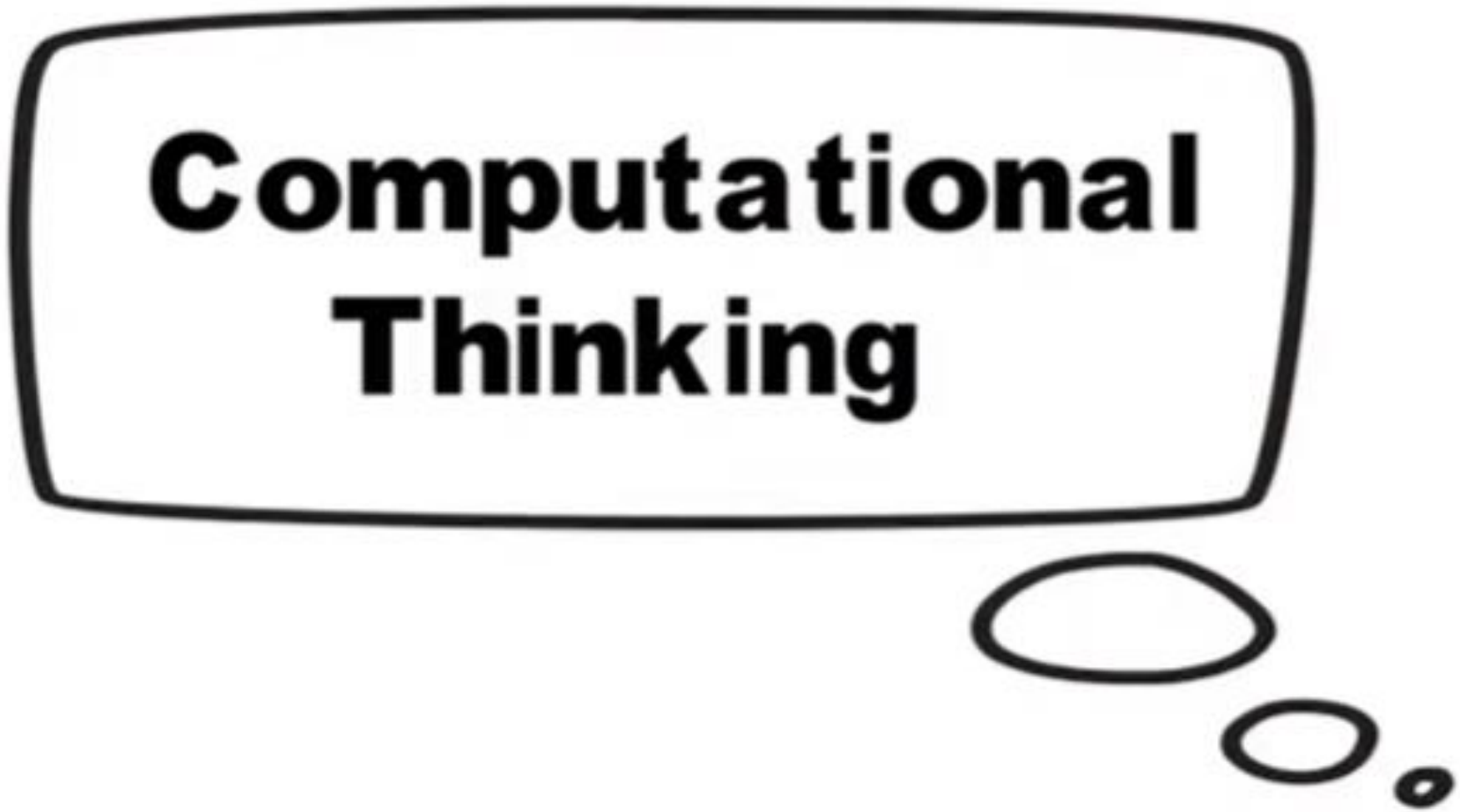
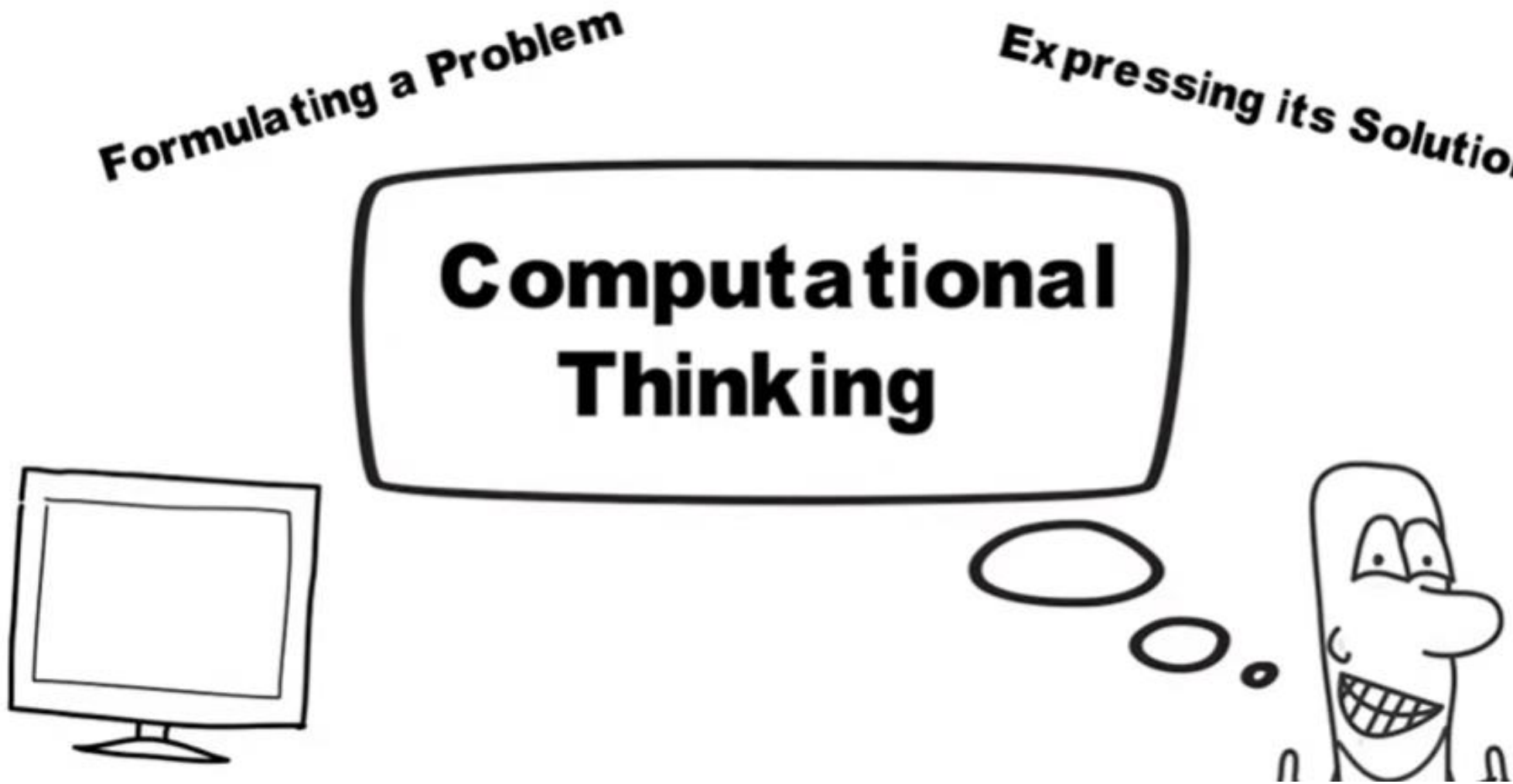

Computational
Thinking


Formulating a Problem
Expressing its Solution
Computational Thinking

The four steps of Computational Thinking are:

1. Decomposition
   - ✓ Which involves breaking down a complex problem into smaller, manageable parts.

2. Pattern Recognition
   - ✓ Where similarities and trends are identified among problems and solutions

3. Abstraction
   - ✓ Which focuses on essential details while ignoring irrelevant information and

4. Algorithm Design
   - ✓ Creating a step-by-step set of instructions to solve the problem.



Breaking a complex problem down into smaller, more manageable parts.

Find the sum of all numbers between 1 and 200

$1 + 2 = 3$

$3 + 4 = 7$

$5 + 6 = 11$

$7 + 8 = 15$

Find the sum of all numbers between 1 and 200

$200 + 1 = 201$

$199 + 2 = 201$

$198 + 3 = 201$

etc....



Identifying patterns or trends within a problem.

Find the sum of all numbers between 1 and 200

$200 + 1 = 201$

$199 + 2 = 201$

$198 + 3 = 201$

etc....

Find the sum of all numbers between 1 and 200

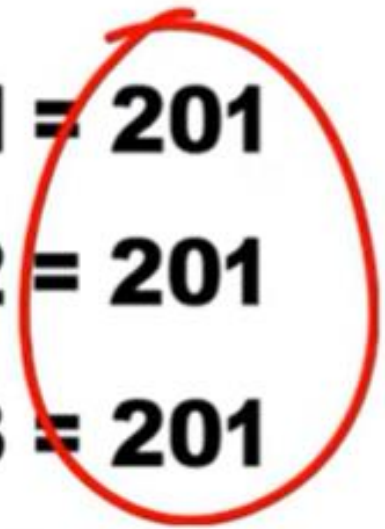
200 + 1 = 201

199 + 2 = 201

198 + 3 = 201

etc....

Find the sum of all numbers between 1 and 200



200 + 1 = 201

199 + 2 = 201

198 + 3 = 201

etc....

Find the sum of all numbers between 1 and 200

200 + 1 = 201

199 + 2 = 201

198 + 3 = 201

etc....

200 / 2 = 100 Pairs



Identify specific similarities and differences among similar problems to work toward a solution.

Find the sum of all numbers between 1 and 200

200 + 1 = 201

199 + 2 = 201

198 + 3 = 201

etc....

