# **Data Representation Booklet**

|  |  |
| --- | --- |
| **Student Name** |  |
| **Group** |  |
| **Year**  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Level** | **Assessment Criteria** | **Achieved (Student)** | **Achieved (Teacher)** |
| 2 | 1. I know different types of data: text, number.
2. I know programs use different data types
 |  |  |
| 3 | 1. I Know that computers use binary to understand what to do
2. I can explain the difference between data and information
 |  |  |
| 4 | 1. I can change a denary number to binary number and vice versa
2. I can convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice
 |  |  |
| 5 | 1. I can explain that binary is at the base of all actions on a computer and all programs have to translate to binary to work
2. I understand the relationship between the number of bits per character in a character set
3. I understand the number of characters which can be represented (for example ASCII, extended ASCII and Unicode).
 |  |  |
| 6 | 1. I can complete binary addition
2. I can explain how data is stored in memory
3. I understand how numbers, images, sound and character sets use the same bit patterns.
4. I recognised how Metadata is included in the file
 |  |  |
| 7 | 1. I understand the relationship between data representation and data quality
2. I understand the need for compression and can explain the two types of compressions techniques (Lossy and Lossless)
 |  |  |

# **Part 1: Unit of Data**

### Exercise 1: Unit of Data

1. Why do computers use binary?
2. What is the smallest unit of data?
3. What is a byte?
4. What is a Nibble?
5. What is a Kilobyte?
6. What is a Megabyte?
7. What is a Gigabyte?
8. What is a Terabyte?

***The size of a music file is typically 5MB:***

1. How many music files can you store in 8GB drive?
2. How many music file can you store in 16GB drive?

# **Part 2: Binary to Denary**

## Converting from 8-bit Binary to Denary

**Method:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Step 1: | Write out the bits | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Step 2: | Write the binary number underneath | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| Step 3: | Write out the corresponding values | 128 | 0 | 32 | 0 | 8 | 4 | 2 | 0 |
| Step 4: | Add Values together | 128 +32+8+4+2 = 174 |

**Example:**

8-bit binary number = 01111001

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 64 | 32 | 16 | 8 | 0 | 0 | 1 |
| 64 + 32 + 16 + 8 + 1 = **121** |

### Exercises 2: Binary to Denary

Convert the following 8-bit binary numbers into denary. Show your working.

1. 11100011

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

1. 01010101

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

1. 00110011

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |

1. 00010010
2. 11010110
3. 11010110
4. 11100011
5. 01010101

# **Part 3: Denary to Binary**

## Converting from Denary to 8-bit Binary

**Method:**

* **Step 1:** Draw the table (starting at 1 to 128)
* **Step 2:** Find out all the numbers we need to add together to get the total
* **Step 3:** Turn all the numbers we add together **ON** by adding **1** under them and **OFF** on the rest by adding **0**

**Example:**

Denary Number = 121

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | 64 | 32 | 16 | 8 | 0 | 0 | 1 |
| 64 + 32 + 16 + 8 +1 = 121 |

### Exercises 3: Denary to Binary

Convert the following denary numbers into 8-bit Binary. Show your workings.

1. 27

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

1. 111

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

1. 45

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

1. 99
2. 86
3. 106
4. 197
5. 237
6. 247
7. 251

### Challenge 1: Denary to Binary

Use the approach above to convert the following numbers into Binary.

|  |
| --- |
| 34 |
| 115 |
| 99 |

### Challenge 2: Binary to Denary Converter

Change these numbers from Binary to Denary or Denary to Binary.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 64 | 32 | 16 | 8 | 4 | 2 | 1 |  | 100 | 10 | 1 |
| Binary |  | Denary |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |  |  |  |  |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |  |  |  |  |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |  |  |  |  |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |  |  |  |  |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | 1 |
|  |  |  |  |  |  |  |  |  |  | 2 |
|  |  |  |  |  |  |  |  |  |  | 3 |
|  |  |  |  |  |  |  |  |  |  | 4 |
|  |  |  |  |  |  |  |  |  |  | 5 |
|  |  |  |  |  |  |  |  |  |  | 6 |
|  |  |  |  |  |  |  |  |  |  | 7 |
|  |  |  |  |  |  |  |  |  |  | 8 |
|  |  |  |  |  |  |  |  |  |  | 9 |
|  |  |  |  |  |  |  |  |  | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |  |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |  |  |  |  |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |  |  |  |  |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |  |  |  |  |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |  |  |  |  |
|  |  |  |  |  |  |  |  | 1 | 0 | 0 |
|  |  |  |  |  |  |  |  |  | 9 | 9 |

