GCSE Computer Science Knowledge Organiser
SLR 1.2.2 Memory and Storage:
The units of data storage

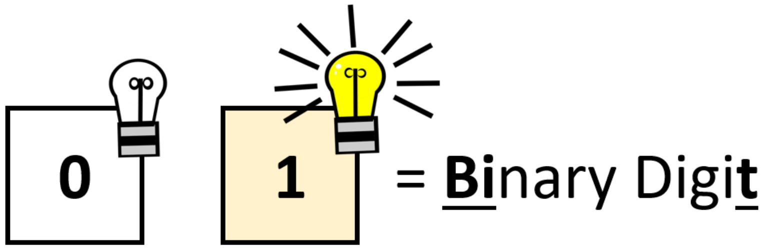
Kilo = Thousand
Mega = Million
Giga = Billion
Tera = Trillion
Peta = Quadrillion

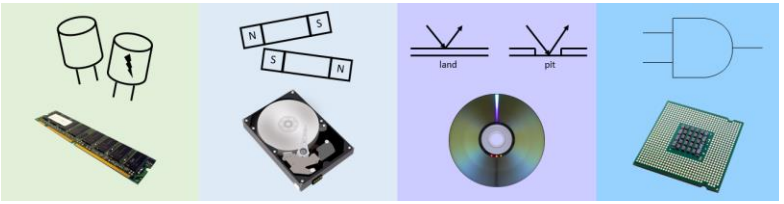
Key Terminology	BCS Definition
Bit	"The smallest unit of storage, represented by either a binary 1 or 0." (b)
Nibble	"Half a byte. Four bits."
Byte	"A collection of eight bits." (B)
Kilobyte	"One kilobyte (KB) is 1024 bytes. For the purpose of calculations in an exam, you can treat a kilobyte as 1000 bytes."
Megabyte	"One megabyte (MB) is 1024 kilobytes (KB). For the purpose of calculations in an exam, you can treat a megabyte as 1000 KB."
Gigabyte	"One gigabyte (GB) is 1024 megabytes (MB). For the purpose of calculations in an exam, you can treat a gigabyte as 1000 MB."
Terabyte	"One terabyte (TB) is 1024 gigabytes (GB). For the purpose of calculations in an exam, you can treat a terabyte as 1000 GB."
Petabyte	"One petabyte (PB) is 1024 terabytes (TB). For the purpose of calculations in an exam, you can treat a petabyte as 1000 TB."

Calculating from one unit to another

	Unit	
Multiply by 8	Bit	Divide by 8
	Byte	
Multiply by 1000	Kilobyte	Divide by 1000
	Megabyte	
	Gigabyte	
	Terabyte	
	Petabyte	

Type	Size	
Microsoft Word Document	1,470 KB	= 1.47 MB
Adobe Acrobat Document	454 KB	
Microsoft PowerPoint Presentation	665 KB	
Microsoft Excel Worksheet	20 KB	
Microsoft Excel Worksheet	20 KB	
Microsoft PowerPoint Presentation	599 KB	
Microsoft Excel Worksheet	22 KB	
Microsoft Excel Worksheet	22 KB	
Microsoft PowerPoint Presentation	740 KB	= 0.74 MB
Microsoft Excel Worksheet	25 KB	
Microsoft Excel Worksheet	25 KB	= 25000B (x 1000)
Microsoft Excel Worksheet	40 KB	= 200000b (x 8)
Microsoft PowerPoint Presentation	479 KB	





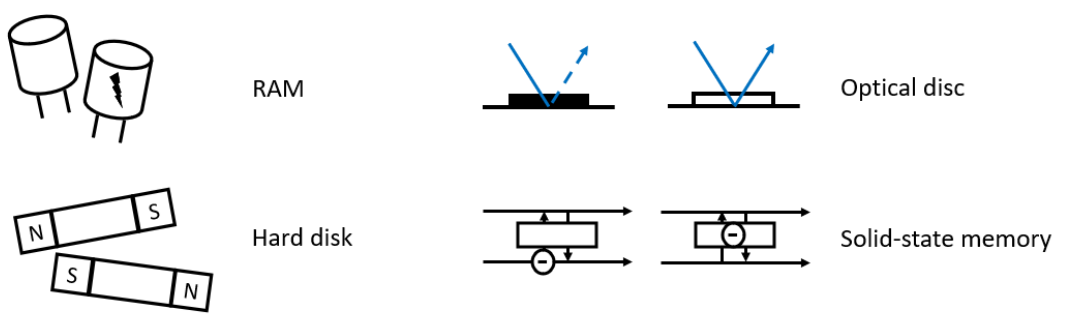
Unit	Symbol	Binary value	Decimal value	Approximation	Written
Bit	b	0 or 1			
Nibble		4 bits		½ byte	
Byte	B	8 bits		1 byte	
Kilobyte	KB	1024 bits	10^3	1,000 bytes	One thousand bytes
Megabyte	MB	1024^2 bits	10^6	1,000,000 bytes	One million bytes
Gigabyte	GB	1024^3 bits	10^9	1,000,000,000 bytes	One billion bytes
Terabyte	TB	1024^4 bits	10^{12}	1,000,000,000,000 bytes	One trillion bytes
Petabyte	PB	1024^5 bits	10^{15}	1,000,000,000,000,000 bytes	One quadrillion bytes

