Module 2: Bits and Bytes

Prof. Dr. Florian Haselbeck
Memoona Shakeel

Weihenstephan-Triesdorf University of Applied Sciences

Learning outcomes

Understand how information is represented and stored digitally.

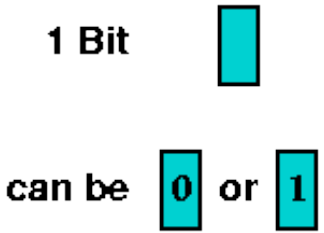
Key Topics

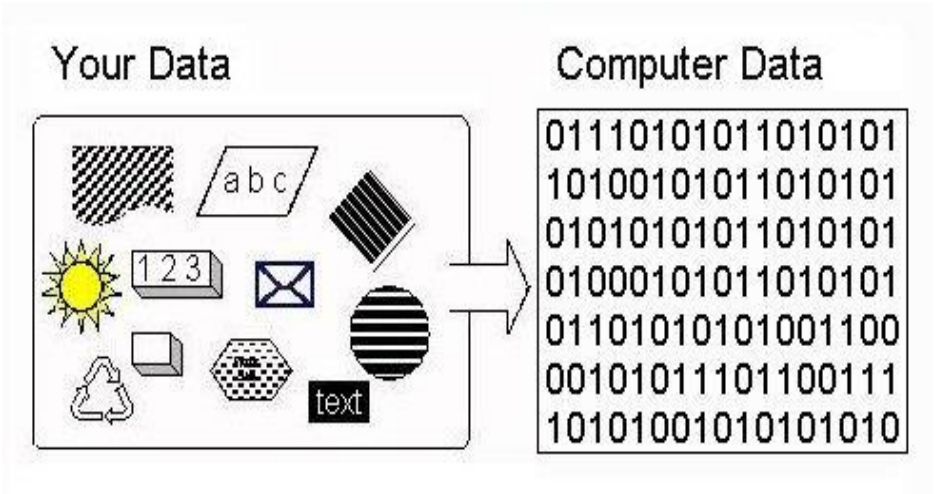
- What is a bit?
- What is a byte?
- Binary numbers
- Hexadecimal
- File sizes (KB, MB, GB)
- How is data stored?

What is a bit?

A bit is the smallest unit of digital information, representing either an "off" (0) or an "on" (1) state.

Why only 0 and 1?
A computer is built from billions of tiny transistors or electronic switches. Each switch is either OFF (0) or ON (1). There are only two reliable states, so all computer data uses these two values.



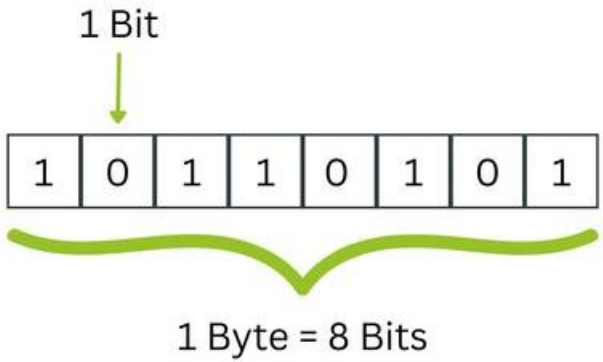


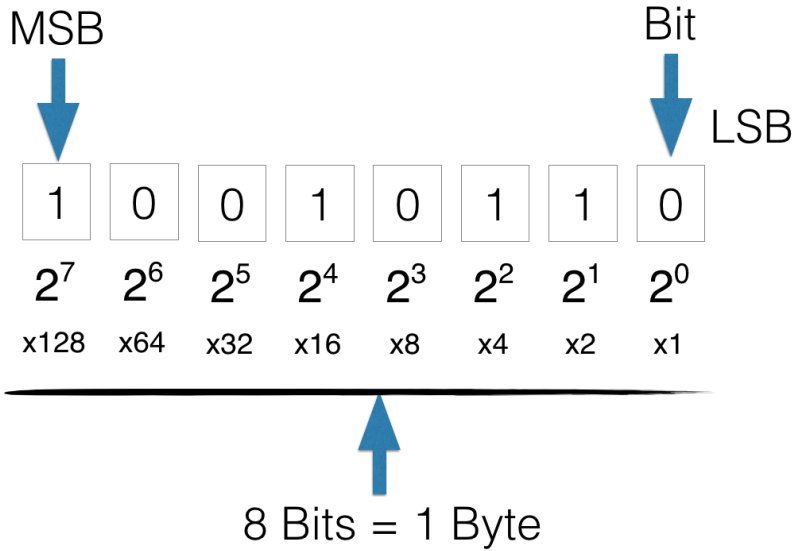
One bit alone is not very useful, but grouping 8 bits together and you get a byte, which can hold 256 different values.

