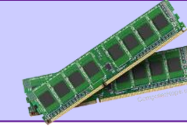
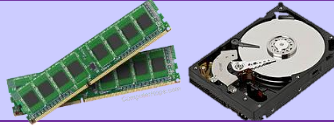
J277 - 1.2 Memory & Storage




1.2 – Memory and storage

Sub topic	Guidance
1.2.1 Primary storage (Memory)	
☐ The need for primary storage ☐ The difference between RAM and ROM ☐ The purpose of ROM in a computer system ☐ The purpose of RAM in a computer system ☐ Virtual memory	**Required** ✓ Why computers have primary storage ▪ How this usually consists of RAM and ROM ✓ Key characteristics of RAM and ROM ✓ Why virtual memory may be needed in a system ✓ How virtual memory works ▪ Transfer of data between RAM and HDD when RAM is filled
1.2.2 Secondary storage	
☐ The need for secondary storage ☐ Common types of storage: ○ Optical ○ Magnetic ○ Solid state ☐ Suitable storage devices and storage media for a given application ☐ The advantages and disadvantages of different storage devices and storage media relating to these characteristics: ○ Capacity ○ Speed ○ Portability ○ Durability ○ Reliability ○ Cost	**Required** ✓ Why computers have secondary storage ✓ Recognise a range of secondary storage devices/media ✓ Differences between each type of storage device/medium ✓ Compare advantages/disadvantages for each storage device ✓ Be able to apply their knowledge in context within scenarios **Not required** ✗ Understanding of the component parts of these types of storage

J277 - 1.2 Memory & Storage



Random Access Memory (RAM)

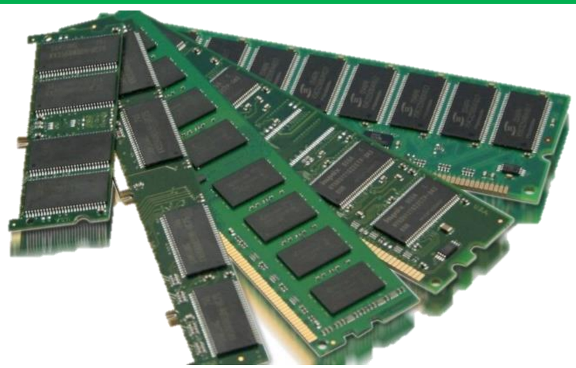
Name

Random Access Memory.

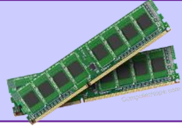
Purpose

A temporary store of instructions and data in use by the CPU for currently executing programs. Programs and data are loaded from the hard disk/solid state storage to RAM for processing.

- Volite memory.
- Can be read and written to.
- Programs and files are copied here from secondary storage while in use.
- Slower than the CPU Cache, but faster than secondary storage.



J277 - 1.2 Memory & Storage



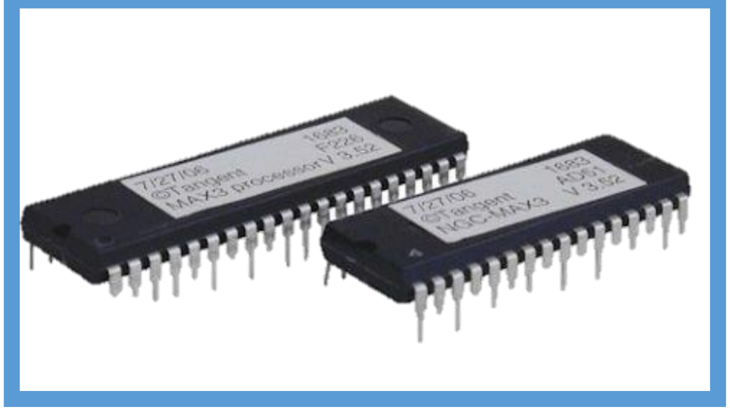
Read Only Memory (ROM)

Name

Read Only Memory.

Purpose

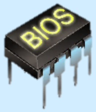
Holds the first instructions to execute when the computer is first turned on. Also known as the 'BIOS'.



Non-Volite memory.

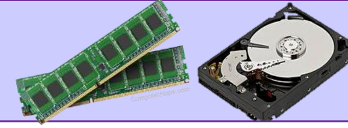
Can only be read from, not written to.