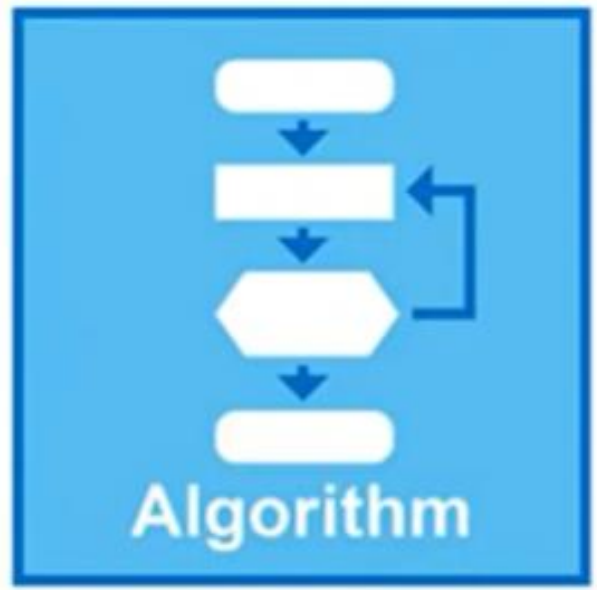
200 / 2 = 100 Pairs

Find the sum of all numbers between 1 and 200

200 + 1 = 201 200 / 2 = 100 Pairs

Find the sum of all numbers between 1 and 200

$$\underbrace{(200+1)}_{\text{sum of each pair}} * \underbrace{(200/2)}_{\text{number of pairs}} = 20{,}100$$



Develop step-by-step instructions for solving the problem.

Find the sum of all numbers between 1 and 200

$$(200 + 1) * (200 / 2) = 20{,}100$$

STEP 1: $200 + 1 = 201$

Find the sum of all numbers between 1 and 200

(200 + 1) * (200 / 2) = 20,100

STEP 1: 200 + 1 = 201

STEP 2: 200 / 2 = 100

Find the sum of all numbers between 1 and 200

$$(200 + 1) * (200 / 2) = 20{,}100$$

STEP 1: 200 + 1 = 201

STEP 2: 200 / 2 = 100

STEP 3: 201 * 100 = 20,100

Find the sum of all numbers between 1 and 1,000

(+ 1) * (/ 2) =

STEP 1:

STEP 2:

STEP 3:

Find the sum of all numbers between 1 and 1,000

(1,000 + 1) * (/ 2) =

STEP 1: 1,000 + 1 = 1,001

STEP 2:

STEP 3:

Find the sum of all numbers between 1 and 1,000

$$(1{,}000 + 1) * (1{,}000 / 2) =$$

STEP 1: 1,000 + 1 = 1,001

STEP 2: 1,000 / 2 = 500

STEP 3:

Find the sum of all numbers between 1 and 1,000

$$(1{,}000 + 1) * (1{,}000 / 2) =$$

STEP 1: 1,000 + 1 = 1,001

STEP 2: 1,000 / 2 = 500

STEP 3: 1,001 * 500 = 500,500

Find the sum of all numbers between 1 and 1,000

$$(1{,}000 + 1) * (1{,}000 / 2) = 500{,}500$$

STEP 1: 1,000 + 1 = 1,001

STEP 2: 1,000 / 2 = 500

STEP 3: 1,001 * 500 = 500,500

Calculation of the Sum

The sum of all natural numbers from 1 to n can be calculated using the formula for the sum of an arithmetic series or the formula for triangular numbers.

Step-by-step Solution

Identify the parameters:

The numbers are from 1 to 200. Therefore, the first term a is 1, the last term l is 200, and the number of terms n is 200.

Apply the sum formula:

The sum S of an arithmetic series can be found using the formula $S = \frac{n}{2}(a + l)$.

Substitute the values:

Substitute $n = 200$, $a = 1$, and $l = 200$ into the formula: $S = \frac{200}{2}(1 + 200)$.

Perform the calculation:

First, calculate $\frac{200}{2} = 100$.

Next, calculate $1 + 200 = 201$.

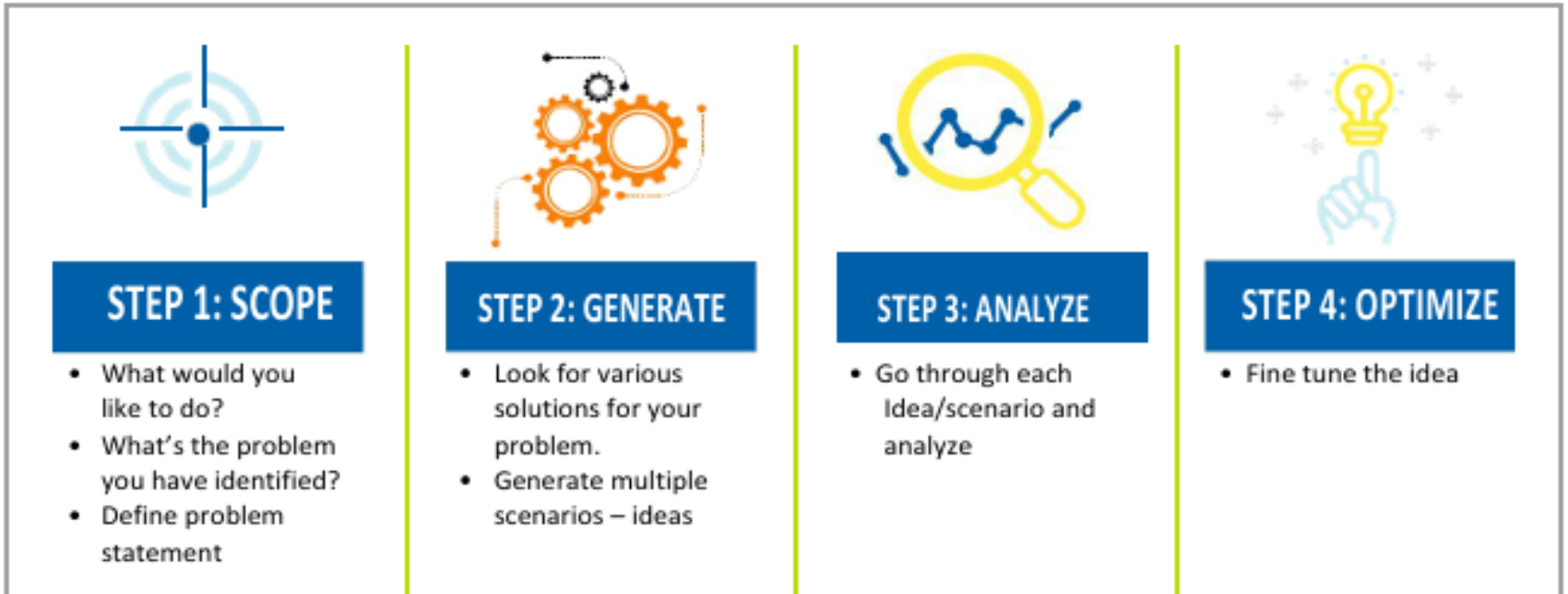
Finally, multiply these results: $S = 100 \times 201 = 20100$.

1. **Decomposition** - breaking down a complex problem or system into smaller, more manageable parts (e.g. where to go, how to complete the level)

2. **Abstraction** - focusing on the important information only, ignoring irrelevant detail (e.g. weather, location of exit)

3. **Pattern recognition** – looking for similarities among and within problems (e.g. knowledge of previous similar problems used)

4. **Algorithms** - developing a step-by-step solution to the problem, or the rules to follow to solve the problem (e.g. to work out a step-by-step plan of action)



Common Number Systems

System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, … 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, … 7	No	No
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

The decimal system (base 10)

- The word decimal is derived from the Latin root decem (ten). In this system the base b = 10 and we use ten symbols.

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

- The symbols in this system are often referred to as decimal digits or just digits.

The binary system (base 2)

- The word binary is derived from the Latin root bini(or two by two).

- In this system the base b = 2 and we use only two symbols.

$$S = \{0, 1\}$$

- The symbols in this system are often referred to as binary digits or bits(binary digit).

