on 60 rather than 10. Their representation of the number 59 in Babylonian 'cuneiform' script is shown above. The use of 60 as a number base had many advantages and there are still traces of their system in use today. There are 60 seconds in a minute, 60 minutes in an hour and six times 60 degrees in a circle — all vestiges of a mathematical system perfected 4000 years ago.

The Roman system was a considerable step backwards. Letters of the alphabet were used to represent numbers, but the position of each Roman numeral gave no indication of its value, making even simple arithmetic almost impossible.

The Hindus used nine signs for the numbers 1 to 9 and later added a sign to represent zero. Their vital contribution was 'place value' — the idea that a digit's position in a number determines how much that digit is 'worth'. Thus the 3 in 30 is 'worth' three tens. The Hindu system was adopted by the Arabs and gradually spread to Europe. One of the leading Arab mathematicians was called Al Khowarizmi. The Latinised pronunciation of his name gave us the mathematical term algorithm and his book 'Al-jabr wa'l Mugabalah' is remembered in the word algebra.

Computers use the binary system because numbers of any size can be represented using only ones and zeros