Programs and files are copied here from secondary storage while in use.



Contains BIOS (Basic Input Output System) – instructions needed for the computer to boot up.

J277 - 1.2 Memory & Storage



Virtual Memory

Name

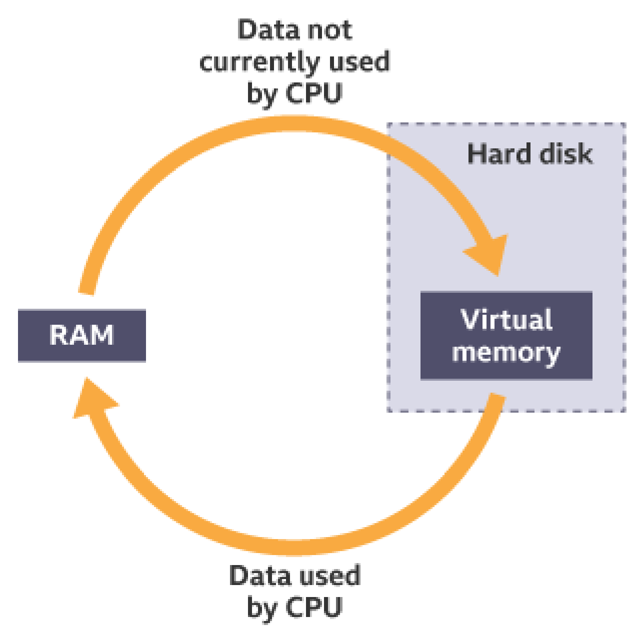
Virtual Memory.

Purpose

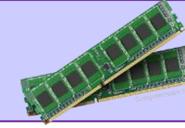
This is where part of the hard disk as if it is memory for programs that are running, when the RAM is full.

Issue

Computer instructions can only be executed from the cache and RAM. As a result, they must be transported back from virtual memory before they can be used. This makes the computer slower as it has to continually move instructions to the RAM from the virtual memory.

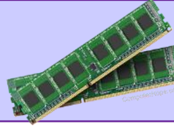


J277 - 1.2 Memory & Storage



1.2 – Memory and storage

Sub topic	Guidance
1.2.1 Primary storage (Memory)	
☐ The need for primary storage ☐ The difference between RAM and ROM ☐ The purpose of ROM in a computer system ☐ The purpose of RAM in a computer system ☐ Virtual memory	**Required** ✓ Why computers have primary storage ▪ How this usually consists of RAM and ROM ✓ Key characteristics of RAM and ROM ✓ Why virtual memory may be needed in a system ✓ How virtual memory works ▪ Transfer of data between RAM and HDD when RAM is filled
1.2.2 Secondary storage	
☐ The need for secondary storage ☐ Common types of storage: ○ Optical ○ Magnetic ○ Solid state ☐ Suitable storage devices and storage media for a given application ☐ The advantages and disadvantages of different storage devices and storage media relating to these characteristics: ○ Capacity ○ Speed ○ Portability ○ Durability ○ Reliability ○ Cost	**Required** ✓ Why computers have secondary storage ✓ Recognise a range of secondary storage devices/media ✓ Differences between each type of storage device/medium ✓ Compare advantages/disadvantages for each storage device ✓ Be able to apply their knowledge in context within scenarios **Not required** ✗ Understanding of the component parts of these types of storage




The need for secondary storage

Because main memory (RAM) is volatile, any data or programs currently being stored there will disappear once the power is lost i.e. the computer is switched off.

So secondary storage is used to retain a copy of programs and data that need to be kept long term.

Two types of Internal Storage

1. Hard Disk Drives (HDDs)

- Has moving parts.
- Stores data magnetically on metal disks.
- Can be noisy.

2. Solid State Drives (SSDs)

- No moving parts.
- Uses flash memory for faster read/write times.
- Usually quiet/silent.

Four types of External Storage

1. Flash drives & Memory cards

Solid state storage used to expand the capacity of small devices.

2. Optical Discs

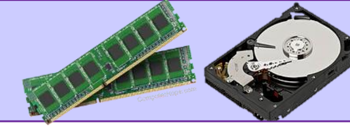
Eg. CDs. Can be read-only, write-once or rewritable.