What is a byte?

A byte is a group of 8 bits. It is the fundamental unit that computers use to store and process data.

8 bits → 2^8 = 256 different patterns (from 00000000 to 11111111) Enough for all letters, digits and symbols

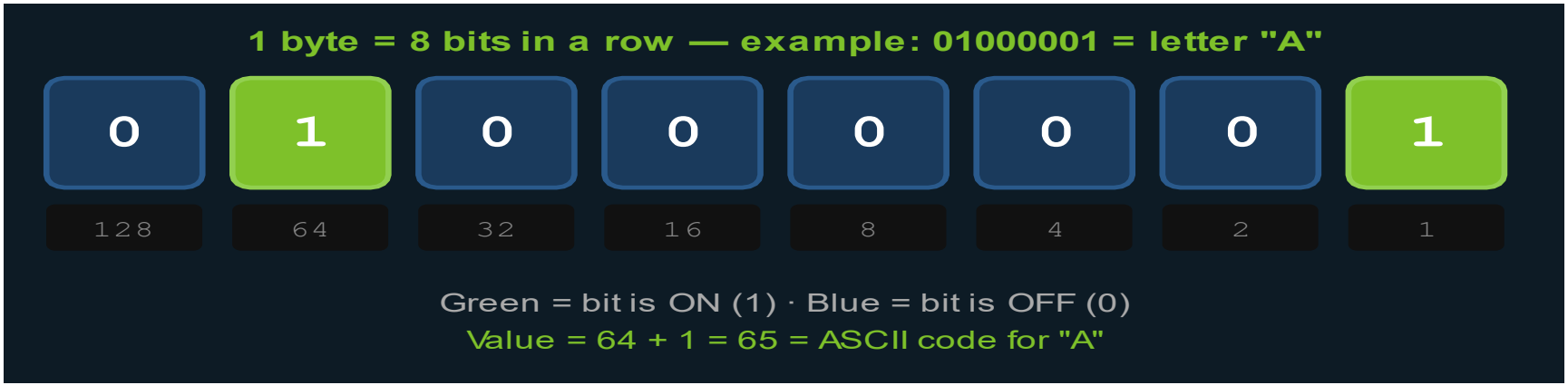




What is a byte?

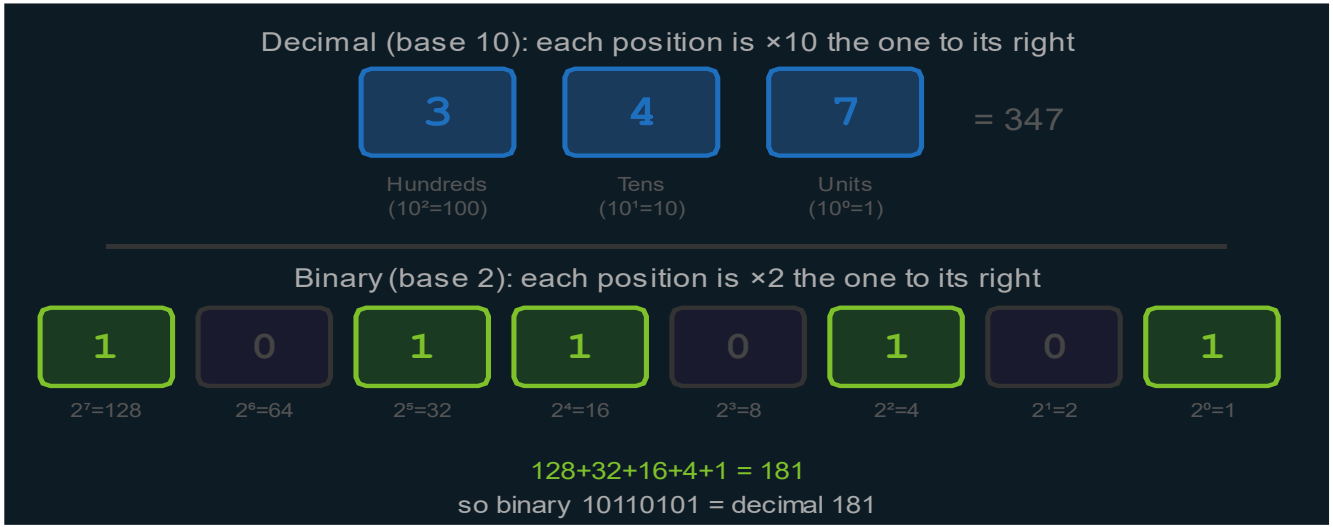
What fits in one byte? One character of text (letter, digit, symbol). A number from 0 to 255. Part of a larger number or decimal