GCSE Computer Science Knowledge Organiser
SLR 1.2.2 Memory and Storage:
Procession Binary Data and Data capacity and calculating data capacity requirements

With just two states, electronic components are:

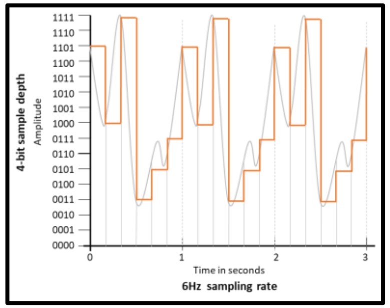
- Easier to manufacture
- Cheaper
- More reliable

Examples of using two states to store data:



Calculating sound file sizes

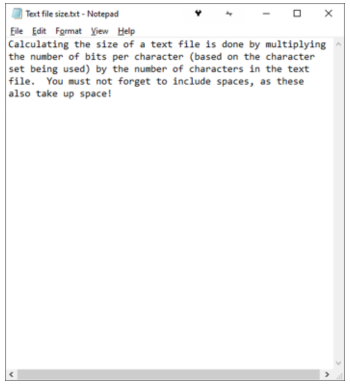
Sample rate × Duration (s) × Bit depth



Sound file size		
Sample rate	6	Number of samples per second
Duration	3	Length of sample in seconds
Bit depth	4	Number of bits required to store each sample
6 × 3 × 4	**=**	**72 bits** = 9 bytes

Calculating text file sizes

Bits per character × Number of characters



Text file size.txt - Notepad

File Edit Format View Help

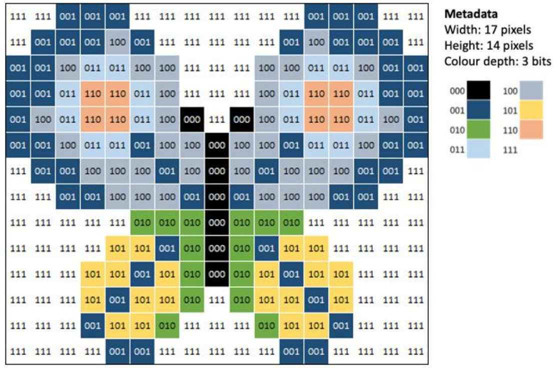
Calculating the size of a text file is done by multiplying the number of bits per character (based on the character set being used) by the number of characters in the text file. You must not forget to include spaces, as these also take up space!

Text file size		
Bits per character	7	Based on the standard ASCII character set being used
No. of characters	244	Including spaces
7 × 244	**=**	**1,708 bits** = 214 bytes

Calculating image file sizes

Colour depth × Image height (px) × Image width (px)

Image file size		
Image height	14	Height measured in pixels
Image width	17	Width measured in pixels
Colour depth	3	Number of bits required to store each pixel
14 × 17 × 3	**=**	**714 bits** = 90 bytes



GCSE Computer Science Knowledge Organiser
SLR 1.2.2 Memory and Storage:
Binary Conversion

Key Terminology	BCS Definition
Denary numbers	"A numerical system of notation that uses 10 as its base. The ten decimal base digits are 0 – 9."
Denary numbers	Random-Access Memory: "Volatile (data is lost when the computer is powered off). Read-and-write. Purpose: Temporary storage of currently executing instructions and data – e.g., applications and the operating system."

There are two main methods you can use to convert from denary to binary:

- Divide-by-two
- Binary number line

Both are equally as valid, so use whichever you feel most comfortable with.

Denary to binary: Divide-by-two method

Convert the denary number 89 into binary

Take the denary number and divide by 2 and write down the remainder

Division by 2	=	Remainder
89	44.5	1
44	22.0	0
22	11.0	0
11	5.5	1
5	2.5	1
2	1	0
1	0.5	1
		0

Write the answer down backwards. In this example from the bottom of the table up.

