Contents

How to use this book

1 Number Bases

1.1 Decimal (base 10) 6
1.2 Binary (base 2) 8
1.3 Converting between decimal and binary 10
1.4 Hexadecimal (base 16) 12
1.5 Converting between decimal and hexadecimal 14
1.6 Converting between hexadecimal and binary 16

2 Binary Calculation

2.1 Binary addition 18
2.2 Binary addition with carry bits 20
2.3 Add three binary numbers 22
2.4 Overflow errors 24
2.5 Binary multiplication 26
2.6 Binary division 28

3 Data representation

3.1 Digital data 30
3.2 Units of digital data 32
3.3 How much storage? 34
3.4 Representing characters with ASCII 36
3.5 Unicode and extended ASCII 38

4 Digital images

4.1 Bitmap images 42
4.2 Turn a bitstream into an image 44

4.3 Representing colour 46
4.4 Colour depth 48
4.5 Image resolution 50

5 Digital Sound

5.1 The nature of sound 52
5.2 Digital recording 54
5.3 Find the bit rate 56
5.4 Audio file size 58

6 Compression

6.1 The importance of file size 60
6.2 Lossless and lossy compression 62
6.3 Run-length encoding 64
6.4 Dictionary compression 66
6.5 The effect of compression 68
6.6 Find the size of a compressed file 70

7 Huffman compression (AQA only) 72

7.1 The Huffman algorithm 72
7.2 Interpreting a Huffman tree 74
7.3 Creating a Huffman tree 76
7.4 Find the size of a Huffman file 78

Summary questions for chapters 4–7 80

8 Logic circuits

8.1 The NOT gate 82
8.2 The AND gate 84
8.3 The OR gate 86
8.4 Truth tables 88
8.5 Logic circuits 90

Summary questions for chapters 1–3 40
How to use this book

This workbook has been written to support the development of key mathematics skills required to achieve success in your GCSE Computer Science course. It has been devised and written by teachers and the practice questions included reflect the exam-tested content for AQA, OCR, Edexcel and Eduqas specifications.

The workbook is structured into chapters, with each chapter relating to an area of computer science. Then, each topic covers a mathematical skill or skills that you may need to practise. Each topic offers the following features:

1. **Opening paragraph** outlines the mathematical skill or skills covered within the topic.
2. **Worked example** – Annotated worked examples offer a step-by-step breakdown of how to answer questions related to each topic.
3. **Remember** is a useful box that will offer you tips, hints and other snippets of useful information.
4. **Summary questions** are ramped in terms of difficulty and all answers are available at www.oxfordsecondary.co.uk
5. **Stretch yourself** – some of the topics may also contain a few more difficult questions to stretch your mathematical knowledge and understanding.
1 NUMBER BASES

1.1 Decimal (base 10)

Decimal digits

The number system that we use in everyday life is called decimal. There are ten digits that can be used to make a decimal number. Each digit stands for a different number value. The ten digits are: 0 1 2 3 4 5 6 7 8 9

Because decimal uses ten digits, we call it ‘base 10’. Another name for this number system is denary.

Counting in base 10

Although there are only ten digits, the decimal system can represent far more than ten numbers. In fact, any whole number value can be represented using the decimal system. This is because digits have different values that depend on their position in the number.

The first ten decimal values are represented by the digits 0 to 9. Counting up to nine reaches the largest decimal digit. To make the next number:

• reset the 9 back to 0
• put 1 in the tens column.

The number 99 has the largest available decimal digit for both tens and ones. To make the next number:

• reset the tens and ones back to 0
• put 1 in the hundreds column.

Nine is the largest digit in decimal numbers. When counting beyond nine, reset to zero and increase the next position by one.

Place values

The value of each digit in a decimal number is based on its position in that number. Think of the four-digit number 1529. The value of each digit depends on its position in the number.

<table>
<thead>
<tr>
<th>Thousands</th>
<th>Hundreds</th>
<th>Tens</th>
<th>Ones</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>

The 1 is in the thousands column so it stands for 1000. The 5 is in the hundreds column so it stands for 500. The 2 is in the tens column so it stands for 20. The 9 is in the ones column so it stands for 9. The ones column is sometimes called the units column.

The key features of the decimal position grid are as follows.

• The position values start at one, on the right of the grid.
• The values increase from right to left.
• Each position value is ten times bigger than the previous position value.

Why use different number bases?

In this book you will learn about several number bases and why they are important in computer science:

• decimal (base 10) – because the earliest counting systems used fingers, and people have ten fingers
• binary (base 2) – because signals within the computer are stored in binary form
• hexadecimal (base 16) – because converting between binary and hexadecimal is easy.

You will learn more about binary and hexadecimal in the rest of this chapter.