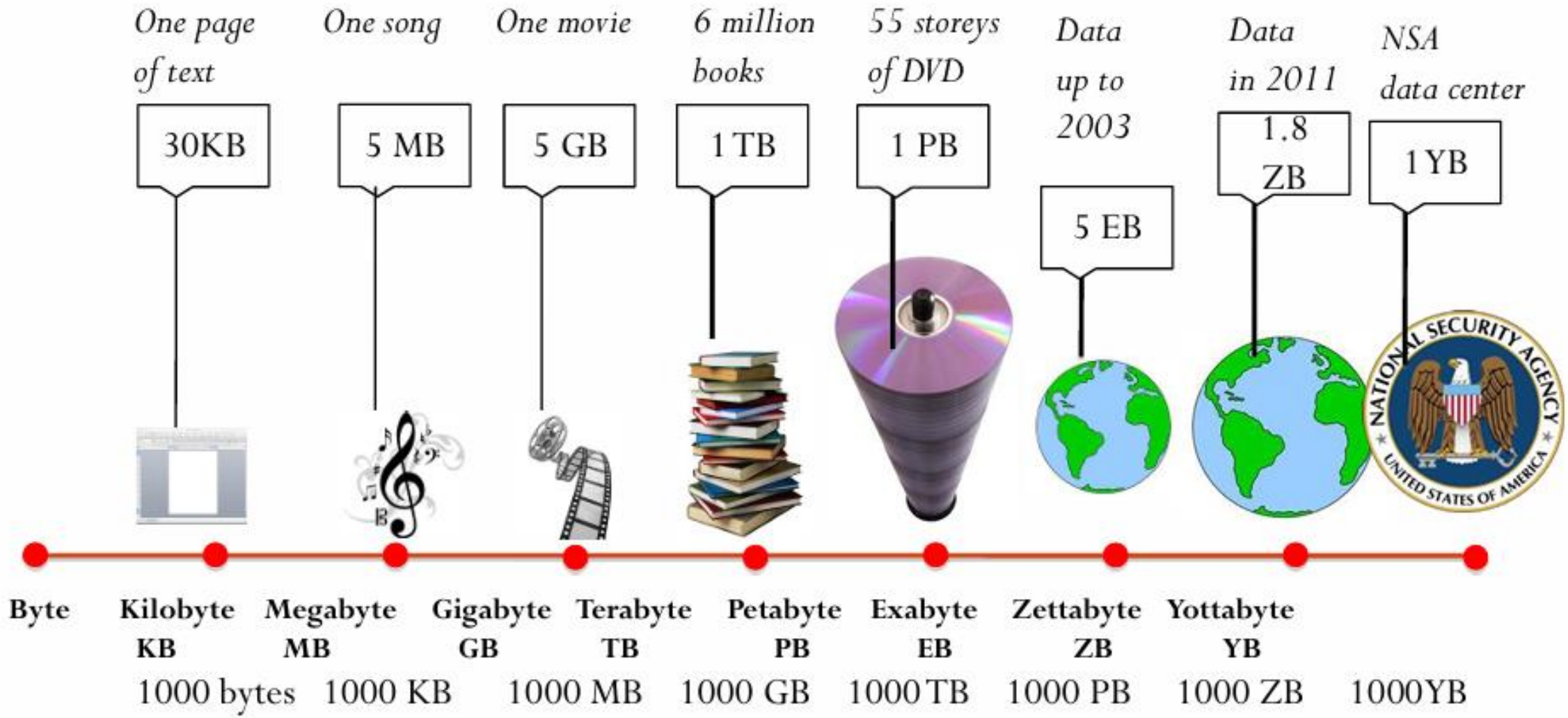
Bits and binary

- All computer data is represented using binary, a number system that uses 0s and 1s.
- Binary digits can be grouped together into bytes.
- Computers use **binary** - the digits 0 and 1 - to store data.
- A binary digit, or **bit**, is the smallest unit of data in computing.
- It is represented by a 0 or a 1.
- **Binary numbers are made up of binary digits (bits)**, eg the binary number **1001**.

Byte to Terabyte

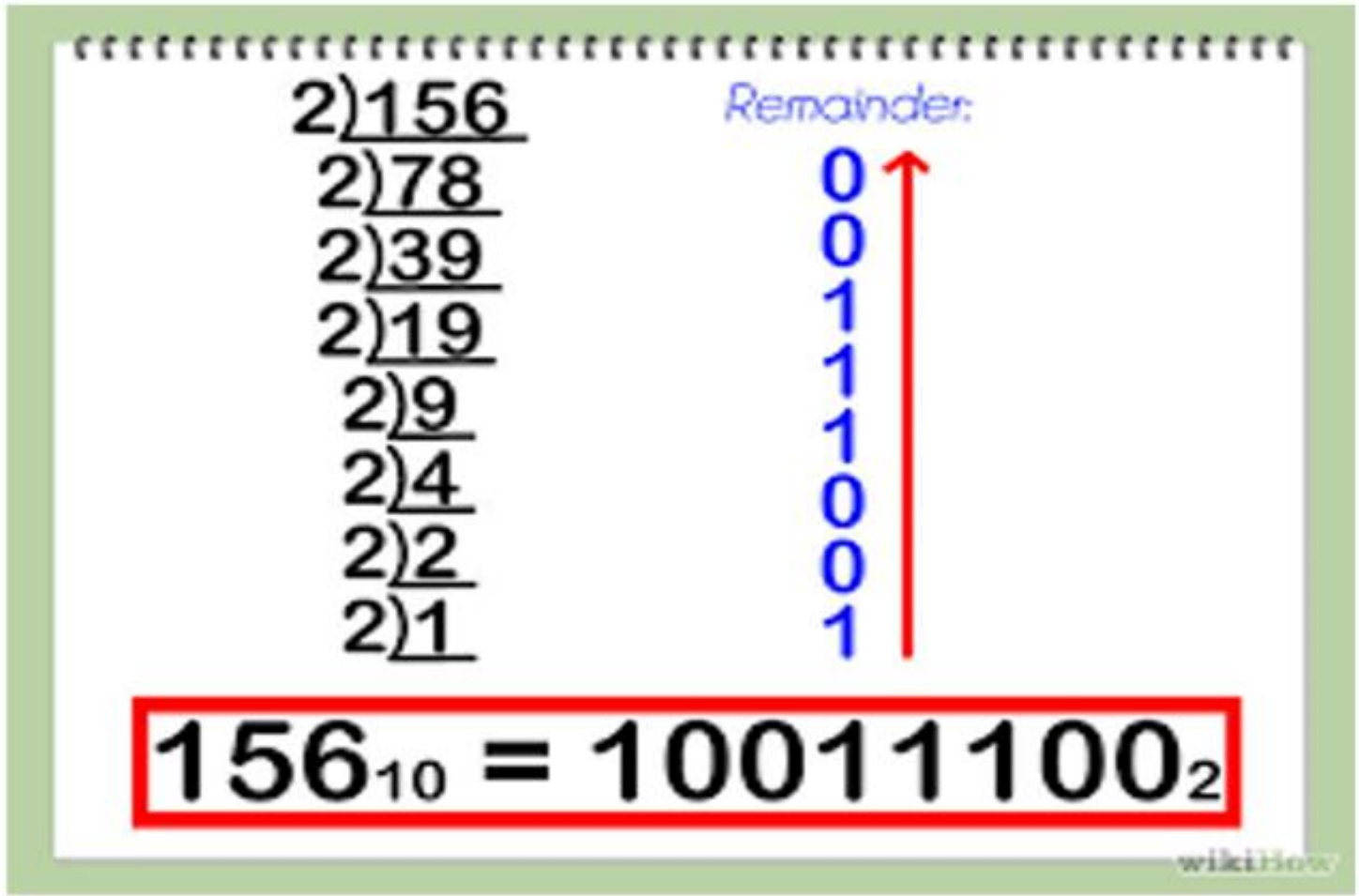
- Bits can be grouped together to make them easier to work with. **A group of 8 bits is called a byte**.
- Other groupings include:
- **Nibble** - 4 bits (half a byte)
- **Byte** - 8 bits
- **Kilobyte** (KB) - 1024 bytes (or 1024 x 8 bits)
- **Megabyte** (MB) - 1024 kilobytes (or 1048576 bytes)
- **Gigabyte** (GB) - 1024 megabytes
- **Terabyte** (TB) - 1024 gigabytes
- Most computers can process millions of bits every second. A hard drive's storage capacity is measured in gigabytes or terabytes. RAM is often measured in megabytes or gigabytes.

Big Data: Volume



Decimal to Binary Conversion

- The number $(156)_{10}$ in decimal is the same as $(10011100)_2$ in binary.
- Let us see how the conversion is done!



Decimal to Binary Conversion Cntd..

- If the decimal number to be converted is $(156.6875)_{10}$, let us see how the fractional part can be converted to binary!

$0.6875 \times 2 = \mathbf{1}.3750$ 1

$0.375 \times 2 = \mathbf{0}.7500$ 0

$0.7500 \times 2 = \mathbf{1}.500$ 1

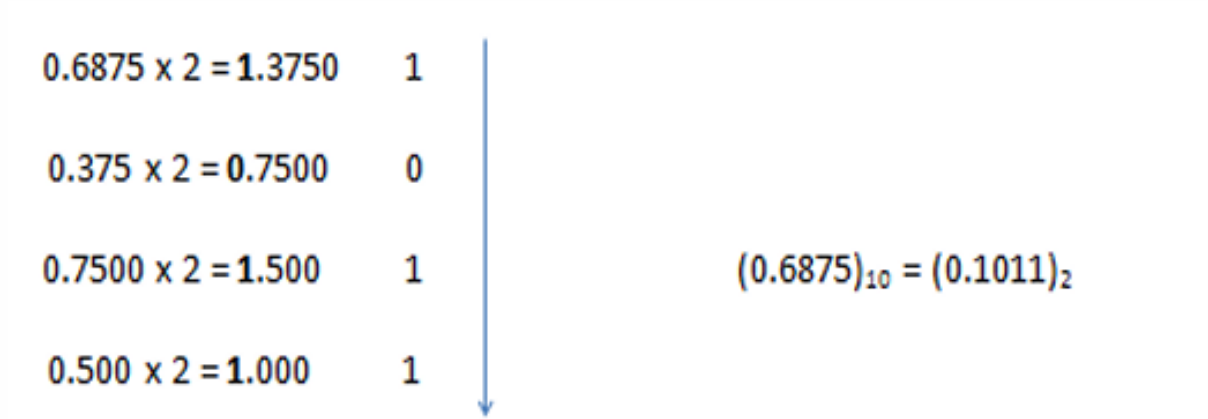
$0.500 \times 2 = \mathbf{1}.000$ 1



$(0.6875)_{10} = (0.1011)_2$

Decimal to Binary Conversion Cntd..

- If the decimal number to be converted is $(156.6875)_{10}$, let us see how the fractional part can be converted to binary!