# **Part 4: Binary to Hexadecimal (Hex)**

## Converting between 8-bit Binary to 2-digit Hex

**Why Use Hexadecimal**

Large binary numbers are hard to remember. Therefore, programmers use hexadecimal values because:

* Each digit represents exactly 4 binary digits;
* Hexadecimal is a useful shorthand for binary numbers;
* Hexadecimal still uses a multiple of 2, making conversion easier whilst being easy to understand;
* Converting between denary and binary is relatively complex;
* Hexadecimal is much easier to remember and recognize than binary;
* This saves effort and reduces the chances of making a mistake

**Method:**

1. Convert 1st nibble into 1st Hex digit and Vice versa
2. Convert 2st nibble into 2st Hex digit and Vice versa

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Binary** | **0000** | **0001** | **0010** | **0011** | **0100** | **0101** | **0110** | **0111** | **1000** | **1001** | **1010** | **1011** | **1100** | **1101** | **1110** | **1111** |
| **Hex** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **A** | **B** | **C** | **D** | **E** | **F** |

**Example:**

Converting Binary to Hex

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01101010 |  | 00010010 |  | 11110011 |
| 0110 | 1010 |  | 0001 | 0010 |  | 1111 | 0011 |
| 6 | A |  | 1 | 2 |  | 15 (F) | 3 |
| Answer = **6A** |  | Answer = **12** |  | Answer = **F3** |

Converting Hex to Binary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6A |  | 12 |  | F3 |
| 6 | A |  | 1 | 2 |  | F | 3 |
| 0110 | 1010 |  | 0001 | 0010 |  | 1111 | 0011 |
| Answer = **01101010** |  | Answer = **00010010** |  | Answer = **11110011** |

**Exercises:**

Convert the following 8-bit binary numbers into their 2-digit Hex equivalents:

1. 01011100
2. 01001101
3. 11001110
4. 10111001
5. 00001001
6. 10010110
7. 00101101
8. 10110111
9. 01011111
10. 10001010

Convert the following 2-digit Hexadecimal numbers into their 8-bit binary equivalents:

1. 6D
2. 5C
3. AC
4. D9
5. B7
6. 0A
7. CC
8. 99
9. 4F
10. CB

# **Part 5: Hexadecimal (Hex) to Denary**

## Converting from 2-digit Hex to Denary

**Remember:**

Denary is base 10 i.e. 0123456789

Hex is base 16 i.e. 0123456789 ABCDEF

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Denary | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Hex | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

**Method 1:**

1. Convert left number (MSB) to denary value & multiply by 16
2. Convert right denary (LSB) and add to results from *Step 1*

**Example:**

Convert Hexadecimal value **F6** to denary

|  |  |  |  |
| --- | --- | --- | --- |
| MSB Denary Value  | MSB \*16 | LSB Denary Value | Total |
| 15 | 15\*16 = **240** | **6** | 240 + 6 = **246** |

**Activity:**

Convert the following 2-digit hexadecimal numbers into denary. Show your working

1. F9

|  |  |  |  |
| --- | --- | --- | --- |
| MSB Denary Value | MSB \*16 | LSB Denary Value | Total |
|  |  |  |  |

1. DD

|  |  |  |  |
| --- | --- | --- | --- |
| MSB Denary Value | MSB \*16 | LSB Denary Value | Total |
|  |  |  |  |

1. BB

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

1. 7C
2. 6F

**Method 2:**

1. Convert hex 2 numbers into two binary nibble
2. Add the two binary numbers together
3. Convert the binary

**Example:**

Convert Hexadecimal value **F6** to denary

|  |
| --- |
| F6 |
| F = 15 | 6 |
| 1111 | 0110 |
| Answer = **11110110**  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 128 | 64 | 32 | 16 | 0 | 4 | 2 | 0 |
| 128 + 64 + 32 + 16 + 4 + 2 = **246** |
| **F6 = 146** |

**Activity**

 Convert the following Hexadecimal value**s** to denary

1. **D7** to denary

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

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

1. **DB** to denary

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

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

1. F5

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

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

1. 6A

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

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

1. 68

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

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

1. 9B

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

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

1. AB
2. 97
3. AA
4. CB

# **Part 6: Denary to Hexadecimal (Hex)**

## Converting from Denary to 2-digit Hex

**Method 1:**

1. Divide denary number by 16
2. Convert Whole number into Hex equivalent
3. Convert remainder into Hex equivalent
4. Put them together