PRACTICE QUESTION

1. Give the value of the digit 6 in the decimal number 106709.
   a six million   b six thousand
   c six hundred   d six hundred thousand

WORKED EXAMPLE

Give the total value of the decimal number 106709.

To calculate the value of a number in any number system:

• multiply each digit value by its position value to give the value represented by that digit
• add together the value represented by each digit to give the total value of the number.

1. Put the digits into the grid to find their position values.

<table>
<thead>
<tr>
<th>Hundred thousands</th>
<th>Ten thousands</th>
<th>Thousands</th>
<th>Hundreds</th>
<th>Tens</th>
<th>Ones</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>

2. Multiply the value of each digit by its position value. In this case the digit 6 in the thousands column stands for 6000. Add the values of all the digits together to give the total value of the number.

   10000 + 6000 + 700 + 0 + 9 = 16709

PRACTICE QUESTION

2. a In the decimal number 192837, state which digit represents the largest value.
   b In the decimal number 456789, give the value that is represented by the digit 4.
   c Give the largest whole number that can be made using each decimal digit once only. Remember to use the digit 0 as well as the others.
   d Give the smallest whole number that can be made using each decimal digit once only.
1.2 Binary (base 2)

Binary digits (bits)

Computer scientists often work with binary numbers. ‘Binary’ means base 2. There are two digits that can be used to make a binary number:

0 1

Binary digits are also called bits. Bit is short for binary digit.

Any number that can be represented in decimal can also be represented in binary.

Counting in base 2

Although there are only two values for each bit, 1 and 0, the binary system can use these two values to represent any whole number.

The first two binary values are represented by single bits:

0 1

There are now no more binary digits available to you to increase the value of this column. To move to the next number value, you should reset the units back to the 0 and put 1 in the next column to the left. The sequence is now:

0 1
1 10
10 11
11 100
100 101
101 110
110 111
111 1000

This process of resetting to 0 and adding a 1 in the column to the left repeats. The sequence of binary numbers from 0 upwards is shown in Table 1.

Table 1 Binary numbers representing values from zero to eight

<table>
<thead>
<tr>
<th>Binary number</th>
<th>Decimal number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>110</td>
<td>6</td>
</tr>
<tr>
<td>111</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
</tr>
</tbody>
</table>

PRACTICE QUESTION

1 Extend the sequence in Table 1 to show all binary numbers up to a value of 15.

Place values in binary

Each bit in the binary number has a different value based on its position.

The position on the right has the value 1. The values increase from right to left. Each position value is double the previous position value. This grid shows the first eight position values.

<table>
<thead>
<tr>
<th></th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To find the value of a binary number, place the bits of the number into the binary grid, following these rules:

• Each position must only hold one bit – a 1 or a 0.

• The bits should be filled in starting on the right of the grid. Any empty spaces on the left of the grid should be filled with a 0.

For example, placing the number 1001 in the grid would look like this:

<table>
<thead>
<tr>
<th></th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Then add together the value of every position where there is a 1.

\[ 8 + 1 = 9 \]

WORKED EXAMPLE

Calculate the decimal value of the binary number 0001 0100.

To work this out, enter the 1s and 0s into the grid that shows position values.

<table>
<thead>
<tr>
<th></th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001 0100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Find the value of every 1 in the number.

There is a 1 in the 16 position.

There is a 1 in the 4 position.

Add these position values together.

\[ 16 + 4 = 20 \]

The value of the binary number 0001 0100 in decimal form is 20.

PRACTICE QUESTIONS

2 Calculate the decimal value of the following binary numbers.
   a 0010 0010   b 0001 0100   c 0100 0001   d 1000 1000

3 Calculate the decimal value of each of these binary numbers. The numbers include more 1s so you will need to add more values together.
   a 0110 1110   b 1101 0110   c 0110 1111   d 1011 1101

NOTE: Binary numbers are often written using 8 bits. To make the number easier to read, the bits are split into two groups of four, for example, 0010 1101.

NOTE: You do not pronounce binary numbers as if they were decimal numbers. For example, the binary number 10 is not pronounced ‘ten’. To name a binary number, name its digits in order. You might say ‘one oh’ or ‘one zero’.

REMEMBER: To convert binary to decimal, place the bits in the binary grid. Add the values of all the positions where there is a 1.
1.3 Converting between decimal and binary

Convert binary to decimal

To convert a binary number to decimal – add up the binary position values as follows.

- Place the bits of the binary number into the binary position grid.

<table>
<thead>
<tr>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There is no need to memorise the position values. Simply start at 1 on the right and double the value each time as you move to the left.

- Remember to place the bits as far to the right as possible, and fill any empty spaces on the right hand side with zeros.
- Finally, add together the values of all positions where there is a 1.