- The binary of $(156)_{10}$ is $(10011100)_2$ and the binary of $(0.6875)_{10}$ is $(0.1011)_2$

  So the complete answer is $(156.6875)_{10} = (10011100.1011)_2$

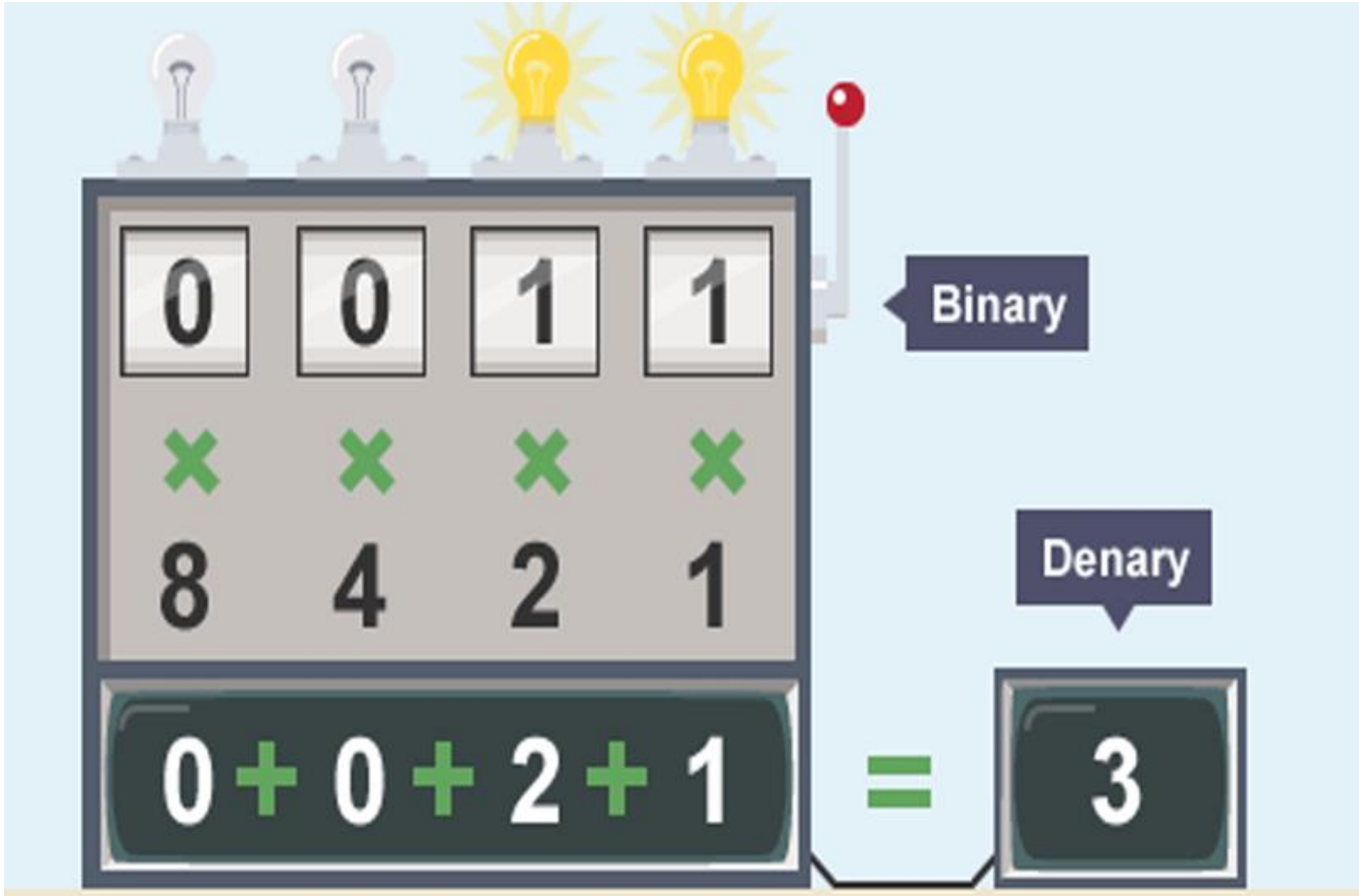
Binary to Decimal Conversion

- The number $(11001)_2$ in binary is the same as 25 in decimal and is represented as…

$$\begin{array}{cccccccccc} & 2^4 & & 2^3 & & 2^2 & & 2^1 & & 2^0 \\ & 1 & & 1 & & 0 & & 0 & & 1 \\ N = & 1\times 2^4 & + & 1\times 2^3 & + & 0\times 2^2 & + & 0\times 2^1 & + & 1\times 2^0 \end{array}$$

- The equivalent decimal number is N = 16 + 8 + 0 + 0 + 1 = 25.
- The number $(101.11)_2$ in binary is equal to the number 5.75 in decimal .

$$\begin{array}{cccccccccccc} & 2^2 & & 2^1 & & 2^0 & & 2^{-1} & & 2^{-2} \\ & 1 & & 0 & & 1 & \bullet & 1 & & 1 \\ R = & 1\times 2^2 & + & 0\times 2^1 & + & 1\times 2^0 & + & 1\times 2^{-1} & + & 1\times 2^{-2} \end{array}$$


0 0 1 1
Binary
× × × ×
8 4 2 1
Denary
0 + 0 + 2 + 1
=
3


1100
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011



✓*Can you perform the indicated conversion?*

Decimal to Binary

a) $13_{10} = ?$

b) $22_{10} = ?$

c) $43_{10} = ?$

d) $158_{10} = ?$

Binary to Decimal

a) $0110_{2} = ?$

b) $11010_{2} = ?$

c) $0110101_{2} = ?$

d) $11010011_{2} = ?$

Decimal → Binary .....Answers

a) $13_{10} = ?$ 1101_2

b) $22_{10} = ?$ 10110_2

c) $43_{10} = ?$ 101011_2

d) $158_{10} = ?$ 10011110_2

Binary → Decimal .....Answers

a) $0110_2 = ?$ 6_{10}

b) $11010_2 = ?$ 26_{10}

c) $0110101_2 = ?$ 53_{10}

d) $11010011_2 = ?$ 211_{10}

The hexadecimal system (base 16)

- The word hexadecimal is derived from the Greek root hex (six) and the Latin root decem (ten). In this system the base b = 16 and we use sixteen symbols to represent a number.
- The set of symbols are:

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$$



Note that the symbols A, B, C, D, E, F are equivalent to 10, 11, 12, 13, 14, and 15 respectively. The symbols in this system are often referred to as hexadecimal digits.

The hexadecimal system (base 16) *contd*---

- The number $(2AE)_{16}$ in hexadecimal is equivalent to 686 in decimal is represented as…

	16^2		16^1		16^0
	2		A		E
N =	2×16^2	+	10×16^1	+	14×16^0

- The equivalent decimal number is N = 512 + 160 + 14 = 686.

Using hexadecimal

- Hex codes are used in many areas of computing to simplify binary codes.
- It is important to note that computers do not use hexadecimal - it is used by humans to shorten binary to a more easily understandable form.
- Hexadecimal is translated into binary for computer use. Some examples of where hex is used include:
  - colour references
  - assembly language programs
  - error messages

Hex colour model

- Hex can be used to represent **colours** on web pages and image-editing programs
- using the format **#RRGGBB** (RR = reds, GG = greens, BB = blues).
- The # symbol indicates that the number has been written in hex format.
- eg #FF6600.

The **Hex color model** uses two hex digits for each colour

Eg: #FF 66 00



1	1	1	1	=	F	R
1	1	1	1	=	F	
0	1	1	0	=	6	G
0	1	1	0	=	6	
0	0	0	0	=	0	B
0	0	0	0	=	0	

As one hex digit represents 4 bits

Two hex digits together make 8 bits (1 byte).

- The values for each colour run between 00 and FF.

- In binary,
  - 00 is 0000 0000
  - FF is 1111 1111

- That provides 2^8 = 256 possible values for each of the three colours.

- That gives a total **spectrum** of 256 reds x 256 greens x 256 blues - which is over 16 million colours in total.

Table : summary of four positional number systems

System	Base	Symbols	Examples
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	2345.56
Binary	2	0, 1	$(1001.11)_2$
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$(A2C.A1)_{16}$

Table : Four positional number systems

Comparison of numbers in the four systems

Decimal	*Binary*	*Octal*	*Hexadecimal*
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Decimal to any base

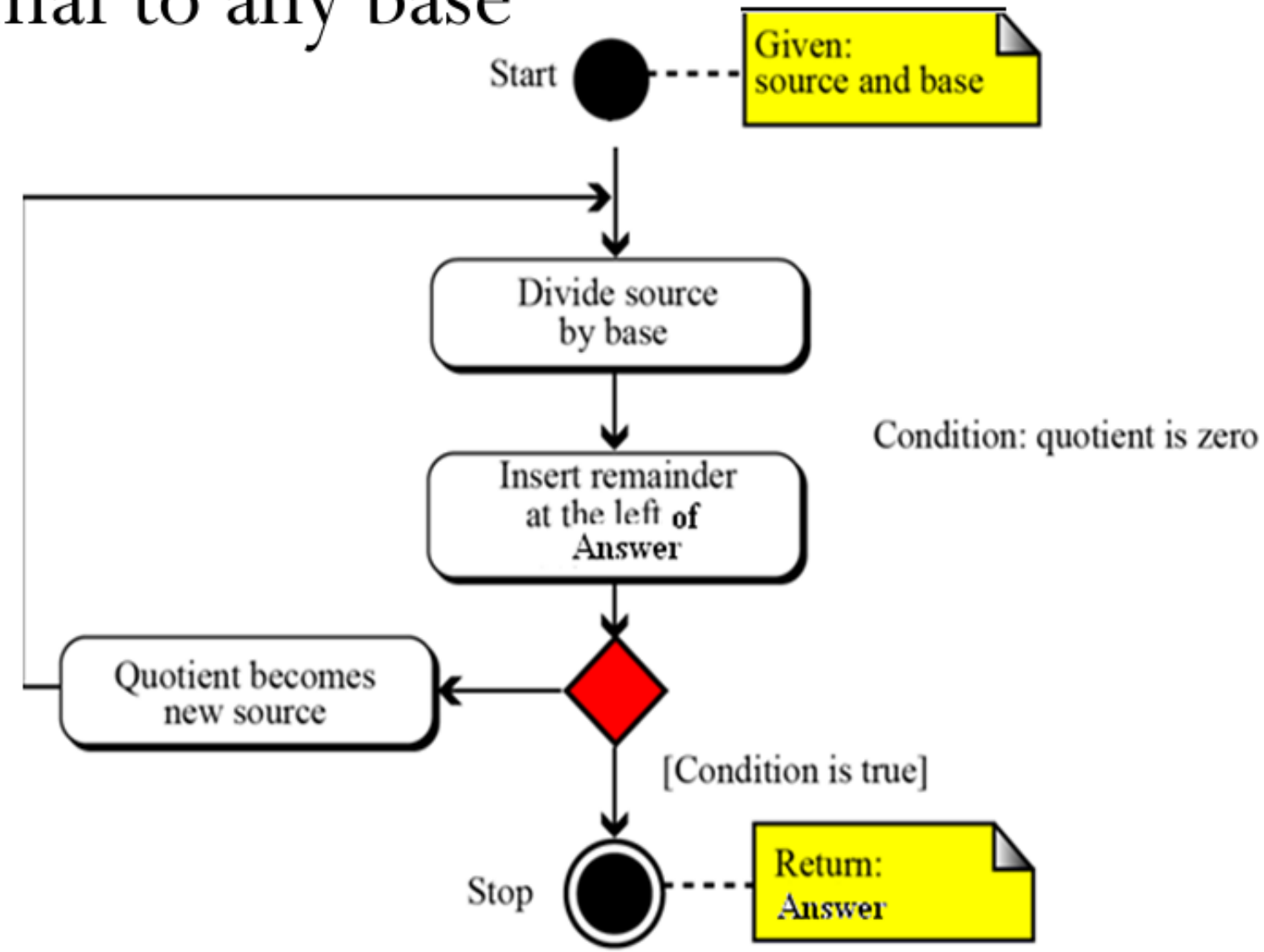


Figure : Converting decimal to other bases (integral part)

✓ *Can you identify the forms of the data in use?*













Answer.....