**Example:**

Convert 237 to 2-digit hexadecimal

|  |  |  |  |
| --- | --- | --- | --- |
| Denary Value | /16 Whole Number | /16 Remainder |  |
| 237 | 14 | 13 |  |
|  | Hex Equivalent | Hex Equivalent | ANSWER |
|  | E | D | **ED** |

**Exercises:**

Convert the following denary numbers into 2-digit hexadecimal. Show your working

1. 239

|  |  |  |  |
| --- | --- | --- | --- |
| Denary Value | /16 Whole Number | /16 Remainder |  |
|  |  |  |  |
|  | Hex Equivalent | Hex Equivalent | ANSWER |
|  |  |  |  |

1. 193

|  |  |  |  |
| --- | --- | --- | --- |
| Denary Value | /16 Whole Number | /16 Remainder |  |
|  |  |  |  |
|  | Hex Equivalent | Hex Equivalent | ANSWER |
|  |  |  |  |

1. 255

|  |  |  |  |
| --- | --- | --- | --- |
| Denary Value | /16 Whole Number | /16 Remainder |  |
|  |  |  |  |
|  | Hex Equivalent | Hex Equivalent | ANSWER |
|  |  |  |  |

1. 169
2. 99

**Method 2:**

1. Convert denary number into 8-bit binary
2. Convert 1st nibble into 1st Hex digit and Vice versa
3. Convert 2st nibble into 2st Hex digit and Vice versa

**Remember:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Denary | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Hex | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

**Example:**

Convert 237 to 2-digit hexadecimal

**Example:**

Denary Number = 237

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 64 | 32 | 0 | 8 | 4 | 0 | 1 |
| 128 + 64 + 32 + 8 + 4 + 1 = 237 |

|  |
| --- |
| 11101101 |
| 1110 | 1101 |
| 14 = E | 13 = D |
| Answer = **ED** |

**Activity:**

Convert the following number to 2-digit hexadecimal

1. 201

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

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

1. 78

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

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

1. 155

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

|  |
| --- |
|  |
|  |  |
|  |  |
|  |

1. 222
2. 247

# **Part 7: Binary Addition**

## Adding Binary Numbers

**Method:**

This is just the same as normal addition, but working only with 1s and 0s

**FOLLOW THE RULES!**

|  |  |
| --- | --- |
| Rule 1 | Start with the least significant bit (LSB) – the one on the right! |
| Rule 2 | 0 + 0 = 0 |
| Rule 3 | 0 + 1 = 1 |
| Rule 4 | 1 + 1 = 0 Carry 1 |
| Rule 5 | 1 + 1 + 1 = 1 Carry 1 |

**Example:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| + | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| Carry |  |  | 1 | 1 |  |  |  |  |
| Answer  | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

Practice task:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 0 | 0 | 1 |  |  |  |  | 1 | 1 | 0 | 1 | 1 |
| + | 0 | 1 | 0 | 1 |  |  |  | + | 0 | 0 | 0 | 1 | 1 |
| Carry |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Answer |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 0 | 0 | 0 |  |  |  |  | 0 | 0 | 0 | 0 | 1 |
| + | 0 | 1 | 0 | 1 |  |  |  | + | 0 | 1 | 0 | 1 | 0 |
| Carry |  |  |  |  |  |  |  | Carry |  |  |  |  |  |
| Answer |  |  |  |  |  |  |  | Answer |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Exercises 4: Binary Addition

Add the following pairs of 8-bit binary numbers. Show your working.

1. 00101000 + 00110101

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| + |  |  |  |  |  |  |  |  |
| Carry |  |  |  |  |  |  |  |  |
| Answer  |  |  |  |  |  |  |  |  |

1. 00011010 + 10100010

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| + |  |  |  |  |  |  |  |  |
| Carry |  |  |  |  |  |  |  |  |
| Answer  |  |  |  |  |  |  |  |  |

1. 10010010 + 10101101

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| + |  |  |  |  |  |  |  |  |
| Carry |  |  |  |  |  |  |  |  |
| Answer  |  |  |  |  |  |  |  |  |

1. 01001001 + 01110100

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| + |  |  |  |  |  |  |  |  |
| Carry |  |  |  |  |  |  |  |  |
| Answer  |  |  |  |  |  |  |  |  |

1. 00110111 + 10000010

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| + |  |  |  |  |  |  |  |  |
| Carry |  |  |  |  |  |  |  |  |
| Answer  |  |  |  |  |  |  |  |  |

1. 00011001 + 00111001

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
| + |  |  |  |  |  |  |  |  |
| Carry |  |  |  |  |  |  |  |  |
| Answer  |  |  |  |  |  |  |  |  |

### Challenge 1 : Adding Binary Task

We can add binary numbers by columns.

