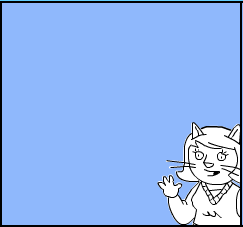
Understanding binary addition Workbook

Name: Date



**Counting binary**

The position of a binary bit shows its value.

1 = 1 1 0 = 2 1 0 0 = 4 1 0 0 0 = 8

In binary if we add 1 0 + 0 1 = 3 or in binary 1 1 one lot of two’s and one lot of ones .If we add 1 1 + 1 = 4 just as 3 + 1 = 4 in binary this is written as 1 0 0.

There are 1 0 types of people in the world.

.

Those who understand binary and those who don’t.

**Values of binary for a byte**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|  |  |  |  |  |  |  |  |  |

If there are zeros in all the columns apart from 16, which has a 1 in, the binary number would be 16.

**Binary addition sums**

0+0 = 0, with no carry,  
1+0 = 1, with no carry,  
0+1 = 1, with no carry,  
 1+1 = 0, and you carry a 1.

**Write in the binary numbers in words:**

Example 110001 is one one zero zero zero one

1 0 is

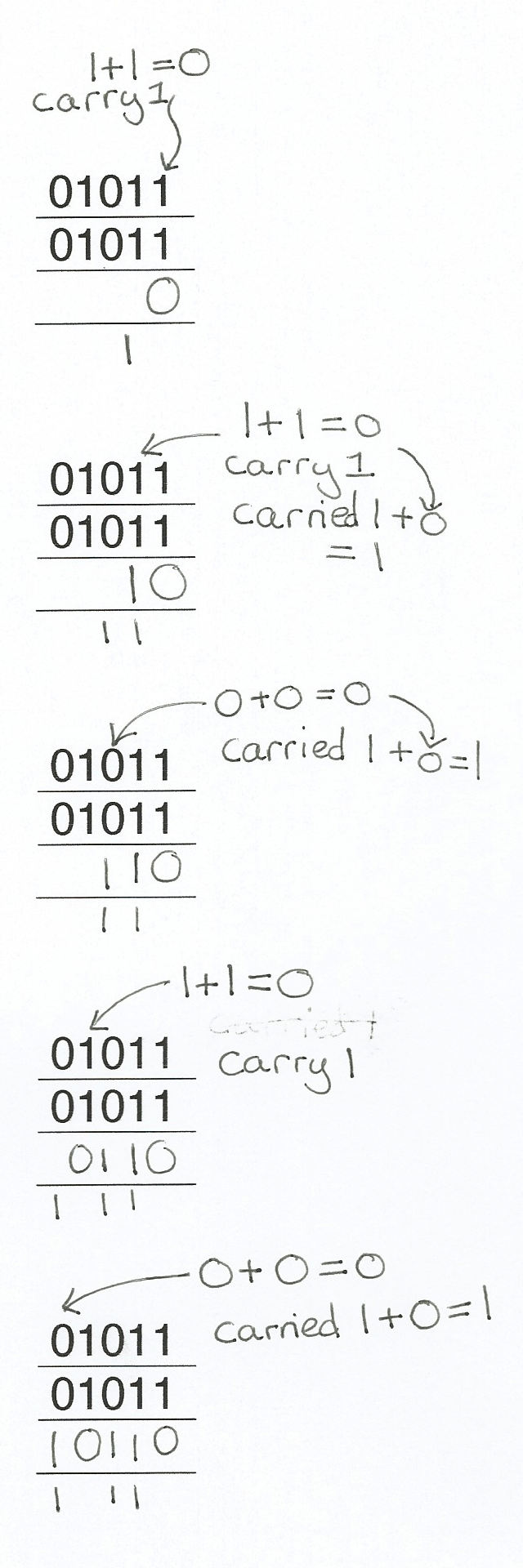
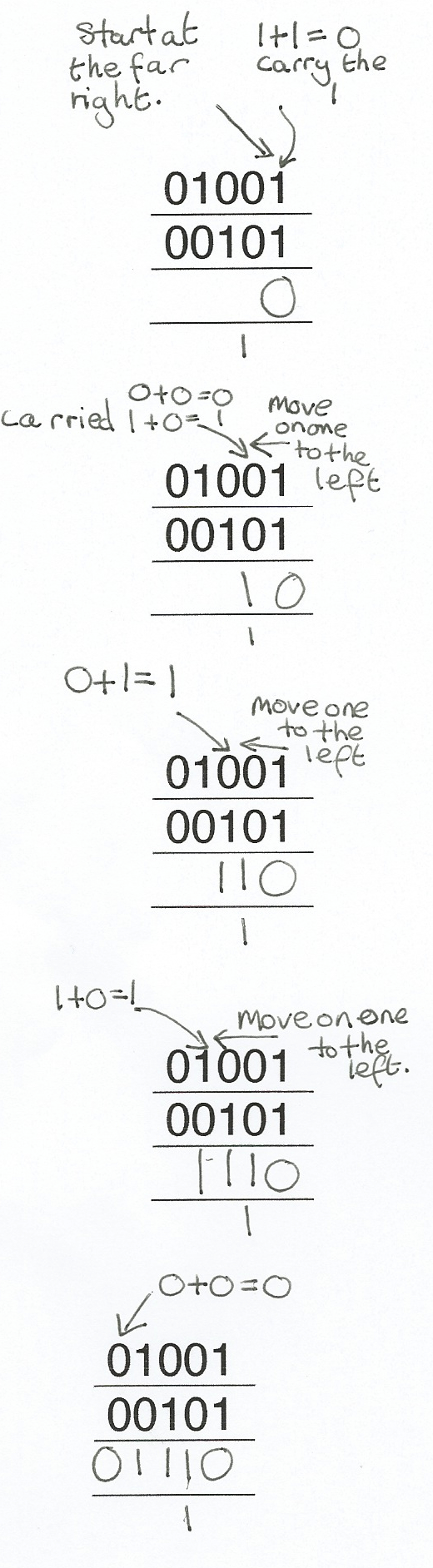
111001 is