The different forms of data are text, image,audio,video....



Image



Text



Image



Audio

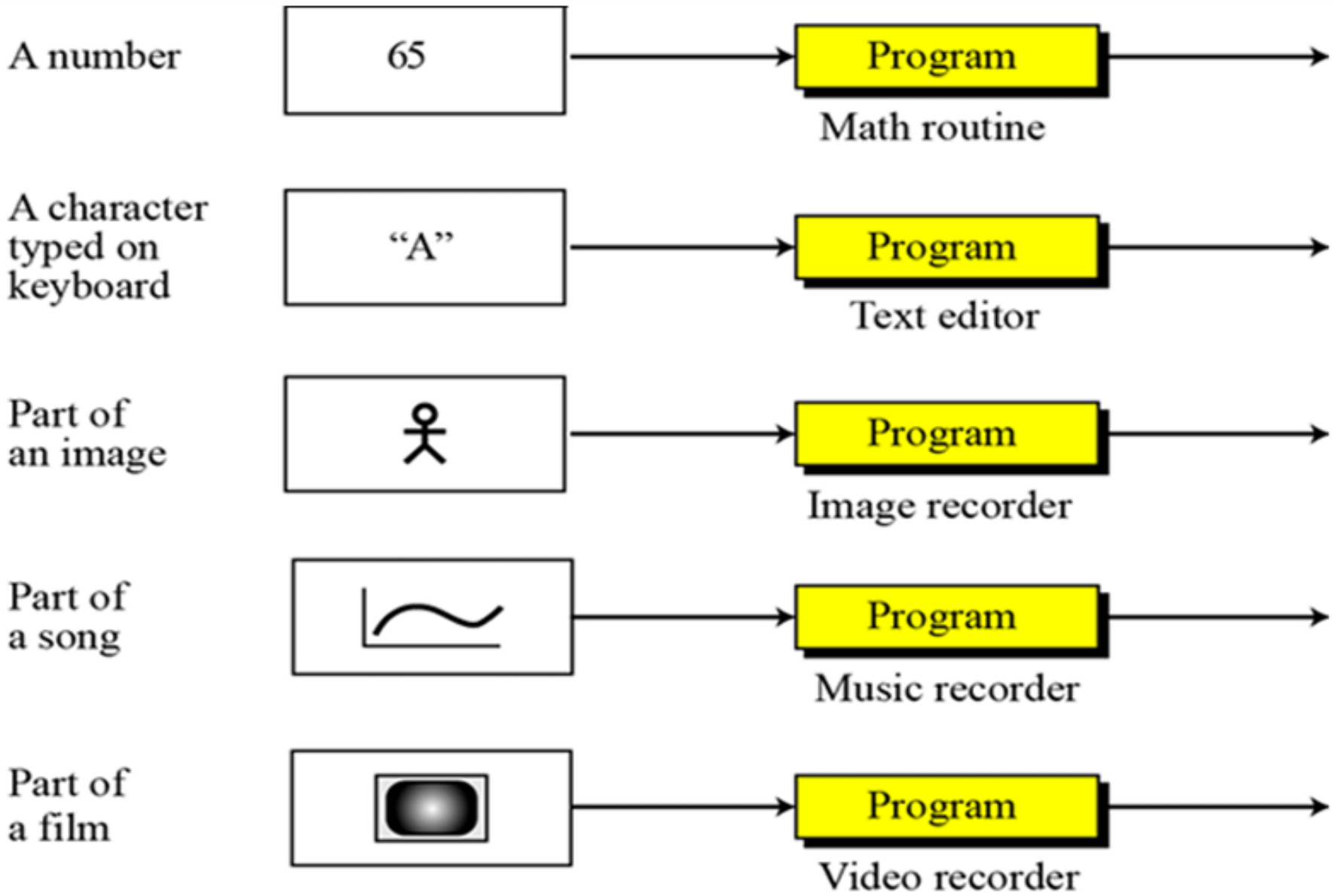


Audio

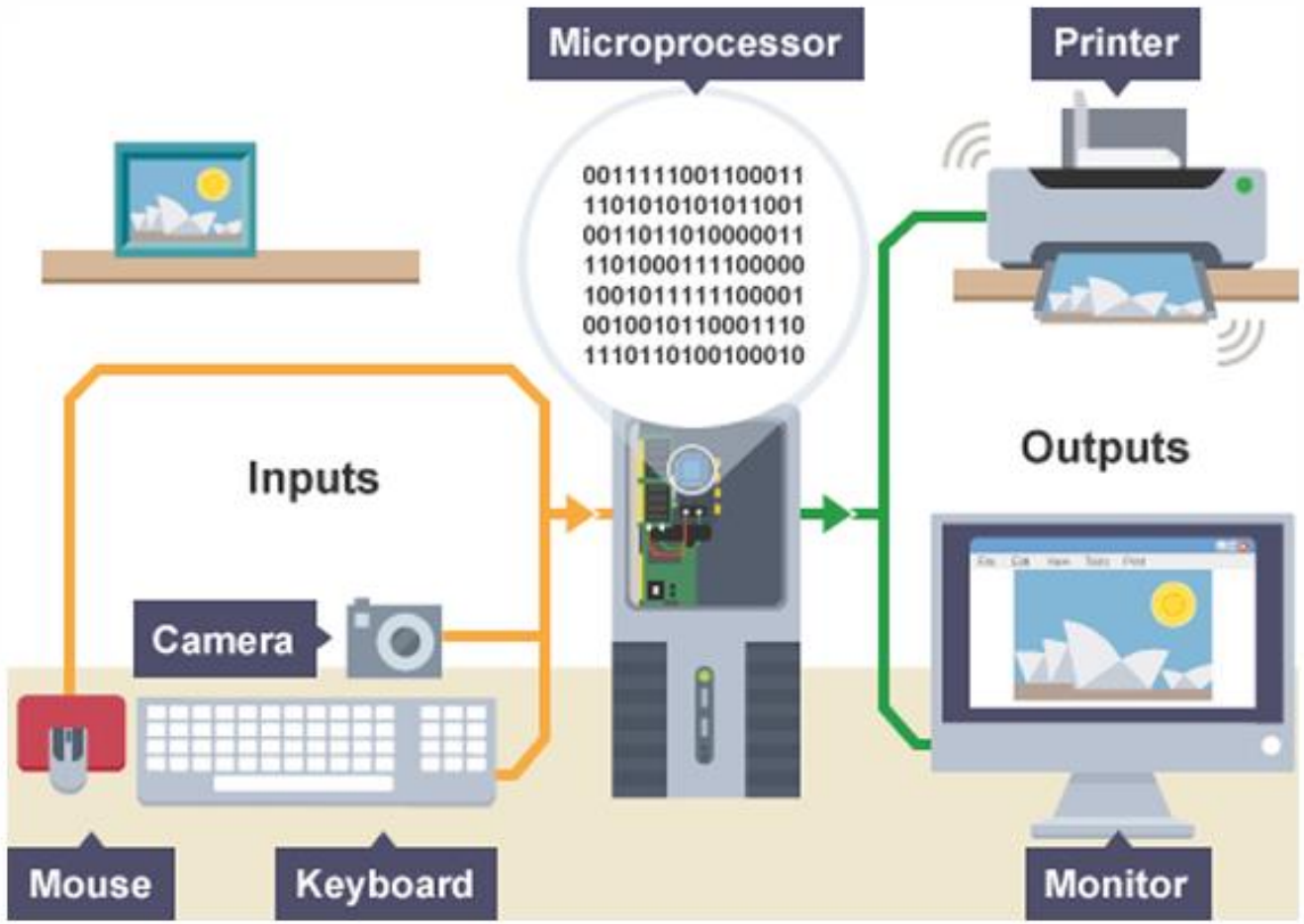


Video

Different Forms of Data



All software, music, documents, and any other information that is processed by a computer, is also stored using binary.



Data in any form (text, audio, video) is represented in binary form only in computers.