89 in binary is 01011001

Denary to binary: Binary number line method

Convert the denary number 89 into binary

1. **Write the number line out**

128	64	32	16	8	4	2	1

2. **Work left to right, take your target number and if it is smaller than the first number (128), write a 0.**

89

128	64	32	16	8	4	2	1
0							

3. **If it larger, write a 1 and subtract the target from the line number**

89 - 64 = 25

128	64	32	16	8	4	2	1
0	1						

4. **Repeat till you get to the end or 0**

89 - 64 = 25 - 16 = 9 - 8 = 1

128	64	32	16	8	4	2	1
0	1	0	1	1	0	0	1

Binary to Denary Conversion

Convert the binary number 01100110 into denary

1. **Write the number line out**

128	64	32	16	8	4	2	1

2. **Write out your binary number under your number line**

128	64	32	16	8	4	2	1
0	1	1	0	0	1	1	0

3. Add all the numbers with a 1 under it.

128	64	32	16	8	4	2	1	
0	1	1	0	0	1	1	0	
	64 +	32 +		+	4 +	2		= 102

GCSE Computer Science Knowledge Organiser
SLR 1.2.2 Memory and Storage:
Binary addition

Key Terminology	BCS Definition
Binary arithmetic	"The process of adding two or more positive 8-bit binary numbers (0 – 255)."
Overflow	"The generation of a number that is too large to be represented by the device intended to store it."

The rules of binary addition

Rule	Calculation	Action
1.	0+0 =	Write 0
2.	0+1 or 1+0 =	Write 1
3.	1+1 =	Write 0 carry 1
4.	1+1+1=	Write 1 carry 1

Example 1

Working right to left take each column one at a time and apply the rules of binary addition. Colum 1 is on the right.

In this example each column you are adding 1+0 or 0+1. So, you follow the rule 2 and write 1

Number	0	1	0	1	0	1	0	1
Number	1	0	1	0	1	0	1	0
Answer	1	1	1	1	1	1	1	1
Carries								

Example 2

Working right to left take each column one at a time and apply the rules of binary addition. Colum 1 is on the right.

In column 2 and 4, you see an example of **rule 3**, 1+1 = write 0 carry 1

In columns 3 and 5, because of the carry, this turns 0+0 into 0+0+1. **Rule 2** is therefore followed

Number	0	0	1	0	1	0	1	1
Number	0	1	0	0	1	0	1	0
Answer	0	1	1	1	0	1	0	1
Carries				1		1		

Example 3

Working right to left take each column one at a time and apply the rules of binary addition. Colum 1 is on the right.

In column 4, you see an example of **rule 3**, 1+1 = write 0 carry 1

In columns 5 and 6 , you see an example of **rule 4**, 1+1+1 = write 1 carry 1

Number	0	0	1	1	1	0	1	1
Number	0	1	1	1	1	1	0	0
Answer	1	0	1	1	0	1	1	1
Carries	1	1	1	1				

Example 4

Working right to left take each column one at a time and apply the rules of binary addition. Colum 1 is on the right.

In columns 2,7 & 8, you see an example of **rule 3**, 1+1 = write 0 carry 1

In columns 3-5 , you see an example of **rule 4**, 1+1+1 = write 1 carry 1

In columns 6, because of the carry, this turns 0+0 into 0+0+1. **Rule 2** is therefore followed

In Column 9, the carry from column 8 creates a 9th bit. This does not fit into a Byte (8 bits) and is called and **Overflow** or **Overflow Error.**

Number		0	1	0	1	1	1	1	1
Number		1	1	0	1	1	1	1	0
Answer		0	0	1	1	1	1	0	1
Carries	1	1		1	1	1	1		

GCSE Computer Science Knowledge Organiser
SLR 1.2.2 Memory and Storage:
Hexadecimal

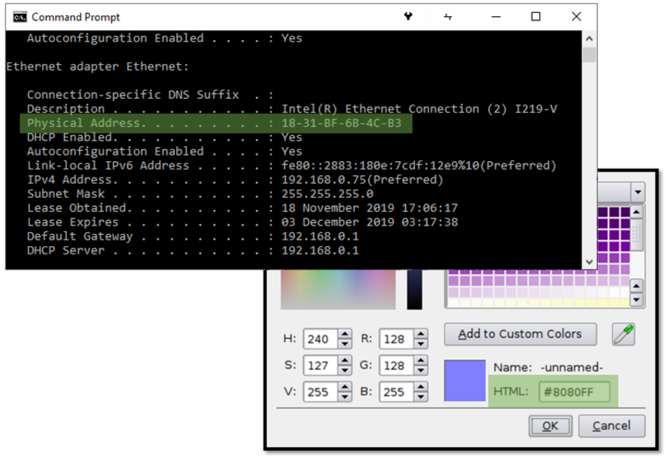
Key Terminology	BCS Definition
Hexadecimal	"A numerical system of notation that uses 16 rather than 10 as its base. The 16 hex base digits are 0 – 9 and the letters A – F."

