

With normal, decimal numbers we add them with the following process:

$$\begin{array}{r} 5932 + \\ \underline{1125} \\ 7 \end{array}$$

Start adding at the right hand side.

$$\begin{array}{r} 5932 + \\ \underline{1125} \\ 57 \end{array}$$

Add the next column.

$$\begin{array}{r} 5932 + \\ \underline{1125} \\ 1057 \end{array}$$

Carry if a result is greater than 9.

$$\begin{array}{r} 5932 + \\ \underline{1125} \\ 17057 \end{array}$$

Add the column including the carry.

We do exactly the same with binary numbers. Because binary is base 2, we carry if the result is greater than 2 rather than 9.

$$\begin{array}{r} 1101 + \\ \underline{1100} \\ 1 \end{array}$$

Start at the right hand side.  $1+0=1$

$$\begin{array}{r} 1101 + \\ \underline{1100} \\ 01 \end{array}$$

Continue on.  $0+0=0$

$$\begin{array}{r} 1101 + \\ \underline{1110} \\ 1001 \end{array}$$

This time we have  $1+1=2$ . 2 in binary is 10. So we carry the 1 and put 0 in.

$$\begin{array}{r} 1101 + \\ \underline{1110} \\ 11001 \end{array}$$

This time we have  $1+1+1=3$ . 3 in binary is 11. So we carry the 1 and put 1 in.

$$\begin{array}{r} 1101 + \\ \underline{1110} \\ 11001 \end{array}$$

Finally we have just the carried 1. We write this below to complete the question.

In computers we have limited storage space. We often store numbers in just 1 byte. We can add 1 byte to another byte and put the result into a third byte. For example:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

Byte 1 +  
Byte 2  
=Byte 3

There is a problem with the next calculation though:

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

Byte 1 +  
Byte 2  
=Byte 3

$1+1$  in the first column should equal 10. But we haven't got the storage space in the byte to put another 1 to the left.

Although we can do the calculation the wrong answer will be stored. This is called an **overflow error**.