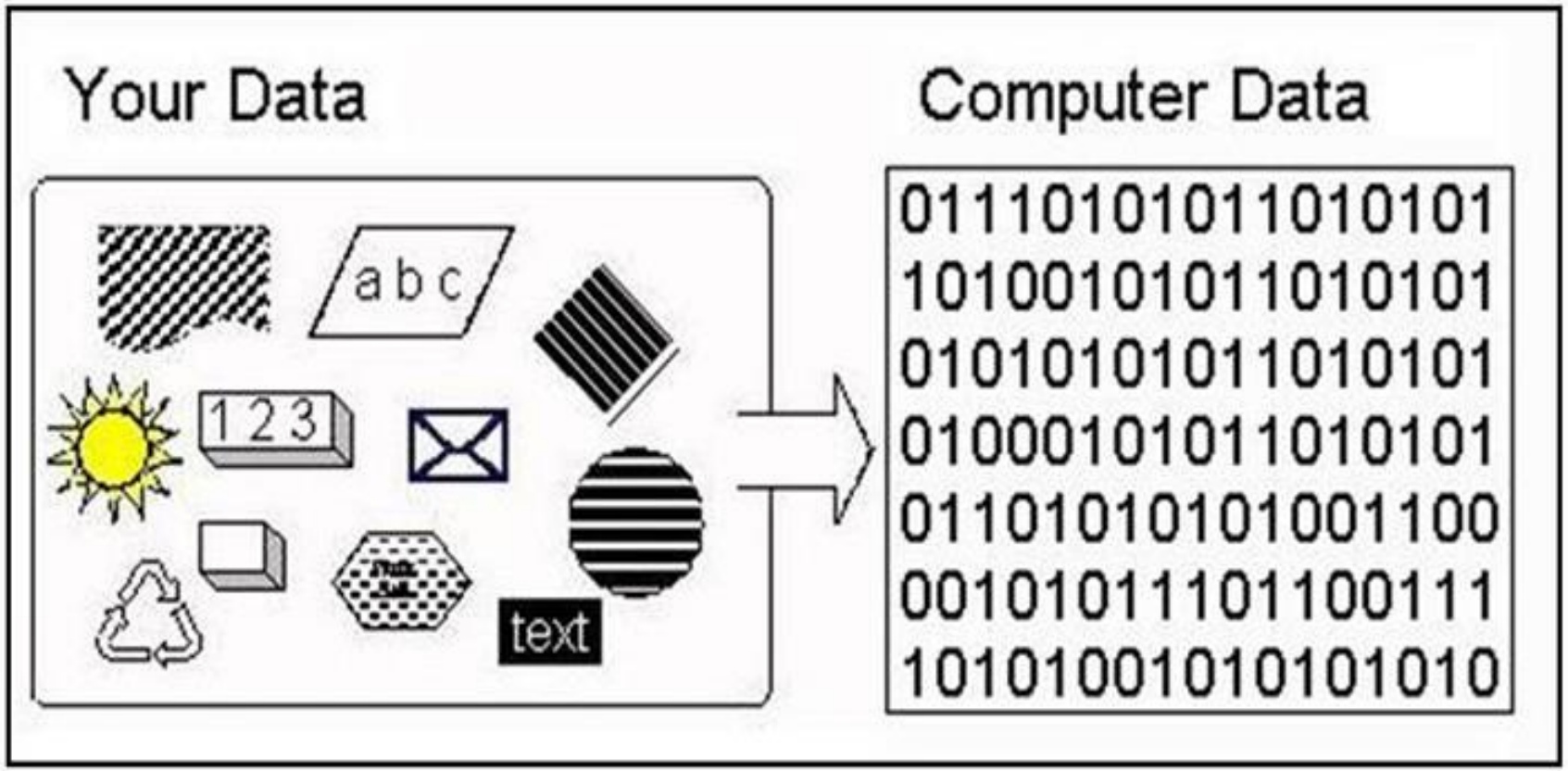


Figure : Representation of Different forms of data

Encoding and Decoding

- Encoding is the process of putting a sequence of characters (letters, numbers, punctuation, and certain symbols) into a specialized digital format for efficient transmission or transfer.

- Decoding is the opposite process -- the conversion of a digital signal into a sequence of characters.

- Encoding and decoding are used in data communications, networking, and storage.

Encoding formats

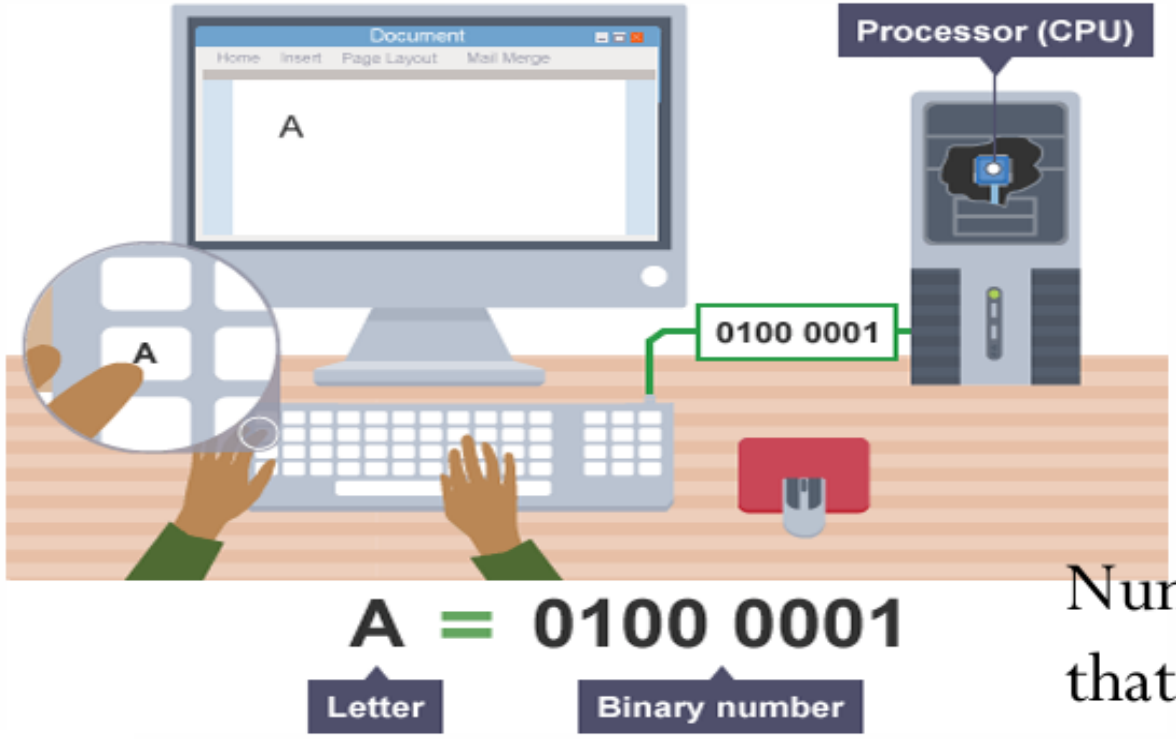
- Encoding formats have been standardised to help compatibility across different platforms.

  - audio is encoded as audio file formats, eg mp3, WAV, AAC
  - video is encoded as video file formats, eg MPEG4, H264
  - text is encoded in character sets, eg ASCII, Unicode
  - images are encoded as file formats, eg BMP, JPEG, PNG

- The more bits used in a pattern, the more combinations of values become available.
- This larger number of combinations can be used to represent many more things, eg a greater number of different symbols, or more colours in a picture.

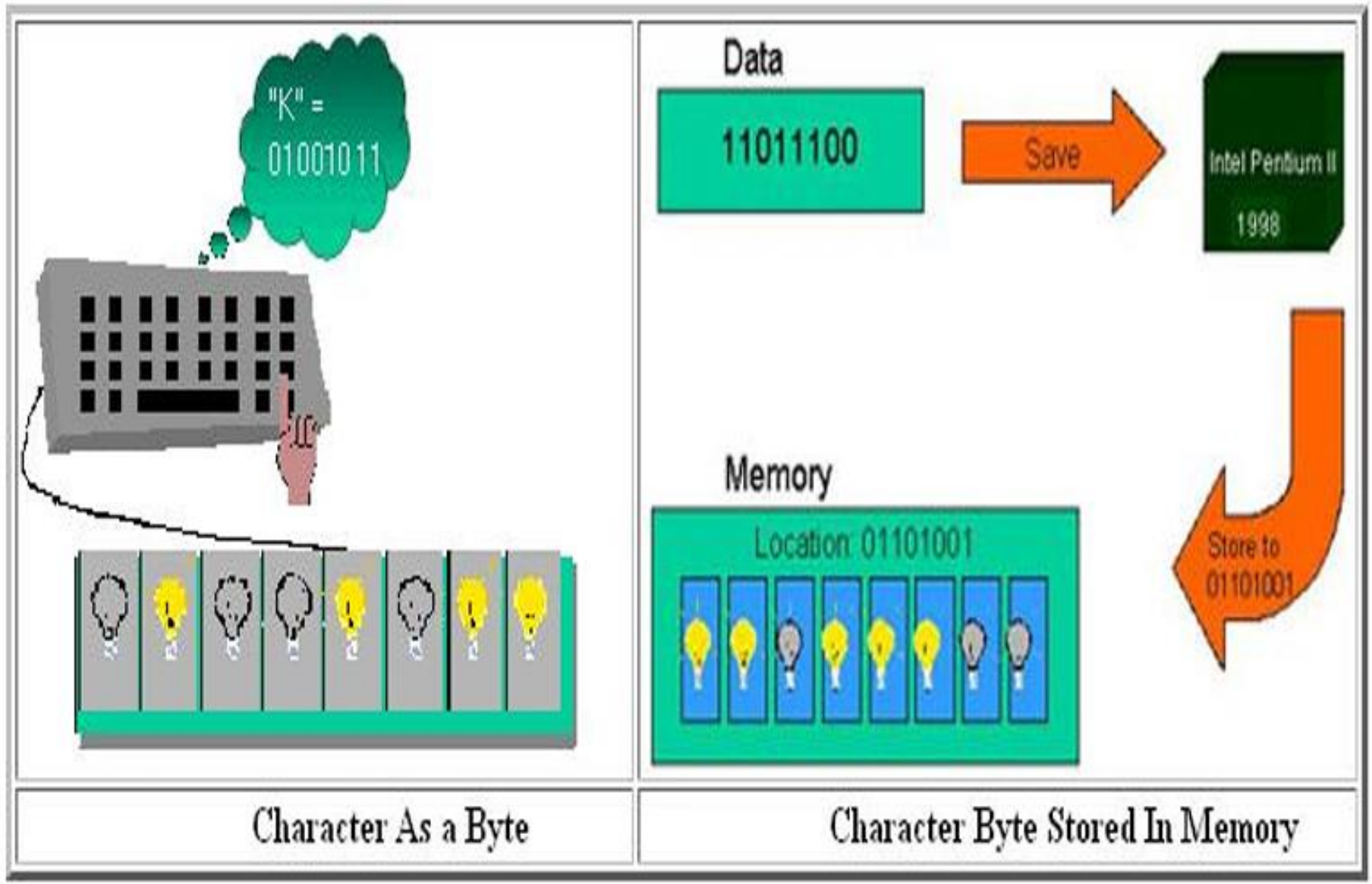
Character sets

- Every word is made up of symbols or characters.
- When you press a key on a keyboard, a number is generated that represents the symbol for that key.
- This is called a **character code**.
- A complete collection of characters is a **character set**.