Common Size: 1 byte = 1 character, 2 bytes = small integer (0–65,535), 4 bytes = standard integer or float, 8 bytes = large number or double



Binary numbers (counting in base 2)

Key idea: In binary, each position to the left is worth double the one to its right, just like decimal but ×2 instead of ×10.



Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Rule to remember: starting from the RIGHT, positions are worth 1, 2, 4, 8, 16, 32, 64, 128 (each doubles)

Binary numbers

Decimal Number	Binary Number	Decimal Number	Binary Number
1	1	11	1011
2	10	12	1100
3	11	13	1101
4	100	14	1110
5	101	15	1111
6	110	16	10000
7	111	17	10001
8	1000	18	10010
9	1001	19	10011
10	1010	20	10100

ASCII Code: Character to Binary

0	0011 0000	O	0100 1111	m	0110 1101
1	0011 0001	P	0101 0000	n	0110 1110
2	0011 0010	Q	0101 0001	o	0110 1111
3	0011 0011	R	0101 0010	p	0111 0000
4	0011 0100	S	0101 0011	q	0111 0001
5	0011 0101	T	0101 0100	r	0111 0010
6	0011 0110	U	0101 0101	s	0111 0011
7	0011 0111	V	0101 0110	t	0111 0100
8	0011 1000	W	0101 0111	u	0111 0101
9	0011 1001	X	0101 1000	v	0111 0110
A	0100 0001	Y	0101 1001	w	0111 0111
B	0100 0010	Z	0101 1010	x	0111 1000
C	0100 0011	a	0110 0001	y	0111 1001
D	0100 0100	b	0110 0010	z	0111 1010
E	0100 0101	c	0110 0011	.	0010 1110
F	0100 0110	d	0110 0100	,	0010 0111
G	0100 0111	e	0110 0101	:	0011 1010
H	0100 1000	f	0110 0110	;	0011 1011
I	0100 1001	g	0110 0111	?	0011 1111
J	0100 1010	h	0110 1000	!	0010 0001
K	0100 1011	I	0110 1001	'	0010 1100
L	0100 1100	j	0110 1010	"	0010 0010
M	0100 1101	k	0110 1011	(	0010 1000
N	0100 1110	l	0110 1100	)	0010 1001
				space	0010 0000

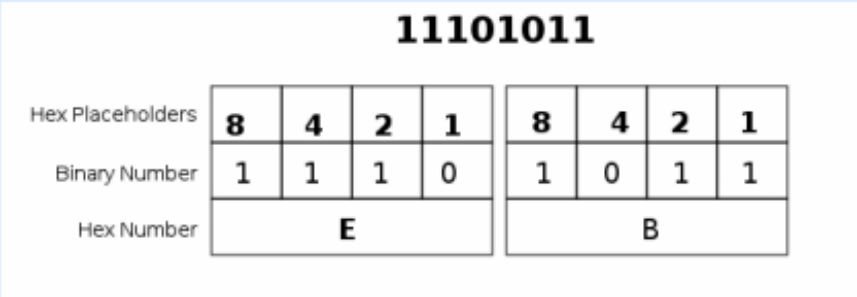
Hexadecimal (shorter way to write binary)

Why hex? Binary strings are long and hard to read. Hexadecimal (base 16) lets you write the same value in a quarter of the space.