3. Magnetic tape

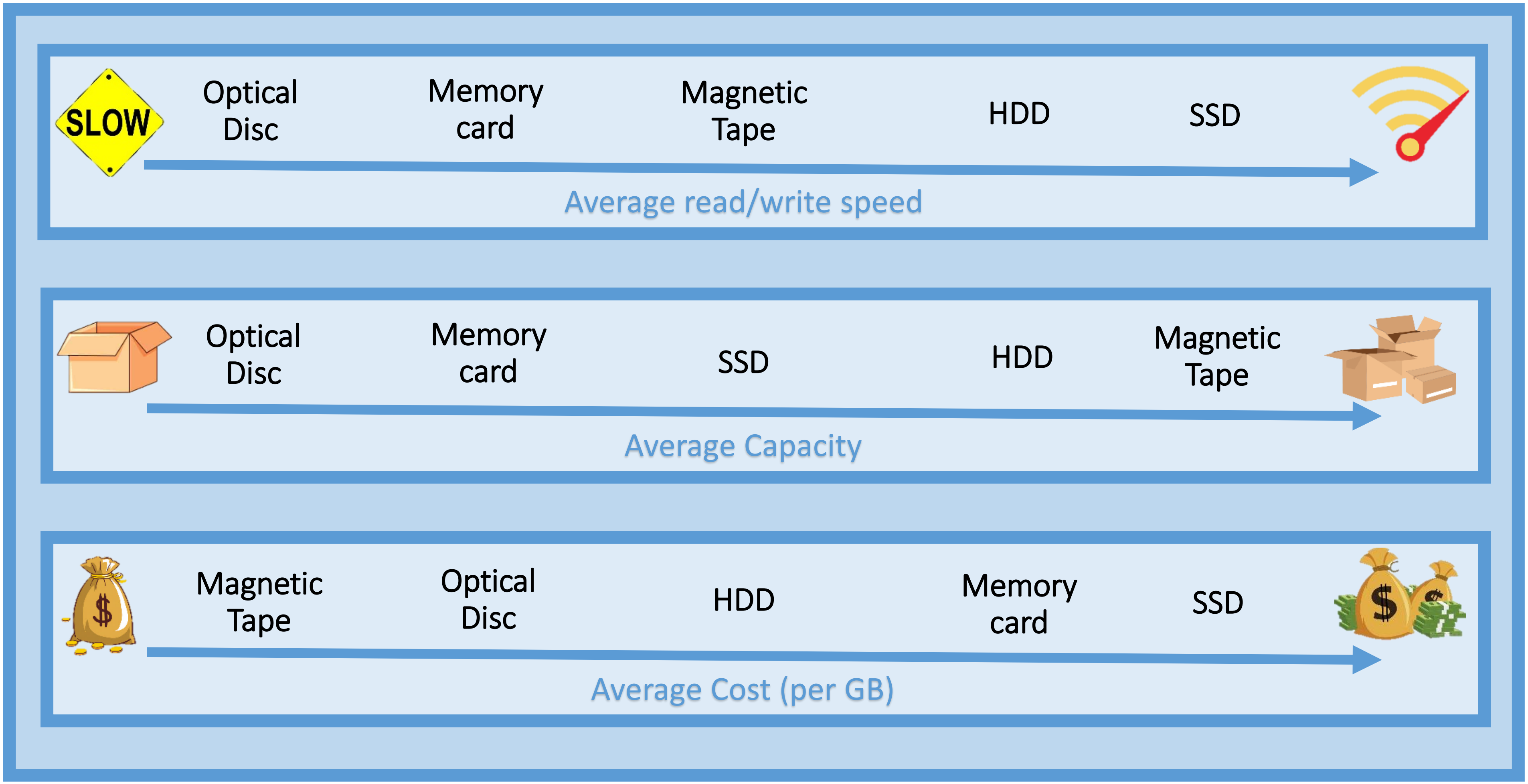
Used by organisations to store huge amounts of data.

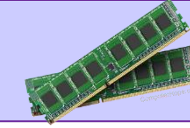
4. External HDDs & SSDs

Portable versions of internal storage. Often used for back ups.