If the total for a column is two (in binary 10) we write down 0 and “carry 1” to the next column.

If the total for a column is three (in binary, 11) we write down 1 and carry 1.

Try these:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | (169) |
| + | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | (42) |
|  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | (91) |
| + | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | (57) |
|  |  |  |  |  |  |  |  |  |  |

What about this one?

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

This is a special case which results in an “overflow”, which is technically an error when we use eight bit bytes. The “Overflow” issue often comes up as an exam question – learn it!

**3a:** Adding Binary Numbers *(week beginning 11/1/16)*

**Using the method demonstrated, complete the following questions:**

1. 10011

+ 1111101

1. 01001011

+ 01010011

1. 00010001

+ 00001100

1. 011

+ 11110110

1. 11001111

+ 11111111

1. 101001

+ 01110011

**What happens in question f? What is this called and what does it mean?**

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

|  |  |
| --- | --- |
| SUCCESSES: |  |
| TARGETS: |  |
| ACTION: |  |

# **Part 8: Binary Shift**

## Multiplying in Binary (Binary Left Shift)

As well as adding, we need to know how to multiply and Divide in Binary. Let’s start with Binary Multiplication.

Let’s look at how to calculate **3 x 8**

**Method:**

1. We have to multiply in factors of 2 so 3 x 8 is actually…

 3 x 23  (2 x 2 x 2 = 8)

1. We then need to convert the 3 into binary which is 0 0 1 1
2. Because we need to multiply to the power of 3 We need to move the Binary 3 places to the left.

0 0 1 1 0 0 0

1. We know 3 x 8 = **24**
2. Let’s check the denary for 0 0 1 1 0 0 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
|  |  |  | 16 | 8 | 0 | 0 |  |
| **16 + 8 = 24** |

**Example:**

5 x 16

1. 5 x 24
2. 5 in binary is 0 1 0 1
3. Need to move 0 1 0 1 four places to the left 0 1 0 1 0 0 0 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 64 | 0 | 16 | 0 | 0 | 0 | 0 |
| **64 + 16 = 80** |

5 x 60 is **80**

The binary conversion proves this.

**Exercises:**

1. **12 x 16**

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

1. **6 x 4**

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

1. **7 x 8**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |

1. **9 x 4**
2. **11 x 16**

## Dividing in Binary (Binary Right Shift)

Now let’s take a look at Dividing in binary Let’s look at how to calculate **16 / 4**

**Method and Example**

1. We have to divide in factors of 2 so 16 / 4 is actually…
2. / 22  (2 x 2 = 4)
3. We then need to convert the 16 into binary which is 1 0 0 0 0
4. Because we need to divide to the power of 2 We need to move the Binary 2 places to the right.

0 0 0 0 0 1 0 0

1. We know 16/4 = 4
2. Let’s check the denary for 0 0 0 0 0 1 0 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 |
| **4** |

16/ 4 = 4

The binary conversion shows this.

**Exercises:**

1. **8 / 4**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |

1. **20 / 2**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |

1. **36 / 4**
2. **184 / 4**
3. **156 / 8**

# **Part 9: Representation of Characters**

### Exercise 5: Binary to ASCII

A scout in the borderlands of Computasia has released a coded message. Can you decode it?