4 bits → 1 hex digit

8 bits → 2 hex digits (1 byte)

FF hex = 11111111 = 255 decimal

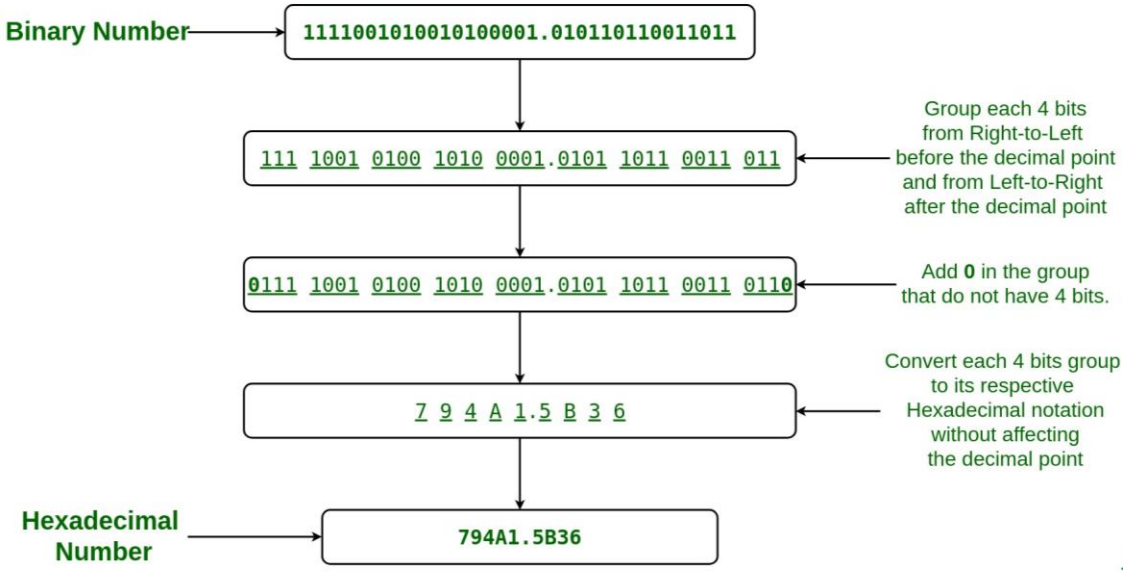


Decimal (Base 10)	Binary (Base 2)	Hexadecimal (Base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

You will see hex every day in computing: memory addresses, color codes, network MAC addresses.

Hexadecimal (shorter way to write binary)

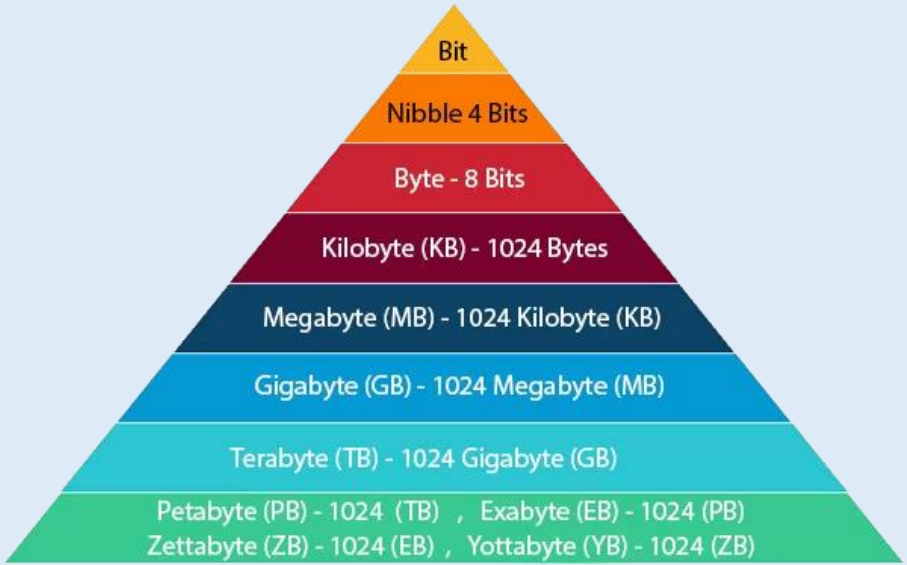
How to convert binary number to hexadecimal number?