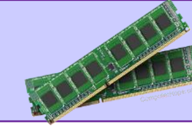
Comparing Storage Types – Characteristics





Comparing Storage Types – Characteristics

	Internal HDD	Internal SSD	Memory card	Optical Disc	Magnetic tape
Portability	**Low**	**Low**	**High**	**High**	**High**
Durability and Reliability	• Damaged by impacts. • Long read/write life.	• Shock resistant. • Limited rewrites.	• Shock resistant. • Limited rewrites.	• Easily scratched. • Limited rewrites. • Suitable long term storage.	• Damaged by impacts, heat and magnets. • Suitable long term storage.



Cloud storage

What is Cloud storage?

Storing programs and data on remote hard drives, accessed over the internet when needed.

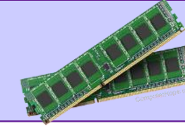
Advantages of cloud storage

1. *Space is not taken up on local drives meaning more space is available overall.*
2. *Files can be accessed anywhere, from any device connected to the internet.*
3. *Backup and version history of files is usually kept automatically.*
4. *Collaboration on files is easy.*
5. *High quality images can be kept on remote servers, with lower quality images available on low storage devices.*

Disadvantages of cloud storage

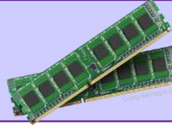
1. *Monthly/annual cost.*
2. *Potential security risks.*
3. *Relying on a third party to look after your data.*
4. *Incompatibility between storage and applications.*

J277 - 1.2 Memory & Storage



Sub topic	Guidance
1.2.3 Units	
☐ The units of data storage: ○ Bit ○ Nibble (4 bits) ○ Byte (8 bits) ○ Kilobyte (1,000 bytes or 1 KB) ○ Megabyte (1,000 KB) ○ Gigabyte (1,000 MB) ○ Terabyte (1,000 GB) ○ Petabyte (1,000 TB) ☐ How data needs to be converted into a binary format to be processed by a computer ☐ Data capacity and calculation of data capacity requirements	**Required** ✓ Why data must be stored in binary format ✓ Familiarity with data units and moving between each ✓ Calculate capacity of devices ✓ Calculate required capacity for a given set of files ✓ Calculate file sizes of sound, images and text files ▪ sound file size = sample rate x duration (s) x bit depth ▪ image file size = colour depth x image height (px) x image width (px) ▪ text file size = bits per character x number of characters **Alternatives** • Use of 1,024 for conversions and calculations would be acceptable • Allowance for metadata in calculations may be used
1.2.4 Data storage	
Numbers ☐ How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa ☐ How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur ☐ How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa ☐ How to convert binary integers to their hexadecimal equivalents and vice versa ☐ Binary shifts	**Required** ✓ Denary number range 0 – 255 ✓ Hexadecimal range 00 – FF ✓ Binary number range 00000000 – 11111111 ✓ Understanding of the terms most significant bit, and least significant bit ✓ Conversion of any number in these ranges to another number base ✓ Ability to deal with binary numbers containing between 1 and 8 bits ▪ e.g. 11010 is the same as 00011010 ✓ Understand the effect of a binary shift (both left or right) on a number

J277 - 1.2 Memory & Storage




Bit, nibble, byte, kilobyte & megabyte

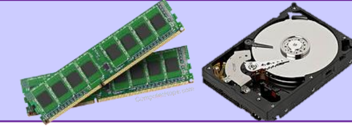
Traditionally, each unit is defined to be 1024 times bigger than the previous one!

Explain how binary data is used to represent numbers in a computer.

- Computers can only store and process binary data.
- They use 1s and 0s to represent the flow of electricity – A 1 shows that electricity is flowing and 0 shows that it isn't.
- Each 1 or 0 in binary data is a **bit** (**b**inary dig**it**).

Name	**Size**
Bit	A single binary digit (0 or 1)
Nibble	4 bits
Byte (B)	8 bits
Kilobyte (KB)	1000 bytes
Megabyte (MB)	1000 kilobytes
Gigabyte (GB)	1000 megabytes
Terabyte (TB)	1000 gigabytes
Petabyte (PB)	1000 terabytes

J277 - 1.2 Memory & Storage



Units of data storage

Unit conversions:

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

Q1. Jenny has 700 high resolution 10Mb photographs.
What size memory card in GB will she need as a minimum to store these images?

7

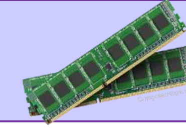
Q2. An SD card stores films in a compressed format. Each film is 6.7GB. How many films can a 64GB card store?