**PRACTICE QUESTION**

1. Convert these binary numbers to decimal.
   - a 0011 0011
   - b 0101 0110
   - c 0110 0111
   - d 1001 1001

Convert decimal to binary

To convert a decimal number to binary – subtract the binary position values by:

- subtracting the largest value possible without making a negative result
- putting a 1 under each value that you subtract in the grid
- continuing to subtract position values until the result is 0.

**WORKED EXAMPLE**

Convert the decimal number 50 to binary.

Look at the grid of binary position values. The largest number that can be subtracted from 50 is 32. Place a 1 under that value.

<table>
<thead>
<tr>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

50 − 32 = 18

The result is 18. The largest number that can be subtracted from 18 is 16. Place a 1 under that value.

<table>
<thead>
<tr>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

18 − 16 = 2

The result is 2. Place a 1 under that value.

<table>
<thead>
<tr>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Subtracting 2 reduces the result to 0 so the process is over. Put zeros into all the remaining empty positions.

To get the final answer, show the bits without the grid. To make them easier to read you can split them into two groups of four:

0011 0010

**PRACTICE QUESTIONS**

2. Convert these decimal numbers to binary.
   - a 23
   - b 99
   - c 60
   - d 88

3. Convert these larger decimal numbers to binary.
   - a 154
   - b 201
   - c 222
   - d 198

**REMEMBER:**

To convert decimal to binary, subtract the values in the binary grid. Start with the highest values. Only subtract if the result is greater than 0. Put a 1 into every position where you have subtracted a value.
1.4 Hexadecimal (base 16)

Hexadecimal digits

Hexadecimal is base 16. Hexadecimal numbers are made from 16 digits.

0 1 2 3 4 5 6 7 8 9 A B C D E F

The first ten digits (0 to 9) have the same value as decimal digits. The next six digits (A to F) represent the numbers 10 to 15.

Counting in base 16

The 16 hexadecimal digits count up from 0 to 15, as shown in Table 1.

Table 1 Hexadecimal digits up to a value of 15

<table>
<thead>
<tr>
<th>Decimal value</th>
<th>Hexadecimal digit</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
</tr>
</tbody>
</table>

Counting beyond 15

F is the largest hexadecimal digit. It represents the decimal value 15. To count beyond F:

• reset the ones column to 0
• add 1 to the next column.

So the decimal number 16 is shown as 10 in hexadecimal, 17 in decimal is shown as 11 in hexadecimal, and so on.

PRACTICE QUESTION

1. Extend Table 1 to give the hexadecimal numbers that represent the decimal values from 16 to 31.

Position values

The decimal values from 0 to 255 can be represented by two-digit hexadecimal numbers. The value of each hexadecimal digit is found by multiplying the digit value by the position value. The total value of the number is found by adding the digit values together.

With a two-digit hexadecimal number this means:

• multiplying the digit on the left by 16
• adding the value of the digit on the right.

WORKED EXAMPLE

Calculate the decimal value of the hexadecimal number 34.

Place the digits of the number in the hexadecimal grid.

<table>
<thead>
<tr>
<th>Sixteens</th>
<th>Ones</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

The digit on the left is 3. Multiply that by 16 to give the value 48.

\[(3 \times 16) = 48\]

Then add the digit on the right.

\[48 + 4 = 52\]

So 34 in hexadecimal is 52 in decimal.

PRACTICE QUESTION

2. Convert these hexadecimal numbers into decimal numbers.
   a 21
   b 45
   c 60
   d 77

NOTE: Some hexadecimal digits are letters of the alphabet but not all hexadecimal numbers include digits of this type. You can’t tell whether a number is decimal or hexadecimal just by checking whether it has letters in it. That is why it is always important to state what number base you are using.

WORKED EXAMPLE

Calculate the decimal value of the hexadecimal number A4.

Some hexadecimal numbers include the digits that stand for the values 10 to 15. The same method is used but you must work out the value of the digit before you multiply it by the position value.

<table>
<thead>
<tr>
<th>Sixteens</th>
<th>Ones</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
</tr>
</tbody>
</table>

The digit on the left is A. This is equivalent to the decimal value 10. Multiply that by 16 to give the value 160.

\[10 \times 16 = 160\]

Then add the digit on the right.

\[160 + 4 = 164\]

The hexadecimal number A4 is 164 in decimal.

PRACTICE QUESTION

Give the decimal equivalent of these hexadecimal numbers.
   a 2B
   b C5
   c E0
   d 7F

REMEMBER: To convert a two-digit hexadecimal number to decimal: Find the values of the two digits. Multiply the left digit by 16. Add the right digit to give the total value.