01010100 01101000 01100101 00100000 01100100 01110010 01100001 01100111 01101111 01101110 01110011 00100000 01100001 01110010 01100101 00100000 01100011 01101111 01101101 01101001 01101110 01100111 00100001 00100000 01000001 01110010 01100101 00100000 01111001 01101111 01110101 00100000 01110010 01100101 01100001 01100100 01111001 00111111

|  |
| --- |
|  |



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. What is the meant by UNICODE?

|  |
| --- |
|  |

1. What is the different between UNICODE and ASCII code

|  |
| --- |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|

|  |  |
| --- | --- |
|  |  |
| **3** | The character é is part of a computer's character set.Describe what is meant by a character set.

|  |
| --- |
|  |

 |

 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |
| --- | --- |
|  |  |
| **4.** | When sending text messages using a mobile phone, people can choose from hundreds of characters, called emoji, to insert in their message. An example of an emoji is C:\core\files\questions\1493051405\J275ComputingA451-012016Jun\img\p2_01_150.png.The Unicode character code for the emoji C:\core\files\questions\1493051405\J275ComputingA451-012016Jun\img\p2_01_150.pngin hexadecimal is 1F64A.Convert the hexadecimal number 1F64A to binary.The first three hexadecimal digits have been done for you.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Hexadecimal:** | 1 | F | 6 | 4 | A |
| **Binary:** | 0001 | 1111 | 0110 |  .............. |  .............. |

1. Describe how Extended ASCII works.

|  |
| --- |
|  |

1. Explain why mobile phones that can send emoji would use Unicode instead of ASCII as their character set.

|  |
| --- |
|  |

 |

 |

 |

# **Part 10: Representation of Image**

 If you look REALLY closely at a digital picture, it looks like this……….

**\***

**\***

0 100 200 300 400

500

400

300

200

100

0

**Y**

**X**



Technology can’t see things in analogue. They have to break the image up in to a grid of little squares …like you just did!

This is called DIGITISING

Each square is called a PIXEL. In Fireworks or Photoshop, there are usually 72 ppi (pixels per inch).

Each pixel has binary code that tells the device where it belongs in the picture. Kind of like a grid reference or a cell address. This \*pixel\*, could have an address of; Y150, X200

Each pixel also has code that tells the device what colour it is.

**How computers create images using Binary**

This tasks will demonstrate how computers use binary to represent images on our computer screens. You will read code to create images and then may have a go at creating your own image and the code that is needed for somebody else to recreate it.

### Exercise 6: Computasia: Image of the Beast!

An enterprising scout recorded an image of one of the beasts terrorising the villages using binary. He sent the message as follows:

0 0 1 0 0 0 0 0 1 0 0 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1

Decipher it to create a black and white image of the beast (0 = white; 1=Black)?

Q: What is the relation between the amount of pixels, the quality of the image and the size of the image file?

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |

**Activity 1:**

What’s a BIT?

|  |
| --- |
|  |

And a BYTE?

|  |
| --- |
|  |

Can you find out what analogue means?

|  |
| --- |
|  |

**Activity 2 : Colour By Number Pixel Puzzle Picture 1: African Animal**

Images can be represented in many ways. In this colour-by-number puzzle you must recreate the picture using the lists of numbers that tell you how to colour each square or ‘pixel’.

Each line of instructions tells you how to colour the next two lines of the picture.

So, for example: 1 1 1 4 4 4

Would mean colour three pixels blue and then three pixels brown.

As you follow each instruction, cross off the steps you have done, so you don’t lose track.

**The Colour Lookup Table**

0 Black

1 Blue

2 Yellow

3 Green

4 Brown

**The Image Key**

1 1 1 1 1 1 1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

3 3 1 2 1 1 1

3 3 1 4 1 1 1

4 1 1 2 1 1 1

4 1 1 2 2 2 1

4 1 1 2 1 2 1

4 1 1 2 1 4 1

4 1 1 2 1 4 1

3 1 1 1 1 1 1

3 3 2 2 1 1 1

3 1 1 2 1 1 1

4 1 1 4 1 2 1

4 1 1 4 2 4 1

4 1 1 4 1 2 1

4 1 1 4 1 2 1

0 0 0 0 0 0 0

**Activity 3 What is the image?**

Each pixel code that tells us what colour it is; is made up of a mix of red, green and blue and there are over 16 million colour combinations! See for yourself! <http://www.colorpicker.com/>

So each pixel has information telling you where it goes and what colour it is.

Back to previous activities, how did you know how to put your class mates’ image together? A device needs this information too. What sort of information do you think a device needs to show an image properly?

|  |
| --- |
|  |

This is known as **METADATA.**
Q. How would you explain what Metadata is to someone who has not done the previous activity?

|  |
| --- |
|  |

**Activity 4: COMPRESSION**:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Let’s try another task: Colour the grid using the instructions

**Method 1: *LOSSLESS*:**Start from the top left cell. Move to the right then go to the next row, working left to right each time.

