Candidates should be able to:

OCR: 2.1.4 (c) convert positive denary whole numbers (0-255) into 8-bit binary numbers and vice versa

AQA: 3.1.10 understand how binary can be used to represent positive whole numbers (up to 255)

CS Unplugged binary numbers

Unplugged - Part 1 Binary
http://www.youtube.com/watch?v=b6vHZ95XDwU

Unplugged: Part 2: Binary - Counting
http://www.youtube.com/watch?v=Pz7dLWvi2w0

Cisco Binary Game
http://forums.cisco.com/CertCom/game/binary_game_page.htm

Posters
Wall posters can help students become familiar and comfortable with binary numbers. These can include tables of binary numbers, powers of two and (later) hexadecimal numbers. Eye-catching posters that appear to defy the normal laws of logic and arithmetic can be very effective and provide talking points. For example:

![Progress through Binary thought.](image1)

![There are 10 types of people in the world: Those who understand binary, and those who don’t.](image2)

![Binary Su Doku](image3)
Decimal to Binary Conversion


Programming Algorithms

The algorithm to work out binary to decimal conversion is quite simple:

\[(128 \times 1) + (32 \times 1) + (16 \times 1) + (8 \times 1) + (2 \times 1) + (1 \times 1)\]

The algorithm to work out decimal to binary conversion involves selection:

Set all bits to 0
If the number is >= 128 then: set bit 8 to 1, take away 128 from the number
If the number is now >=64 then: set bit 7 to 1, take away 64 from the number
If the number is now >=32 then: set bit 6 to 1, take away 32 from the number
If the number is now >=16 then: set bit 5 to 1, take away 16 from the number
If the number is now >=8 then: set bit 4 to 1, take away 8 from the number
If the number is now >=4 then: set bit 3 to 1, take away 4 from the number
If the number is now >=2 then: set bit 2 to 1, take away 2 from the number
If the number is now 1 then: set bit 1 to 1
Programming Challenge

Invite students to write the following number conversion programs:
Program 1: Input a binary number. Convert it to denary and output the result.
Program 2: Input a denary number. Convert it to binary and output the result.

If necessary, the students can be given the following algorithms.

Program 1
Output "Enter a binary number 
Input BinaryString
BinaryLength = Length of BinaryString
Multiplier = 1
DenaryValue = 0
For n = BinaryLength Down To 1
   Digit = nth character in BinaryString
   DenaryValue = DenaryValue + Digit * Multiplier
   Multiplier = Multiplier * 2
Next n
Output "The denary equivalent is ", DenaryValue

Program 1 alternative
If students have not done much programming this program could be written in visual basic using forms with a single textbox for each binary digit and a separate textbox for denary output. The code is then reduced a button click event:

Denarytextbox.text = binarytextbox1.text*1 + binarytextbox2.text*2 + binarytextbox3.text*4

Program 2
Output "Enter a denary number 
Input DenaryValue
BinaryString = 
While DenaryValue > 0
   BinaryString = CStr(DenaryValue Mod 2) + BinaryString
   DenaryValue = DenaryValue Div 2
Endwhile
Output "The binary equivalent is ", BinaryString

Program 2 alternative for an 8 bit binary number
Output "Enter a denary number between 0 and 255"
Input DenaryValue
BinaryString = 
For i =8 to 1 step -1
   If DenaryValue >= (2 ^ (i - 1)) then
      BinaryString = BinaryString + “1”
      DenaryValue = DenaryValue - (2 ^ (i - 1))
   else
      BinaryString = BinaryString + “0”
   Endif
Next
Output "The binary equivalent is ", BinaryString
Candidates should be able to:
OCR: 2.1.4 (d) add two 8-bit binary integers and explain overflow errors which may occur

Binary Arithmetic

Candidates should be able to:
OCR: 2.1.4 (e) convert positive denary whole numbers (0-255) into 2-digit hexadecimal numbers and vice versa
AQA: 3.1.10 know how to convert between binary, denary and hexadecimal

The place values for base 10 are 1,10,100,1000 etc
The place values for base 2 are 1,2,4,16,32,64,128,256 etc
The place values for base 16 are 1,16,256 etc

