What’s a computer?
Memory

• **bit** = (binary digit) a *smallest* memory device
  – A **bit** is in fact a **switch** that can remember **0** or **1**
  – (The digits **0** and **1** are digits used in the **binary number system**)

• **Byte** = 8 bits
  – A **byte** is in fact one row of the **RAM memory**

• **KByte** = kilo byte = **1024** (= **2**\(^{10}\)) **bytes** (approximately 1,000 bytes)

• **MByte** = mega byte = **1048576** (= **2**\(^{20}\)) **bytes** (approximately 1,000,000 bytes)

• **GByte** = giga byte = **1073741824** (= **2**\(^{30}\)) **bytes** (approximately 1,000,000,000 bytes)

• **TByte** = tera byte
Encoding

• **Information** can be stored as *numbers* by using an **encoding method**.

• An **encoding method** is simply an **agreement** on a **representation** of some facts by specific **numbers**.
• ASCII encoding
  – Most files use the ASCII encoding method to represent the characters
  – For details on the ASCII encoding, see: AsciiT able.com

• Unicode encoding
  – However, files containing non-English characters (e.g., Greek letters α, β, etc, Chinese characters, and so on) use the Unicode encoding method.
  – For details on the Unicode encoding, see: Unicode.org
Binary numbers

• The **binary number system** uses 2 digits (0 and 1) to encode a number.
• An $n$ digit cell can encode $2^n$ different numbers.

- A **byte** has 8 bits and therefore, it can store:
  - $2^8 = 256$ different patterns
  - (These 256 patterns are: 00000000, 00000001, 00000010, 00000011, ..., 11111111)

- Each pattern can be encoded exactly one number:
  - 00000000 = 0
  - 00000001 = 1
  - 00000010 = 2
  - 00000011 = 3
  - ...
  - 11111111 = 255

Therefore, one byte can store one of **256 possible values**

(You can store the number 34 into a byte, but you cannot store the number 556, the value is out of range)
• Use consecutive bytes to encode more numbers.
• Binary and decimal numbers (non-negative)

\[ \begin{align*}
1 \times 2^0 &= 1 \times 1 = 1 \\
0 \times 2^1 &= 0 \times 2 = 0 \\
0 \times 2^2 &= 0 \times 4 = 0 \\
1 \times 2^3 &= 1 \times 8 = 8 \\
1 \times 2^4 &= 1 \times 16 = 16 \\
0 \times 2^5 &= 0 \times 32 = 0 \\
1 \times 2^6 &= 1 \times 64 = 64 \\
1 \times 2^7 &= 1 \times 128 = 128 \\
1 + 8 + 16 + 64 + 128 &= 217
\end{align*} \]

\[ \begin{array}{c}
2) 156 \\
2) 78 \\
2) 39 \\
2) 19 \\
2) 9 \\
2) 4 \\
2) 2 \\
2) 1 \\
\hline
\text{Remainder:} & 0 \\
& 1 \\
& 1 \\
& 1 \\
& 0 \\
& 0 \\
& 1 \\
\end{array} \]

\[ 156_{10} = 10011100_2 \]
Compiler

Algorithm written in prog. language

```
main()
{
    printf("Hello\n");
}
```

file

Then you translate the algorithm into machine instructions with a compiler

```
10101011
01011010
11100011
```

application

Machine instructions

```
01101011
10011010
10111001
```

application

First you write the algorithm in a prog. language with an editor (gedit)

Finally, you execute the algorithm in machine instructions
Computer Algorithm

• Algorithm is a step-by-step procedure for solving a problem or accomplishing some task, especially by means of a computer.

• Computer Algorithm is an algorithm that can be executed by a computer.