Numbers are generated that represent keyboard symbols

Representing Character



- You can check what character encoding your **web browser** is using by looking in your browser settings:

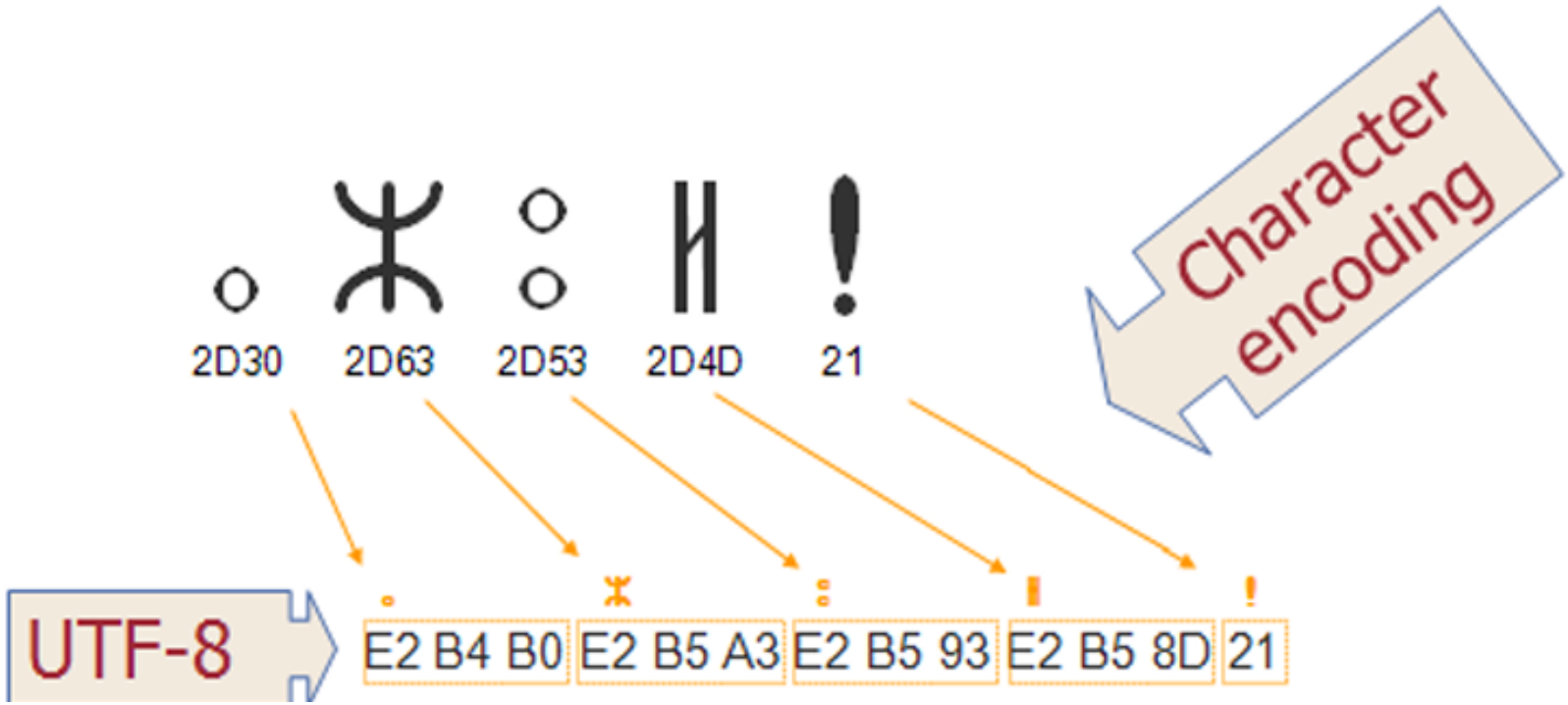
- Mozilla Firefox > Tools > Page Info: Encoding

- Microsoft Internet Explorer > View > Encoding

- Google Chrome > Tools > Encoding

Text Encoding

- Characters are usually encoded as integer values using the encoding schemes.
- The associations between numbers and text are known collectively as a character encoding scheme.



ASCII - American Standard Code for Information Interchange

- Focus on **unaccented, English letters.**

- Every letter, number, capital, etc

- Represented by codes 0-127.

- Space, 32; "A", 65; "a", 97.

- Only the 7-bit patterns were standardized under ASCII.
- Standard 8-bit ASCII codes
    - start with a zero-valued bit (followed by 7-bit ASCII code).

EBCDIC (Extended Binary Coded Decimal Interchange Code)

- Developed by IBM.
- Restricted mainly to IBM or IBM compatible mainframes.
- Conversion software to/from ASCII available.
- Common in archival data.
- Character codes differ from ASCII.

	ASCII	EBCDIC
Space	20_{16}	40_{16}
A	41_{16}	$C1_{16}$
b	62_{16}	82_{16}

Unicode

- **Unicode** uses between 8 and 32 bits per character

- so it can represent characters from languages from all around the world.

- It is commonly used across the internet.

- As it is larger than ASCII, it might take up more **storage** space when saving documents.

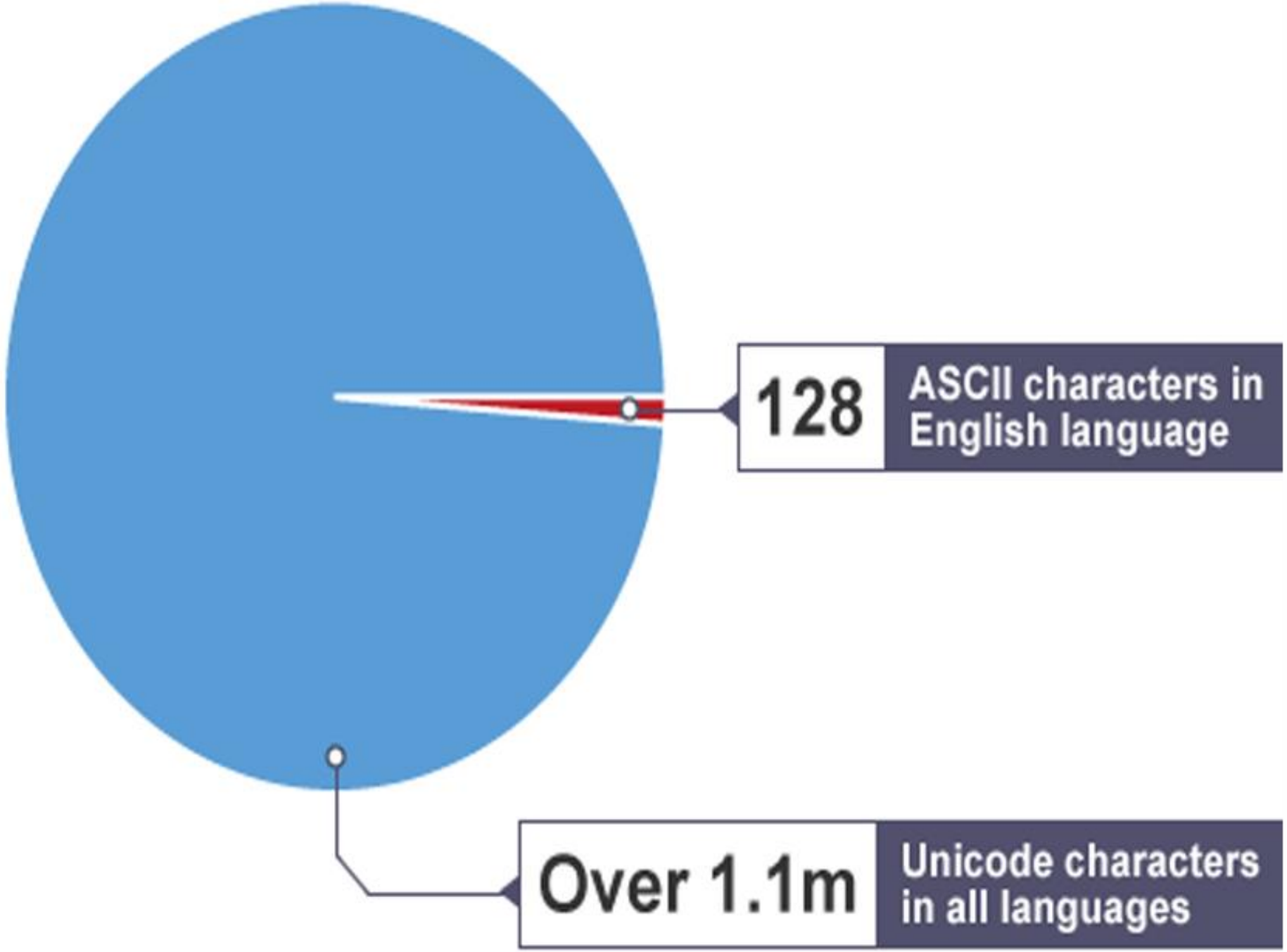
- Global companies, like Facebook and Google, would not use the ASCII character set because their users communicate in many different languages.