What is Hexadecimal?
Hexadecimal is a 16 based number system. Using the numbers 0-9 and then the letters A-F.

Why do we use Hexadecimal?
For humans working with 8 bit binary (groups of eight 1's and 0's) is difficult as we are more than likely to make mistakes.

Converting to binary is easy to do and creates two digits which is easier for humans to remember and work with rather than eight 1's and 0's.

Examples of Hexadecimal in use:
1. MAC addresses – the physical address for your devices - are written in Hexadecimal.
2. HTML colour codes uses 3 sets of hexadecimal. The first to represent **RED** the second to represent **GREEN** and the third to represent **BLUE** (**RGB**).



Denary	Binary	Hexadecimal
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

Converting Denary to Hexadecimal

1. **Convert the denary into binary**

Denary	**230**							
Binary	128	64	32	16	8	4	2	1
	1	1	1	0	0	1	1	0

2. **Split the binary in half to create two nibbles (4bits) and covert each nibble back into a Denary number**

Binary	**128**	**64**	**32**	**16**	**8**	**4**	**2**	**1**
	1	1	1	0	0	1	1	0
Nibble	8	4	2	1	8	4	2	1
	8+4+2=14				4+2=6			

3. **Convert each nibble into the its hexadecimal value**
   - **If it is between 0-9 this is the same in hexadecimal**
   - **If it is 10 or higher. Convert to letters A=10, B=11, C=12… F=15.**

Denary	**230**							
Binary	128	64	32	16	8	4	2	1
	1	1	1	0	0	1	1	0
Nibble	8	4	2	1	8	4	2	1
Hexadecimal	**E**				**6**			

- To convert Hexadecimal into Denary, follow the steps above in reverse order.
- Notice that you can also convert Binary into Hexadecimal by following only Steps 2 and 3.

GCSE Computer Science Knowledge Organiser
SLR 1.2.2 Memory and Storage:
Binary Shift

Key Terminology	BCS Definition
Binary Shift	“Allows you to easily multiply or divide a base-2 binary number. A left shift multiplies the number by 2, while a right shift divides it by 2.

Binary left shift (Multiply by 2)

Performing a shift of 1 to the left

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	1	0	22

1. **Start are the 128 column. Move the bit into the next column on the left – in this case there is not a column to move it to, so we drop this bit and continue**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	1	0	22

2. **Move to the next column on the left, 64, and move the bit over to the 128 column**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	1	0	22
0								

3. **Repeat this with the remaining columns**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	1	0	22
0	0	1	0					

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	1	0	22
0	0	1	0	1	1			

4. **Any gaps are filled with a zero**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	1	0	22
0	0	1	0	1	1	0	0	

5. **By performing a 1-bit left shift, 22 has become 44.**

   We have multiplied the original number by 2.

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	1	0	22
0	0	1	0	1	1	0	0	44
		32 +		8 +	4			

Exam Tip:
If the question asked you to perform a shift of three to the left, then follow the above instructions 3 times.

Binary right shift (Divide by 2)

Performing a shift of 1 to the right

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20

1. **Start are the 128 column. Move the bit into the next column on the right.**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
	0							

2. **Move to the next column, 64 and move the bit over to the 32 column**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
	0	0						

3. **Repeat this with the remaining columns**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
	0	0	0	1				

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
	0	0	0	1	0	1	0	

4. **When you move the bit in the 1 column move the bit a space to the right. There is no space here, so we drop the bit.**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
	0	0	0	1	0	1	0	0

5. **Any gaps are filled with a zero**

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
0	0	0	0	1	0	1	0	

6. **By performing a 1-bit right shift, 20 has become 10.**

   We have divided the original number by 2.

128	64	32	16	8	4	2	1	
0	0	0	1	0	1	0	0	20
0	0	0	0	1	0	1	0	10
				8 +		2		

Exam Tip:
If the question asked you to perform a shift of two to the right, then follow the above instructions 2 times.

- **If a number is shifted 3 places to the right, it would be halved three times (i.e. divided by 2^3=8)**
- **If a number is shifted 4 places to the left it be doubled four times (i.e. multiplied by 2^4 = 16)**
- **Left shifts can cause Overflows (if extra bits are needed), and right shifts can cause bits to “drop off” the end. Bits dropping off or overflowing can lead to a loss of accuracy/data**