Data Representation - Part 1

Name	Abbreviation	Multiple of...
Bit	b	-
Nibble	N	4 bits
Byte	B	8 bits
Kilobytes	kB	1000 bytes
Megabytes	MB	1000 kilobytes
Gigabytes	GB	1000 megabytes
Terabytes	TB	1000 gigabytes
Petabytes	PB	1000 terabytes

ALL data must be translated into binary to be processed by a computer.

Adding with binary

Description	Example
Take the first column on the right and add together the individual 1's. In this case 0 + 1 = 1	1 0 1 0 + 1 1 1 1
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.	1 0 1 0 + 1 1 1 0 1 1
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.	1 0 1 0 + 1 1 1 0 0 1 1 1
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.	1 0 1 0 + 1 1 1 1 0 0 0 1 1 1

CHECK DIGITS

When data is transferred across networks it can easily become corrupted by outside interference. This can cause problems if, for example, a credit card number was sent incorrectly causing the wrong person's account to be debited.

For instance, on a UPC bar code the check digit is the last digit shown (in this case a 3). The other numbers are used in the calculation to generate the final check digit.

ASCII, Extended ASCII and Unicode

ASCII uses 7-bits to represent characters allowing 127 characters to be represented. **Extended ASCII** code is an 8-bit character set that represents 256 different characters making it possible to use characters such as ö or é. Extended ASCII is useful for most European languages. **Unicode** contains 136,755 characters covering 139 modern and historic languages, as well as lots of symbols which are used in maths and other specialist areas.

CONVERTING BINARY INTO DENARY

01101001 =

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

64 + 32 + 8 + 1 = 105

CONVERTING DENARY INTO BINARY

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).

BINARY SHIFT

Moving a pattern of binary digits to the left or right will multiply or divide the denary value.

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

Left = Multiply by 2

Right = Divide by 2

CONVERTING BINARY INTO HEXADECIMAL

Denary	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

0100 1110 = 4E