ASCII only contains 127 characters
though an extended version exists with 257 characters

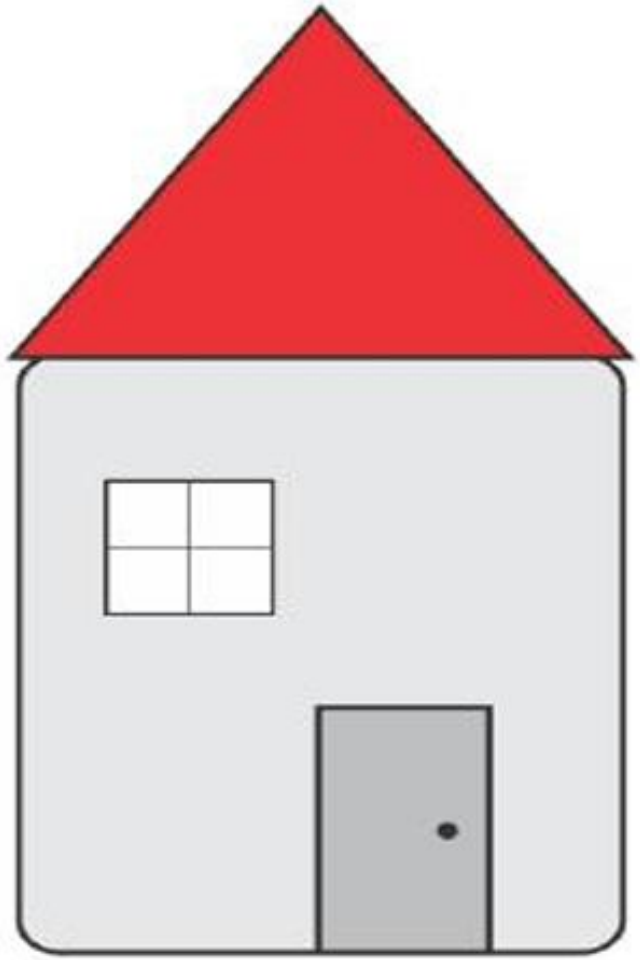
This is by far not enough as it is too restrictive to the English language.

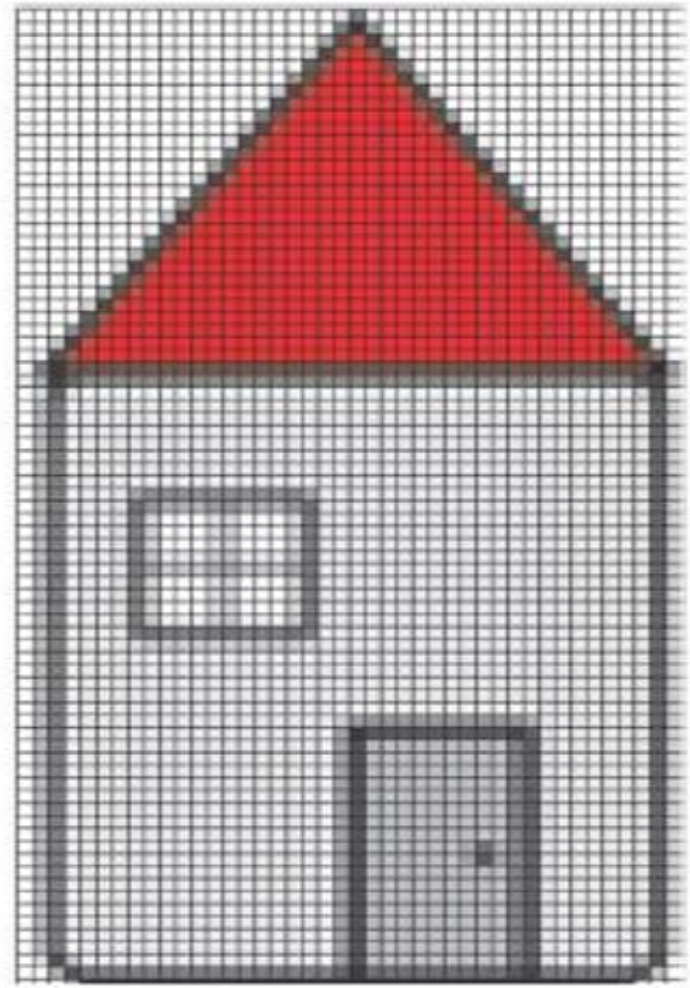
UNICODE was developed to alleviate this problem:

the latest version, UNICODE 5.1.0 contains more than 100,000 characters, covering most existing languages.


128
ASCII characters in English language
Over 1.1m
Unicode characters in all languages

PIXELS





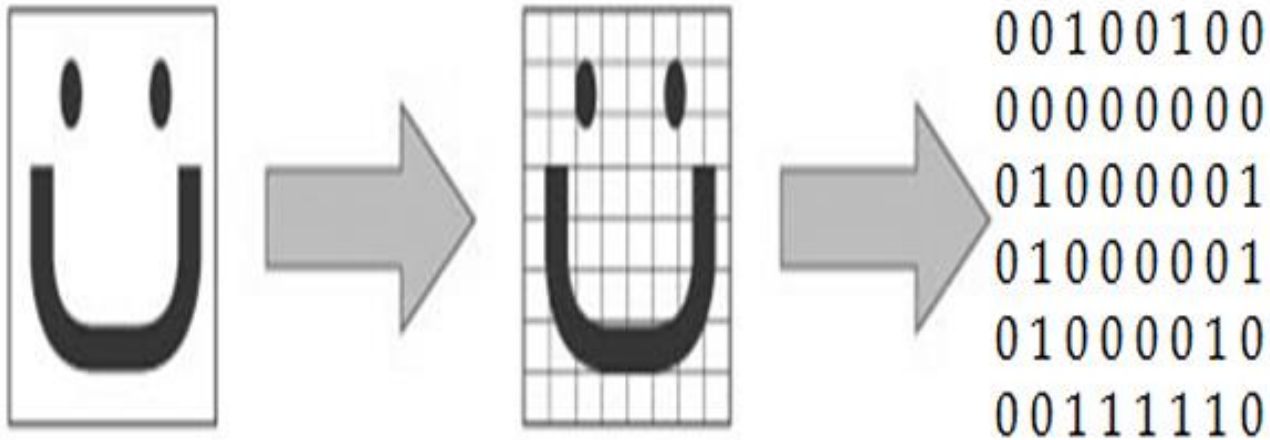
Pixels





Image Encoding

- Binary representation of bitmap images
- All bitmap images are stored as array of pixels.
- A monochrome images store
  - 1 for black pixel and
  - 0 for a white pixel
  - (or vice versa depending on the encoding protocol)
- It could also be necessary to store the dimensions of the image.

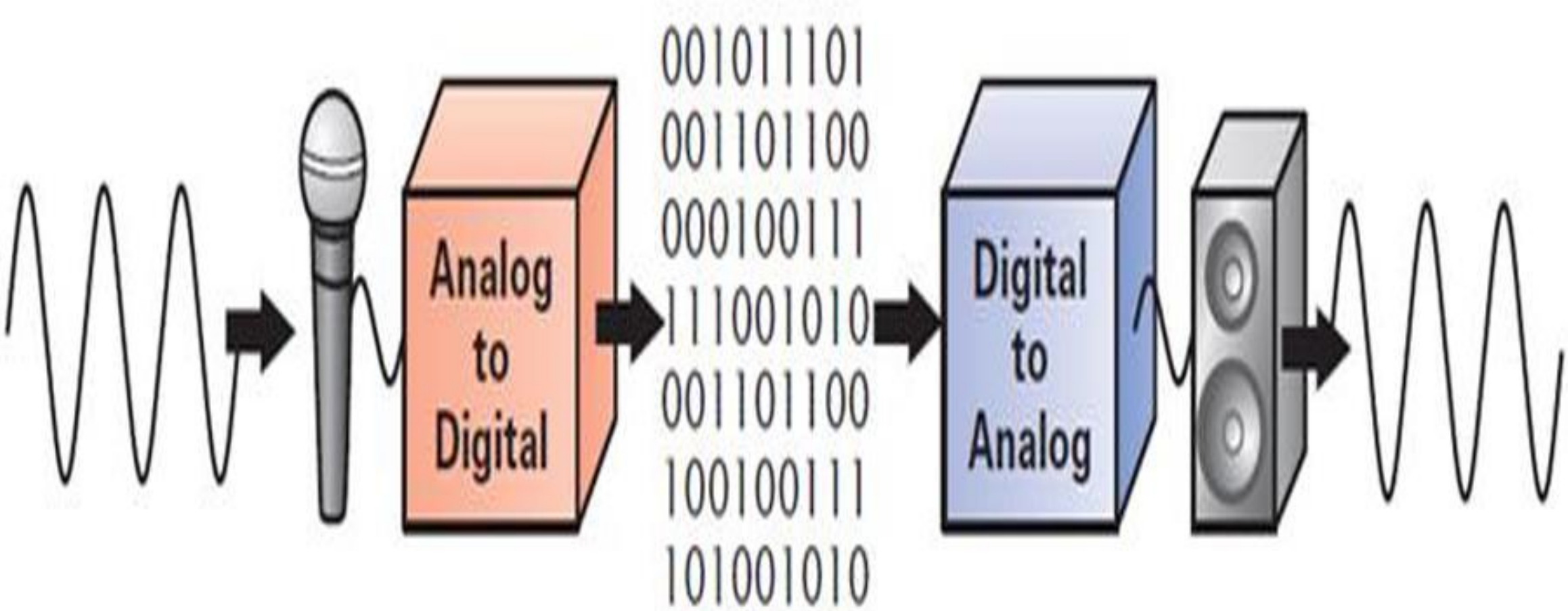


```
00000000
00100100
00000000
01000001
01000001
01000010
00111110
00000000
```

1. To generate a bitmap, the image is first partitioned into grid of pixels, here 8 X 8

2. For a Black-and-White bitmap, a black pixel is represented with a 1 bit, a white pixel with a 0 bit

Representing Sound



Data Compression

- To **reduce the volume of data** to be transmitted (text, fax, images).

- To **reduce the bandwidth** required for transmission and to reduce storage requirements (speech, audio, video).



Classification

- *Lossless compression*
  - Lossless compression for legal and medical documents, computer programs.
- *Lossy compression*
  - Digital audio, image, video where some errors or loss can be tolerated.



Video and Audio Compression

- Video and Audio files are very large. Unless we develop and maintain very high bandwidth networks (Gigabytes per second or more) we have to compress the data.
- Relying on higher bandwidths is not a good option.
- Compression becomes part of the representation or *coding* scheme which have become popular audio, image and video formats.