Template Week 1 – Bits & Bytes

Student number: 583168

Assignment 1.1: Bits & Bytes intro

What are Bits & Bytes?

A bit (binary digit) is the smallest unit of data in a computer, representing either a 0 or 1. It is the basic building block of binary code, which is the language computers use. Each bit can show a choice such as On or Off or True or False. Bits are the foundation of all digital data.

A byte consists of 8 bits and is the standard unit used to represent a character of data, like a letter or symbol. A byte can show 256 different values (0 to 255 in decimal) and is commonly used to measure file sizes and memory capacity. Data is often measured using bytes, kilobytes (KB), megabytes (MB), gigabytes (GB), and terabytes (TB).

What is a nibble?

A nibble is a small unit of data equal to 4 bits, which is half of a byte. Since a byte is made up of 8 bits, a nibble represents half of the information a byte can hold. It can show values from 0 to 15 in decimal and is often used to represent a single hexadecimal digit.

What relationship does a nibble have with a hexadecimal value?

A nibble and a hexadecimal value are connected because one nibble (4 bits) can represent exactly one hex digit. A nibble can show numbers from 0 to 15, which is the same range as a single hexadecimal digit (0 to F). So basically, each hex digit is stored using one nibble in binary.

Why is it wise to display binary data as hexadecimal values?

It's wise to display binary data as hexadecimal because hex is shorter and easier to read. Binary numbers can get very long and confusing, while hexadecimal represents the same value using fewer digits. This makes it easier to understand, write, and work with data without losing any information.

What kind of relationship does a byte have with a hexadecimal value?

A byte has a close relationship with a hexadecimal value because one byte is made up of 8 bits, and 8 bits can be split into two nibbles. Since each nibble represents one hex digit, a single byte can be written as two hexadecimal digits. This makes hex a clear and simple way to show the value of a byte.

An IPv4 subnet is 32-bit, show with a calculation why this is the case.

An IPv4 subnet is 32-bit because it matches the size of the IPv4 address it is masking. An IPv4 address is made up of four numbers (like 192.168.1.5), and each number uses 8 bits, so the total is 8 + 8 + 8 + 8 = 32 bits. Since a subnet mask works directly on these addresses, it must also be 32 bits, which is why subnet masks like 255.0.0.0, 255.255.0.0, or 255.255.255.0 each contain four 8-bit values, adding up to the same 32-bit length.

Assignment 1.2: Your favourite color

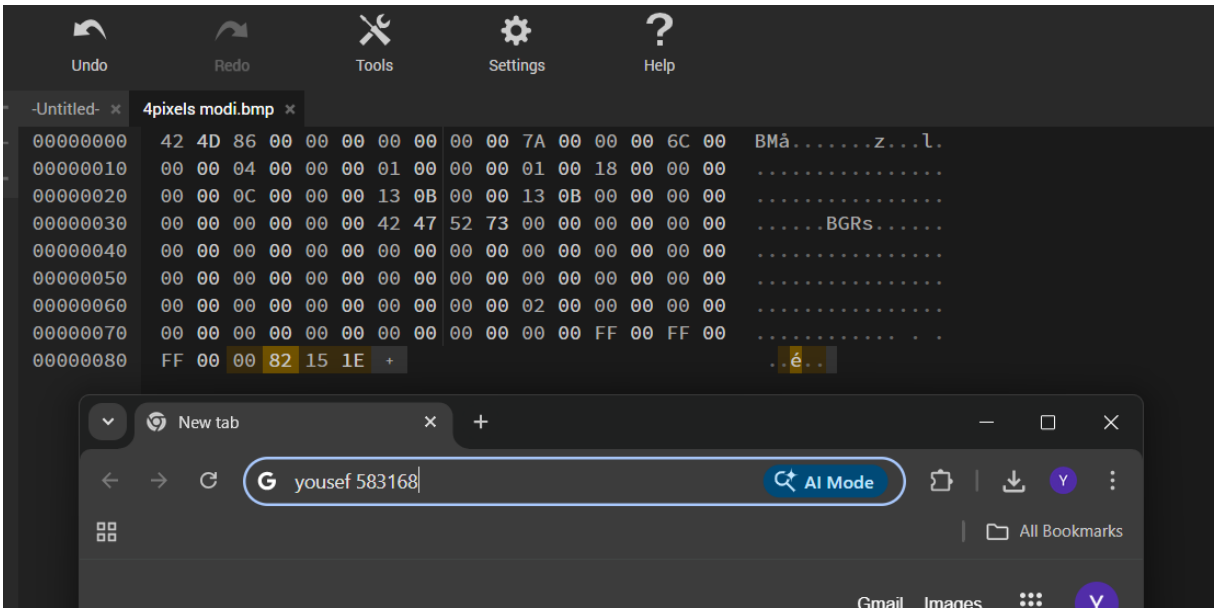
Hexadecimal color code:

#1e1582

Assignment 1.3: Manipulating binary data

Color	Color code hexadecimaal (RGB)	Big Endian	Little Endian
RED	FF 00 00	FF 00 00	00 00 FF
GREEN	00 FF 00	00 FF 00	00 FF 00
BLUE	00 00 FF	00 00 FF	FF 00 00
WHITE	FF FF FF	FF FF FF	FF FF FF
Favourite (previous assignment)	1E 15 82	1E 15 82	82 15 1E

Screenshot modified BMP file in hex editor:



Assignment 1.4: Student number to HEX and Binary

Convert your student number to a hexadecimal number and a binary number.

Student number 583168

HEXADECIMAL

583168 / 16 = 36448 remainder 0

36448 / 16 = 2278 remainder 0

2278 / 16 = 142 remainder 6

142 / 16 = 8 remainder 14 (E)

8 / 16 = 0 remainder 8

Answer : 8E 60 0

BINARY

583168 / 2 = 291584 remainder 0

291584 / 2 = 145792 remainder 0

145792 / 2 = 72896 remainder 0

72896 / 2 = 36448 remainder 0

36448 / 2 = 18224 remainder 0

18224 / 2 = 9112 remainder 0

9112 / 2 = 4556 remainder 0

4556 / 2 = 2278 remainder 0

2278 / 2 = 1139 remainder 0

1139 / 2 = 569 remainder 1

569 / 2 = 284 remainder 1

284 / 2 = 142 remainder 0

142 / 2 = 71 remainder 0

71 / 2 = 35 remainder 1

35 / 2 = 17 remainder 1

17 / 2 = 8 remainder 1

8 / 2 = 4 remainder 0

4 / 2 = 2 remainder 0

2 / 2 = 1 remainder 0

1 / 2 = 0 remainder 1

Answer: 1000111001100000000

Explain in detail that the calculation is correct. Use the PowerPoint slides of week 1.

Ready? Save this file and export it as a pdf file with the name: **week1.pdf**