Following on from the discussion about place value in binary (base 2) hexadecimal numbers (base 16) have 16 different digits:

<table>
<thead>
<tr>
<th>Denary</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>255</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hexadecimal</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>10</td>
<td>11</td>
<td>FF</td>
</tr>
</tbody>
</table>

To convert from 2-digit hexadecimal to denary:

Example: To convert hexadecimal AF into denary 175:
Convert the least significant digit directly using the table above. (F=15)
Convert the second digit using the table and then multiply it by 16. (A = 10 so 10 x 16 = 160)
Add the 2 numbers together. (15 + 160 = 175)

To convert denary into 2-digit hexadecimal:

Example: To convert denary 175 into hexadecimal AF
Divide the denary number by 16. (175 / 16 = 10 remainder 15).
To get the least significant digit, convert the remainder directly into hexadecimal. (15 = F)
To get the next significant digit, convert the quotient directly into hexadecimal. (10 = A)
Programming Challenge

Invite students to write the following number conversion programs:

Program 1: Input a hexadecimal number as a text string. Convert it to denary and output the result.
Program 2: Input a denary number. Convert it to hexadecimal and output the result.

If necessary, the students can be given the following algorithms.

Program 1

Output "Enter a hexadecimal number 
Input HexString
HexLength = Length of HexString
Multiplier = 1
DenaryValue = 0
For n = HexLength Down To 1
    Digit = nth character in BinaryString
    Select Case Digit
        Case "A"
            DenaryDigit = 10
        Case "B"
            DenaryDigit = 11
        Case "C"
            DenaryDigit = 12
        Case "D"
            DenaryDigit = 13
        Case "E"
            DenaryDigit = 14
        Case "F"
            DenaryDigit = 15
        Case Else
            DenaryDigit = Digit
    End Select
    DenaryValue = DenaryValue + (DenaryDigit * Multiplier)
    Multiplier = Multiplier * 16
Next n
Output "The denary equivalent is ", DenaryValue
Program 2
Output "Enter a denary number 
Input DenaryValue
HexString = ""
While DenaryValue > 0
    DenaryString = CStr(DenaryValue Mod 16)
    Select Case DenaryString
        Case 10
            HexString = "A"
        Case 11
            HexString = "B"
        Case 12
            HexString = "C"
        Case 13
            HexString = "D"
        Case 14
            HexString = "E"
        Case 15
            HexString = "F"
    End Select
    HexValue = HexString + HexValue
    DenaryValue = DenaryValue Div 16
Endwhile
Output "The hexadecimal equivalent is ", HexValue

Candidates should be able to:
OCR: 2.1.4 (f) convert between binary and hexadecimal equivalents of the same number
AQA: 3.1.10 know how to convert between binary, denary and hexadecimal

<table>
<thead>
<tr>
<th>Place value</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary number</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Calculation</td>
<td>1<em>8 + 1</em>4 + 1<em>2 + 1</em>1</td>
<td>1<em>8 + 1</em>4 + 1<em>2 + 1</em>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal number</td>
<td>15</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hexadecimal number</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary number</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Calculation</td>
<td>1<em>8 + 1</em>4 + 1<em>2 + 1</em>1</td>
<td>1<em>8 + 1</em>4 + 1<em>2 + 1</em>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decimal number</td>
<td>7</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hexadecimal number</td>
<td>7</td>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Binary and Hexadecimal dominoes
Divide the students into groups of three or four and give one set of dominoes to each group. Ask the students to divide the dominoes evenly between themselves. One student puts down their first domino. The next student has to see if they have a domino that matches either end of the domino on the table. For example, they may match a binary number to its denary equivalent, or a hex number to its binary equivalent. There are a variety of ways in which the dominoes may be matched.

Binary to Decimal to Hexadecimal Converter
http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html

Binary to Decimal to Hexadecimal Dominoes

Candidates should be able to:
OCR: 2.1.4 (g) explain the use of hexadecimal numbers to represent binary numbers.
AQA: 3.1.10 understand why hexadecimal number representation is often used

Hexadecimal Notation