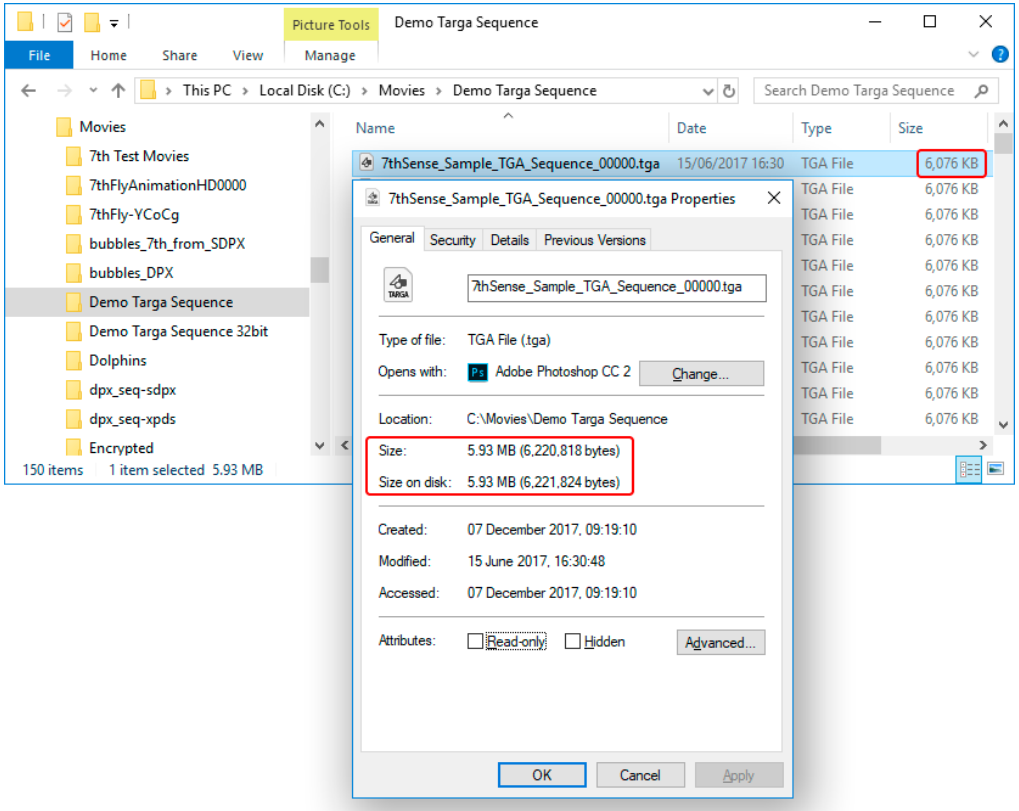


File Size (KB, MB, GB, TB)

Key idea: Storage is measured in bytes. Because computers work in powers of 2, each unit is 1,024× the one below it.

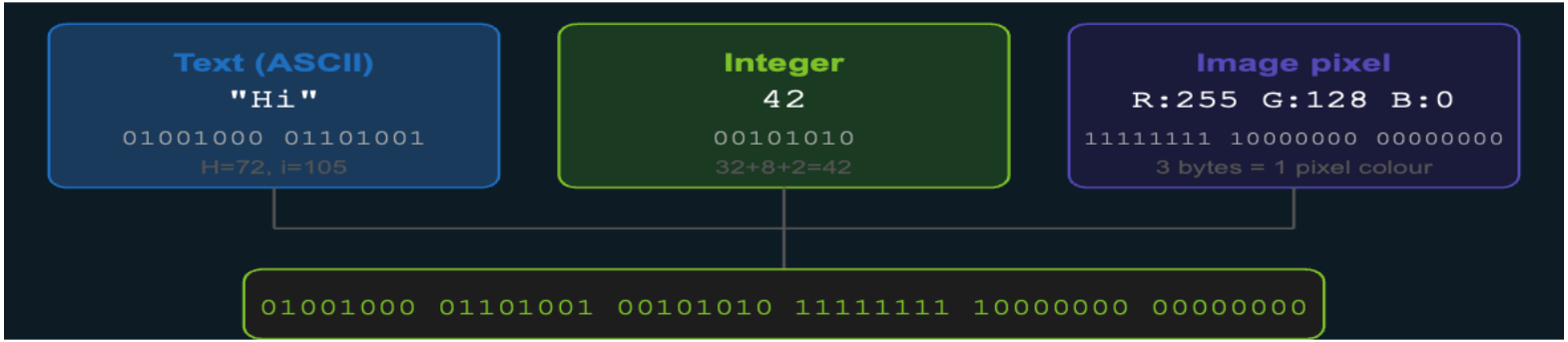
Note: hard drive manufacturers use 1 KB = 1,000 bytes (decimal). OS uses 1 KB = 1,024 bytes (binary). That is why a '256 GB' drive shows as ~238 GB.