9

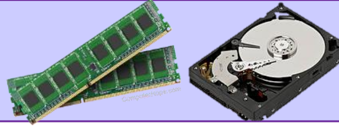
Q3. A CCTV system records video at 180MB per second. How much storage is required in GB for 90 second of footage?

16.2

J277 - 1.2 Memory & Storage




Sub topic	Guidance
1.2.3 Units	
☐ The units of data storage: ○ Bit ○ Nibble (4 bits) ○ Byte (8 bits) ○ Kilobyte (1,000 bytes or 1 KB) ○ Megabyte (1,000 KB) ○ Gigabyte (1,000 MB) ○ Terabyte (1,000 GB) ○ Petabyte (1,000 TB) ☐ How data needs to be converted into a binary format to be processed by a computer ☐ Data capacity and calculation of data capacity requirements	**Required** ✓ Why data must be stored in binary format ✓ Familiarity with data units and moving between each ✓ Calculate capacity of devices ✓ Calculate required capacity for a given set of files ✓ Calculate file sizes of sound, images and text files ▪ sound file size = sample rate x duration (s) x bit depth ▪ image file size = colour depth x image height (px) x image width (px) ▪ text file size = bits per character x number of characters **Alternatives** • Use of 1,024 for conversions and calculations would be acceptable • Allowance for metadata in calculations may be used
1.2.4 Data storage	
Numbers ☐ How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa ☐ How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur ☐ How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa ☐ How to convert binary integers to their hexadecimal equivalents and vice versa ☐ Binary shifts	**Required** ✓ Denary number range 0 – 255 ✓ Hexadecimal range 00 – FF ✓ Binary number range 00000000 – 11111111 ✓ Understanding of the terms most significant bit, and least significant bit ✓ Conversion of any number in these ranges to another number base ✓ Ability to deal with binary numbers containing between 1 and 8 bits ▪ e.g. 11010 is the same as 00011010 ✓ Understand the effect of a binary shift (both left or right) on a number



How to convert positive denary whole numbers (0-255) into 8 bit binary numbers

Converting 183 into Binary involves the following steps:

- 1. Does 128 go into 183? Yes so put a 1 Under 128. This leaves 55.

128	64	32	16	8	4	2	1
1							

- 2. Does 64 go into 55? No so put a 0 under 64. This leaves 55 still.

128	64	32	16	8	4	2	1
1	0						

- 3. Does 32 go into 55? Yes so put a 1 under 32. The leaves 23.

128	64	32	16	8	4	2	1
1	0	1					

- 4. Does 16 go into 23?. Yes so put a 1 under 16. This leaves 7.

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

- 5. Does 8 go into 7? No so put a 0 under 8. This still leaves 7.

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

- 6. Does 4 go into 7? Yes so put a 1 under 4. This leaves 3.

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

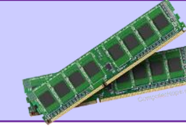
- 7. Does 2 go into 3? Yes so put a 1 under 2. This leaves 1.

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

- 8. Does 1 go into 1? Yes so put a 1 under 1.

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

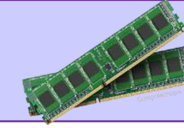
J277 - 1.2 Memory & Storage




Convert these denary numbers into their 8 bit binary equivalent:

1. 20
2. 54
3. 125
4. 2
5. 99
6. 47

1. 00010100
2. 00110110
3. 01111101
4. 00000010
5. 01100011
6. 00101111



Adding two 8 bit binary integers and overflow errors.

Binary numbers can easily be added using a few addition rules. These are:

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10
1 + 1 + 1 = 11

Examples

1. 00000101 + 00000110 = 00001011
2. 00001110 + 00000101 = 00010011
3. 00001001 + 00000011 = 00001100
4. 00000111 + 00001011 = 00010010
5. 00011011 + 00001010 = 00100001

Overflow errors can occur when adding binary numbers because:

There are not enough bits to hold the answer and a carry remains that cannot be stored.

