

FUZE BASIC



Worksheet: **10a**

"Binary Invaders"

A computer understands two things, **On** and **Off**. **Off** is the same as **zero** and **on** is **one**. The most common and by far the most important part in a computer is called a transistor. A transistor is a tiny electronic switch that can be set to **on** or **off** (**one** or **zero**) using electrical charges.

If a computer only understands **ones** and **zeros** how does it send emails, play games, make music, show movies and surf the web?

The Binary number system (or base 2) is based on **ones** and **zeros** so is perfect for computers. Rather than counting in tens, hundreds and thousands it is based on the power of 2. Each binary digit from **right to left** is increased by the power of 2 so the first is 1, the second is 2, and then 4, 8, 16, 32, 64 and 128.

Each binary digit is called a **BIT**. Binary numbers are usually read in blocks of eight **BITS** at a time called a **BYTE**. **BITS** and **BYTES** are the very lifeblood of all computers. From music to pictures, from maths to documents, everything in a computer is stored and processed in **BITS** and **BYTES**.

Take the letter 'A' for example, what exactly is the letter 'A' to a computer? Computers use an index of numbers to define all the letters and characters used in a font. The letter 'A' for example is 65. The number '1' is indexed as 48. Very confusing yes? Well it gets worse!

A computer only understands binary so how do we express 65 as a binary number? **01000001** that's how!

128	64	32	16	8	4	2	1	
0	1	0	0	0	0	0	1	= 65 = A

If all the **BITS** were on, or 1s, and you add them all together you have a decimal value of 255. $1+2+4+8+16+32+64+128 = 255$. There's still one more value though and that's if all the **BITS** are off or zero. This is the same as the decimal value 0. This gives us 256 possible values for an 8 **BIT** number.

Enough already - lets have some fun!

```
CLS
DEFCHAR (1, 0, 66, 36, 126, 90, 255, 189, 129, 102, 0)
PRINTAT (0,0); CHR$(1);
END
```

The humble 'SPACE INVADER' can be expressed in binary, as can just about any shape, sound or text. **RUN F3** the program on the left.

Using the **DEFCHAR** command we can redefine a character to look like a Space Invader. In binary it looks like this;

128	64	32	16	8	4	2	1	
0	0	0	0	0	0	0	0	= 0
0	1	0	0	0	0	1	0	= 66
0	0	1	0	0	1	0	0	= 36
0	1	1	1	1	1	1	0	= 126
0	1	0	1	1	0	1	0	= 90
1	1	1	1	1	1	1	1	= 255
1	0	1	1	1	1	0	1	= 189
1	0	0	0	0	0	0	1	= 129
0	1	1	0	0	1	1	0	= 102
0	0	0	0	0	0	0	0	= 0

Enter the program on the left and **RUN** to see what happens. Cool eh..

Take a Space Invader...

This is how it might look on screen (left) but to the computer it is just a series of switches left in an on or off position (right)



```
CLS
DEFCHAR (1, 0, 66, 36, 126, 90, 255, 189, 129, 102, 0)
DEFCHAR (2, 0, 24, 36, 126, 90, 255, 255, 66, 60, 0)
FONTSIZE ( 10 )
INK = Green
LOOP
PRINTAT (0,0); CHR$(1)
WAIT (0.2)
PRINTAT (0,0); CHR$(2)
WAIT (0.2)
REPEAT
END
```