010101 is 

**Reading binary numbers**

When you see 10 in binary you should think of this as one zero not ten. The 1 means there is one lot of two and the zero means there are no ones. In the same way when you see 100 in binary you should say one zero zero not one hundred as this means one lot of four, no two’s and no ones.

This comic strip was generated at <http://www.MakeBeliefsComix.com>. Used by permission of author and site creator Bill Zimmerman. All other material used by permission of Ms K. Donkin.



Example 2

Example 1

Worked examples

Binary addition

Remember the second column represents ones and the second column represents twos. You could write one and two above the columns. If you need to carry a one you can write it under the column or above the next column to the left.

|  |
| --- |
| 00 |
| 00 |
|  |

|  |
| --- |
| 01 |
| 00 |
|  |

|  |
| --- |
| 01 |
| 01 |
|  |

Don’t forget to carry the 1

|  |
| --- |
| 00 |
| 01 |
|  |

+

In the same way as addition with denary numbers you start binary addition at the column to the furthest right then add the next column to the left.

|  |
| --- |
| 000 |
| 000 |
|  |

|  |
| --- |
| 000 |
| 010 |
|  |

|  |
| --- |
| 010 |
| 000 |
|  |

|  |
| --- |
| 010 |
| 010 |
|  |

|  |
| --- |
| 001 |
| 010 |
|  |

+

|  |
| --- |
| 011 |
| 010 |
|  |

|  |
| --- |
| 001 |
| 011 |
|  |

|  |
| --- |
| 010 |
| 101 |
|  |

|  |
| --- |
| 110 |
| 001 |
|  |

|  |
| --- |
| 011 |
| 011 |
|  |

+

When you are adding more bits just work along from the right using the same method.

|  |
| --- |
| 01101 |
| 00111 |
|  |

|  |
| --- |
| 01101 |
| 01001 |
|  |

|  |
| --- |
| 01011 |
| 01011 |
|  |

|  |
| --- |
| 00111 |
| 10011 |
|  |

+

|  |
| --- |
| 00101101 |
| 01001111 |
|  |

|  |
| --- |
| 00101101 |
| 01001111 |
|  |

|  |
| --- |
| 00101101 |
| 01001111 |
|  |

Find the spy using binary addition

Top secret information has been stolen from a government database and you have just received a coded message which contains the location of the spy. To decrypt the message add up the binary numbers below and convert the results into denary (base 10).

Then go onto the second page and find the letter for each number. This will spell out the location of the spy.

C:\Users\k.donkin\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\V9I5C78I\MC900070935[1].wmf**Remember** 0+0 =0, 1+0 =1, 0+1=1, 1+1= (carried) 1 0

|  |
| --- |
| 00010011 |
| 00000001 |
|  |

+

Converted into denary =

|  |
| --- |
| 00000101 |
| 00100100 |
|  |

+

Converted into denary =

|  |
| --- |
| 00010000 |
| 00010101 |
|  |

+

Converted into denary =

|  |
| --- |
| 00011001 |
| 00011010 |
|  |

+

Converted into denary =

|  |
| --- |
| 00011001 |
| 00010000 |
|  |

+

Converted into denary =

Use the table to decrypt the message by writing the letter that is above the denary number you found from the binary sums above. Write each letter in the boxes below to reveal to location of the spy.

The spy and the data are in

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G | H | I | J | K | L |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| M | N | O | P | Q | R | S | T | U | V | W | X |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| Y | Z | a | b | c | d | e | f | g | h | i | J |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 31 | 33 | 34 | 35 | 36 |
| k | l | m | n | o | p | q | r | s | t | u | v |
| 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 46 | 48 |
| w | x | y | z | ! | \* | ( | ) | + | = | & | @ |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |

Overflow error

An overflow error occurs in a computer when the result of a calculation is bigger than the place meant to store it can handle. For example register or storage location can hold 8 bits but the result of a binary addition is 9 bits long not all the answer will fit in the location. In this case the answer will be wrong and an error occurs.

Complete the binary additions below, if the answer is more than a byte (8 bits) write in overflow error as the answer.

|  |
| --- |
| 10110101 |
| 00011011 |
|  |

|  |
| --- |
| 11011011 |
| 01111011 |
|  |

|  |
| --- |
| 01011011 |
| 10010110 |
|  |

|  |
| --- |
| 11011011 |
| 11011110 |
|  |