White, White, White, White, White, White, White, White, White, White, White, White, White, White, White, White, White, White, White, White, White, White, Yellow, Yellow, Yellow, Yellow, White, White, White, White, White, White, White, White, White, White, Yellow, Yellow, Yellow, Yellow, Yellow, Yellow, Yellow, Yellow, White, White, White, White, White, White, White, Yellow, Yellow, Yellow, Yellow, Yellow, Yellow, Yellow, Yellow, Yellow, Yellow, White, White, White....

Each pixel has coordinates and a colour code. You can see from above that this is going to take up a lot of space. Let’s finish the picture, If you followed the instructions correctly, we can carry on where we left off and start on row 5.

**Method 2: *LOSSY*:**Row 5; 3 white, 2 yellow, 2 black, 2 yellow, 2 black, 2 yellow, 3 white Row 6; 2 white, 3 yellow, 2 black, 2 yellow, 2 black, 3 yellow, 2 whiteRow 7; 2 white, 12 yellow, 2 whiteRow 8; 2 white, 12 yellow, 2 whiteRow 9; 2 white, 1 yellow, 1 black, 8 yellow, 1 black, 1 yellow, 2 whiteRow 10; 2 white, 2 yellow, 1 black, 6 yellow, 1 black, 2 yellow, 2 whiteRow 11; 2 white, 3 yellow, 1 black, 4 yellow, 1 black, 3 yellow, 2 white

Can you guess the image yet?
How did the second method “compress” the image?

|  |
| --- |
|  |

**Extension:** So is Lossy better? Why?

|  |
| --- |
|  |

### Activity 5 Exam Style Question

Alex is producing images and sound effects for a website. Part of a bitmap image is shown in Figure below



The letters represent a colour, as shown in Figure below



Using the example in the first Figure, explain how a bitmap image is stored on a computer.

|  |
| --- |
|  |

### Challenge Task:

Now repeat the image map (from today’s lesson), but this time using 4 colours.

|  |  |  |  |
| --- | --- | --- | --- |
| 00 White  | 01 Red | 10 Green  | 11 Blue |

00 00 01 00 00 00 00 00 01 00 00 10 11 10 01 00 00 00 01 10 10 01 10 10 10 00 01 00 00 00 10 00 00 00 00 10 01 01 01 00 01 10 00 00 00 00 10 10 01 00 01 10 10 00 00 00 11 11 10 10 00 10 10 00 00 00 00 11 11 10 10 10 10 10 00 00 00 00 00 11 10 10 10 10 10 00 00 00 00 00 10 10 10 10 10 10 00 00 00 00 10 10 00 10 10 10 10 10 00 00 11 11 11 11 11 11 11 10 10 00 00

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |

# **Part 11: Representation of Sound**

**Binary Representation of Sound**

What is sound? Find a picture to represent sound on the eardrum

|  |
| --- |
|  |

Amplitude is how loud the sound is



Frequency is how often the sound is measured. So, frequency is how many sound waves you can get in one second. This is called Hertz (Hz for short). Frequency is also called “pitch”



A frequency of 1 Hz means one wave cycle per second. A frequency of 10 Hz means ten wave cycles per second, where the cycles are much shorter and closer together.



Audacity uses 44.1Khz. Even this is not good enough for the music industry. They use 44Khz

**Use audacity to record the following 2 sentences**

1. In a quiet, squeaky voice (you are a mouse called “Amp”) ; say

**“Amplitude is how loud the sound is, Squeak!”**

1. In a loud deep voice, (you are an army sergeant called “Mega Hurts”) say;

**“Frequency is measured in megahertz, Hoo-hah!”**

Zoom in on the sound wave until you can see a clear line then play the sound for the mouse.

And Sergeant. What happens to the sound wave when;

 It is quiet or loud?

|  |
| --- |
|  |

Low or high pitched?

|  |
| --- |
|  |

Still a bit unsure on how frequency and amplitude affect sound? Go play on the link below……..

<http://www.classzone.com/books/ml_science_share/vis_sim/wslm05_pg18_graph/wslm05_pg18_graph.html>

**Sampling**:

So how does the computer know what the sound sounds like?