How to convert 8 bit binary numbers to positive denary whole numbers

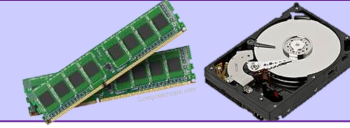
The binary number 10110111 can be converted to Denary as:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	1	1	0	1	1	1

If a '1' is present in that column, then you need to add that number to the others…

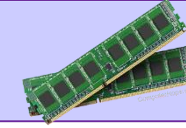
This means you get: 128 + 0 + 32 + 16 + 0 + 4 + 2 + 1 = 183

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	1	1	0	1	1	1



Convert these binary numbers into their denary equivalent:

1. 00000000 = 0
2. 00000001 = 1
3. 00001111 = 15
4. 00010010 = 18
5. 00101000 = 40
6. 01010101 = 85
7. 10000000 = 128
8. 11001100 = 204
9. 11100011 = 227
10. 11111111 = 255



How to convert from binary to hexadecimal equivalents and vice versa.

To convert 4C from Hexadecimal into base 10:

STEP 1 – split the Hex up and find the binary representation:

4	C
0100	1100

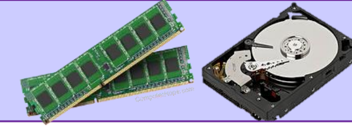
STEP 2 – Put the binary number together, then convert to denary:

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

01001100 = 64+8+4 = 76

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

J277 - 1.2 Memory & Storage



How to convert from hexadecimal to binary and vice versa.

We can convert from to Hexadecimal by using the following method:

e.g. Convert 167 into Hexadecimal

STEP 1 – Convert the number into binary:

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

STEP 2 – Split the binary number into two nibbles:

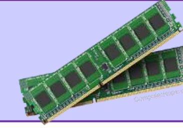
1010 0111

STEP 3 – Convert the two nibbles to Hex:

1010 = A 0111 = 7

Therefore the answer is = **A7**

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111




Binary shifts

A left shift performs a: multiplication.

Question 2:

Number	0	0	0	0	0	1	0	1	Left shift	1
Answer	**0**	**0**	**0**	**0**	**1**	**0**	**1**	**0**		

Overflow **0**

Correct? Y

Number 5 becomes 10

A right shift performs a: division.

Question 2:

Number	0	0	0	0	1	1	0	0	Right shift	1
Answer	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**0**		

