What the Hex this all about?

Hexadecimal, or Base 16, often referred to as Hex is a very useful number system to understand in computing. It allows us to express the two nibbles of a byte using just two characters from 0 to F.

If you're reading this, you should already have watched my PowerPoint presentation, What the Hex this. If not, it's worth watching that first, then coming back to this handout. The presentation tells you what Hex is, and why it's useful to us in computing. This sheet looks at how to convert numbers from decimal to hex, and gives you some exercises.

For values from 0 to 15_{10} the corresponding values in binary and hex as shown in the table below:

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

What are the digits?

Just as, in decimal, we change columns in powers of 10:

Column "name"	**Thousands**	**Hundreds**	**Tens**	**Units**
Multiples of:	10^3	10^2	10^1	10^0
Values:	0, 1000, 2000, … 9000	0, 100, 200, … 900	0, 10, 20, … 90	0-9

So, in hex we ascend in powers of 16:

Column "name"	**FourtousandAndNinetySixes**	**TwoHundredAndFiftySixes**	**Sixteens**	**Units**
Multiples of:	16^3	16^2	16^1	16^0
Values:	0, 1000, 2000, 3000, 4000, … F000	0, 100, 200, 300, 400, … F00	0, 10, 20, 30, … F0	0-F
Decimal equivalent:	0, 4096, 8192, 12288, 16384, …61440	0, 256, 512, 768, 1024, … 3840	0, 16, 32, 48, … 240	0-15

Looks scary? It's not quite so bad – at GCSE we only have to bother ourselves with the first two (rightmost) columns – so long as we have some notional understanding of how quickly hex numbers get a lot bigger in the next column.

What do I need to be able to do?

The important thing for us, at this stage, is to be able to count from 0_{16} to FF_{16} and convert those values to and from decimal – that's 0_{10} to 255_{10} – and 8-bit binary as a byte shown as two nibbles – that's 0000 0000_2 through to 1111 1111_2

For example, we need to be able to work out any of the missing values in the table below, For example, we need to be able to work out any of the missing values in the table below, **without** writing the entire table!

8-bit binary [byte]	2-digit Hexadecimal	Decimal
0000 0000	00	0
0000 0001	01	1
0000 0010	02	2
…	…	
000 1001	09	9
000 1010	0A	10
000 1011	0B	11
…		
000 1110	0E	14
000 1111	0F	15
0001 0000	10	16
0001 0001	11	17
0001 0010	12	18
…	…	
1001 1110	9E	158
1001 1111	9F	159
1010 0000	A0	160
1010 0001	A1	161
…		
1010 1110	AE	174
1010 1111	AF	175
1011 0000	B0	176
1011 0001	B1	177
…		
1111 1100	FC	252
1111 1101	FD	253
1111 1110	FE	254
And finally		
1111 1111	FF	255

How do I convert …

You may already have spotted, converting between Hex and Binary is dead easy, so long as you can count from 0_{10} to 15_{10} and do the same in a binary nibble and from 0_{16} to F_{16} – because for every single digit in hex, there are four corresponding binary digits, or one nibble.

So, once we know:

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

Then if I give you a string of hex something like: FF 00 0B BA 7F FF
Then the corresponding bytes are: 1111 1111   0000 0000   0000 1011   1011 1010   0111 1111   1111 1111

Hopefully, that also lets you see how hex makes binary machine code just a **little** bit easier for a human to read.

Just for fun:
If we don't split it into bytes, but just treat that as a 12-digit hex number, then:

$FF000BBA7FFF_{16}$ = $111111110000000000001011101110100111111111111111_2$ or $280,375,661,854,719_{10}$

(… and no, you won't have to do calculations like that into decimal, even at A Level!)

So what do I need to be able to do with this?

To put it very simply, you need to be able to convert between binary nibbles and hex in both directions in your head (one nibble at a time) without looking at that table. So you should be able to say something like "One-zero-one-one in binary is eleven in decimal, so that'll be B in hex" – or similarly "D in hex is thirteen to us, so that would be one-one-zero-one in binary".

Forgive me while I flip page orientation again …

Exercise

Please try to do these without looking back to the tables on the previous pages – you are very, very likely to get some of this stuff in the exam, and it really helps if you can do it without the table.

Otherwise, if it really helps, then be able scribble to 0_{10} to 15_{10} table with 4-digit bytes and hex values very quickly – and in fairly small writing – there should be somewhere on the exam paper you can scribble that as "workings"!

Task 1

Convert the following hex-code bytes into binary:

00

A8

7F

D2

9B

2D

C5

4E

Task 2

Convert the following bytes into hex:

0110 1011

0011 1101

1010 0101

1101 1011

0111 1110

1000 1100

1110 0100

1111 1010