Picture Tools
Demo Targa Sequence
File
Home
Share
View
Manage
This PC › Local Disk (C:) › Movies › Demo Targa Sequence
Search Demo Targa Sequence
Movies
7th Test Movies
7thFlyAnimationHD0000
7thFly-YCoCg
bubbles_7th_from_SDPX
bubbles_DPX
Demo Targa Sequence
Demo Targa Sequence 32bit
Dolphins
dpx_seq-sdpx
dpx_seq-xpds
Encrypted
150 items
1 item selected 5.93 MB
Name
Date
Type
Size
7thSense_Sample_TGA_Sequence_00000.tga
15/06/2017 16:30
TGA File
6,076 KB
7thSense_Sample_TGA_Sequence_00000.tga Properties
General
Security
Details
Previous Versions
7thSense_Sample_TGA_Sequence_00000.tga
Type of file:
TGA File (.tga)
Opens with:
Adobe Photoshop CC 2
Change...
Location:
C:\Movies\Demo Targa Sequence
Size:
5.93 MB (6,220,818 bytes)
Size on disk:
5.93 MB (6,221,824 bytes)
Created:
07 December 2017, 09:19:10
Modified:
15 June 2017, 16:30:48
Accessed:
07 December 2017, 09:19:10
Attributes:
Read-only
Hidden
Advanced...
OK
Cancel
Apply

How data is stored? (everything is bits)

Key idea: A computer does not know about letters, images or numbers by itself, it only stores bits. The encoding rules tell it what the bits mean.



Text: ASCII / Unicode: Every character has a number code.

A=65, B=66, a=97, space=32

'Hello' = 5 bytes = 40 bits

Integers: Stored as straight binary.

Python int: 28 bytes (with overhead)

C int: exactly 4 bytes (32 bits)

Images: Each pixel = 3 bytes (R, G, B, each 0–255).

A 12 MP photo ≈ 36 MB uncompressed.

JPEG compression reduces this to ~3 MB

Conversion Examples

Binary → Decimal

Binary Digit	Place Value	Product
1	$2^3 = 8$	8
1	$2^3 = 4$	4
0	$2^1 = 2$	0 (since the binary digit is 0, we can ignore this calculation)
1	$2^0 = 1$	1

Result: 8 + 4 + 0 + 1 = 13

Conversion Examples

Decimal → Binary

Step 1: Divide the given number 13 repeatedly by 2 until you get "0" as the quotient

$13 \div 2 = 6$ (Remainder 1)
$6 \div 2 = 3$ (Remainder 0)
$3 \div 2 = 1$ (Remainder 1)
$1 \div 2 = 0$ (Remainder 1)

Step 2: Write the remainders in the revers 1 1 0 1 order

$\therefore 13_{10} = 1101_2$

(Decimal) (Binary)

Conversion Examples

Decimal → Binary

Converting 25 to binary

Division	Quotient	Remainder	Why does it have this reminder?
25 ÷ 2	12	1	25÷2=12,5 which is bigger than the quotient. So, we place 1
12 ÷ 2	6	0	12 ÷ 2 = 6 It's the same as the quotient. So, we place 0
6 ÷ 2	3	0	6 ÷ 2 = 3 It's the same as the quotient. So, we place 0.
3 ÷ 2	1	1	3 ÷ 2 = 1,5 It's bigger than the quotient. So, we place 1.
1 ÷ 2	0	1	3 ÷ 2 = 1,5 It's bigger than the quotient. So, we place 1.

Binary equivalent: 11001

Thank you!

Contact Detail

Prof. Dr. Florian Haselbeck

florian.haselbeck@hswt.de

Raum D1.230a

For any questions, please feel free to use the Moodle-Course Forum!