Join the dots…you need to draw your own dots though! Here they are:

On the X axis, put a dot at the following co ordinates. Have a go at drawing the sound wave.

|  |  |  |
| --- | --- | --- |
| 1 | 0 |  |
| 2 | 0.5 |  |
| 3 | 0 |  |
| 4 | -0.5 |  |
| 5 | -1.5 |  |
| 6 | -1 |  |
| 7 | 1 |  |
| 8 | 0.5 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | You have just done what the computer does to reassemble an audio file.To record a sound, the computer SAMPLES (aka, measures) the sound at regular points, (frequency). Then it converts the sample measurements into binary code. |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -0.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -1.5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| -2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

 Plenary discussion: How is sampling a sound the same as digitising an image?

When recording a sound file on a computer, the sound needs to be sampled.

1. Describe how sampling is used when storing sound.

|  |
| --- |
|  |

1. Explain the effect of the sampling interval on the size and quality of the sound file recorded.

|  |
| --- |
|  |

 Alex needs to create an audio recording of himself playing his guitar.

1. Explain how sampling is used to make the recording.

|  |
| --- |
|  |

1. State the effects of increasing the sample rate of the recording.

|  |
| --- |
|  |

**Binary - End of Unit Test**

* 1. Convert the denary number 106 into an 8 bit binary number. (2 marks)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **64** | **32** | **16** | **8** | **4** | **2** | **1** |
|  |  |  |  |  |  |  |

* 1. Convert the denary number 106 into hexadecimal. (2 marks)

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

1. Peter takes a high resolution picture with a digital camera. The picture is stored in a bitmap file.
	1. Describe how a picture is stored in a bitmap file. (2 marks)
	2. Peter wants to send the picture as an email attachment. State **two** methods for reducing the size of the picture file so that it is suitable for sending as an email. (2 marks)

1. Data stored in computers can be measured in bits, bytes and kilobytes.
	1. State what is meant by
		1. A bit (1 mark)
		2. A byte (1 mark)
	2. A file contains 5120 bytes. Calculate the size of the file in kilobytes. You must show your working. (2 marks)
	3. Calculate the denary value of the 8-bit binary number 10010111. You must show your working. (1 mark)

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

* 1. Add the following two 8-bit binary numbers ***and*** explain the result. You ***must*** show your working. (3 marks)
	
	2. Explain how ASCII is used to represent text in a computer system. (2 marks)
	3. State what is meant by the character set of a computer. (2 marks)
1. Describe the difference between *lossy* and *lossless* compression. (2 marks)

	1. Convert the hexadecimal number 6A to denary. You **must** show your working. (3 marks)

* 1. Convert the hexadecimal number 6A to binary. (3 marks)

* 1. Convert the binary number 00111101 to hexadecimal. (3 marks)

1. Bytes, Kilobytes and Megabytes are units used for the amount of data stored in a computer.
	1. State which of these units is most appropriate for the following items of data.
		1. A one page text document (1 mark)
		2. A ten minute movie clip (1 mark)
		3. A person’s surname (1 mark)
2. Peter takes a picture of himself and his friends to put on a social networking site. The picture is converted into pixels and stored as a bitmap file.
	1. Tick one box in each row to show whether or not each of the following items must be included in the bitmap file. (5 marks)
	
	2. What is meant by the resolution of the picture? (2 marks)

10. How does the resolution affect the size of the bitmap file? (2 marks)

**(3))**

|  |  |  |  |
| --- | --- | --- | --- |
| **Level** | **Assessment Criteria** | **Achieved (Student)** | **Achieved (Teacher)** |
| 2 | 1. I know different types of data: text, number.
2. I know programs use different data types
 |  |  |
| 3 | 1. I Know that computers use binary to understand what to do
2. I can explain the difference between data and information
 |  |  |
| 4 | 1. I can change a denary number to binary number and vice versa
2. I can convert positive denary whole numbers (0–255) into 2 digit hexadecimal numbers and vice
 |  |  |
| 5 | 1. I can explain that binary is at the base of all actions on a computer and all programs have to translate to binary to work
2. I understand the relationship between the number of bits per character in a character set
3. I understand the number of characters which can be represented (for example ASCII, extended ASCII and Unicode).
 |  |  |
| 6 | 1. I can complete binary addition
2. I can explain how data is stored in memory
3. I understand how numbers, images, sound and character sets use the same bit patterns.
4. I recognised how Metadata is included in the file
 |  |  |
| 7 | 1. I understand the relationship between data representation and data quality
2. I understand the need for compression and can explain the two types of compressions techniques (Lossy and Lossless)
 |  |  |

**SUCCESSES:** See ticked *(above)*

**TARGETS:** See unticked *(above)*

**ACTION:**