**P3**

**Mini – Guide**

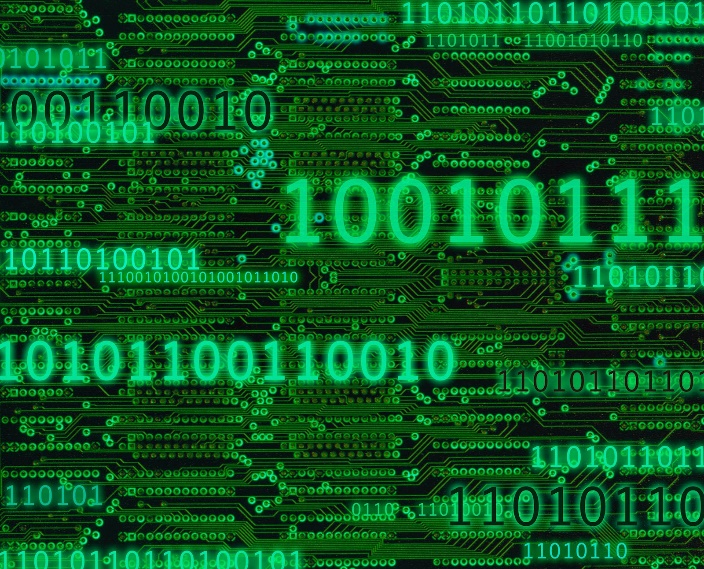
**Converting binary to denary**

**Binary**

Binary is two digits that the computer uses in order to identify and communicate with each other. The two digits are zero and one. Computers operate in binary, this means they store data and perform tasks with zeros and ones. This table below demonstrates how to convert numbers to binary, zero and one.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 27= | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| Decimal | **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |
| 72 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 65 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 85 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

**Denary**

Denary are integers to what the numbers are converted to. For example, if 01010111 is binary, denary are numbers that it is used to represent.

CONVERT THESE BINARY NUMBERS INTO DENARY:

The first step you do is write down is the place values: 128 64 32 16 8 4 2 1. Each of these are represented for each code. This is an 8 bit number. The next step you do is place the numbers into the place values.

128 64 32 16 8 4 2 1

0 1 0 1 0 1 1 1

Each of the ‘1’ that is placed underneath, it will be added together. Therefore, 64 + 16 + 4 + 2 + 1 = 87

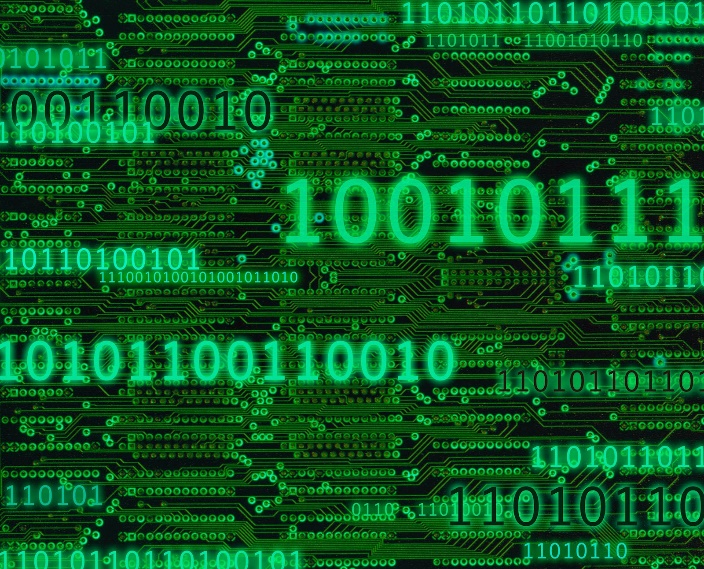
**CONVERSION**

The main way to work it out is to convert it by using the numbers– 128, 64, 32, 16, 8, 4, 2, and 1 - to get what you want. For example, if you want 13 as a denary, you should add the ones that make 12. Therefore, 8+4. They are represented as 1 and the others are 0s. It should look like this 01100. I only used 16, 8, 4, 2, and 1 for it, as I did not need the others. However if you do want to work out 256, you would need to double 128 and carry on as normal. The ones you added to make the number you put a one and the ones you did not you put a zero.

**Converting Denary to Binary**

**Mini-Guide**

**Binary**

Binary is two digits that the computer uses in order to identify and communicate with each other. The two digits are zero and one. Computers operate in binary, this means they store data and perform tasks with zeros and ones. This table below demonstrates how to convert numbers to binary, zero and one.

**Denary**

Denary are integers to what the numbers are converted to. For example, if 01010111 is binary, denary are numbers that it is used to represent.

CONVERT THESE DENARY NUMBERS INTO BINARY:

The first step you do is write down is the place values: 128 64 32 16 8 4 2 1. Each of these are represented for each code. This is an 8 bit number. The next step you do is place the numbers into the place values.

128 64 32 16 8 4 2 1

0 1 0 1 0 1 1 1

Each of the ‘1’ that is placed underneath, it will be added together. So, you put in ‘1’ below 87. So 64, 16, 4, 2 and 1 fits in and makes 87. So, it would be 87.

The first way to convert denary to binary is to use the place values and submit in to the place value. Every ‘1’ that there is, you have to add it together. For example, if I were to convert 87, you have to add it together so it is correct. You insert it within 87. So 64 + 16 + 4 + 2 + 1 = 87. This all works.

128 64 32 16 8 4 2 1

0 1 0 1 0 1 1 1