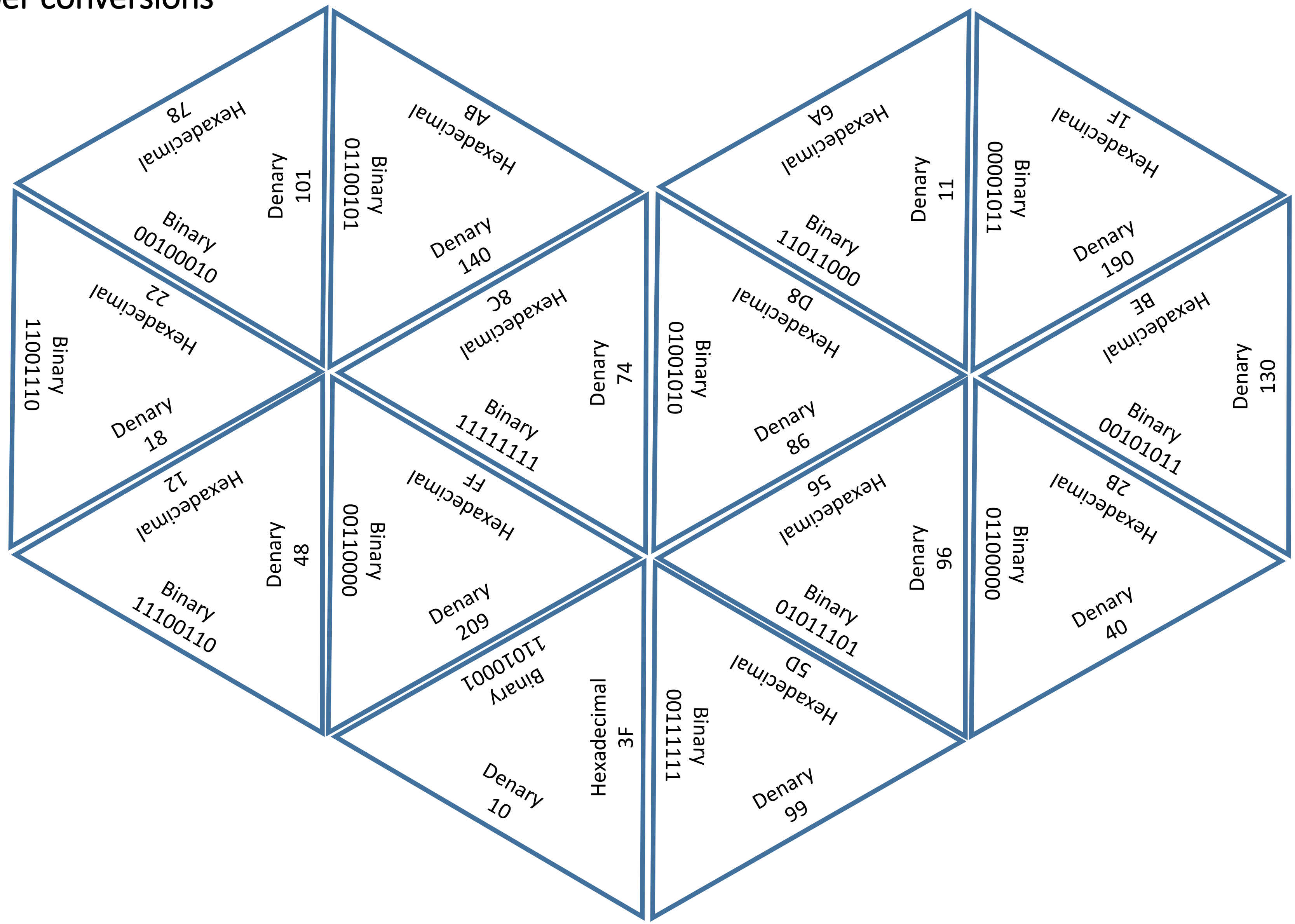
0 Underflow

Correct? Y

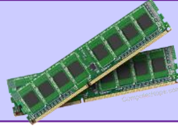
Number 12 becomes 6

J277 - 1.2 Memory & Storage

Number conversions

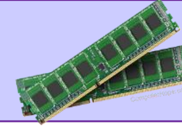


J277 - 1.2 Memory & Storage




Sub topic	Guidance
Characters ☐ The use of binary codes to represent characters ☐ The term 'character set' ☐ The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ○ ASCII ○ Unicode	**Required** ✓ How characters are represented in binary ✓ How the number of characters stored is limited by the bits available ✓ The differences between and impact of each character set ✓ Understand how character sets are logically ordered, e.g. the code for 'B' will be one more than the code for 'A' ✓ Binary representation of ASCII in the exam will use 8 bits **Not required** ✗ Memorisation of character set codes
Images ☐ How an image is represented as a series of pixels, represented in binary ☐ Metadata ☐ The effect of colour depth and resolution on: ○ The quality of the image ○ The size of an image file	**Required** ✓ Each pixel has a specific colour, represented by a specific code ✓ The effect on image size and quality when changing colour depth and resolution ✓ Metadata stores additional image information (e.g. height, width, etc.)
Sound ☐ How sound can be sampled and stored in digital form ☐ The effect of sample rate, duration and bit depth on: ○ The playback quality ○ The size of a sound file	**Required** ✓ Analogue sounds must be stored in binary ✓ Sample rate – measured in Hertz (Hz) ✓ Duration – how many seconds of audio the sound file contains ✓ Bit depth – number of bits available to store each sample (e.g. 16-bit)
1.2.5 Compression	
☐ The need for compression ☐ Types of compression: ○ Lossy ○ Lossless	**Required** ✓ Common scenarios where compression may be needed ✓ Advantages and disadvantages of each type of compression ✓ Effects on the file for each type of compression **Not required** ✗ Ability to carry out specific compression algorithms

J277 - 1.2 Memory & Storage



The use of binary codes to represent characters

Characters
These are uppercase and lowercase letters, the digits 0-9 and symbols like ?, + and £.

Character Sets
These are collections if characters that a computer recognises from their binary representation, used to convert characters to binary code and vice versa.

Button pressed on keyboard.

Binary signal sent to computer.

Computer translates code using character set.

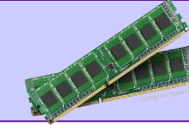
Two important character sets

ASCII
- Each character is given a 7-bit binary code – so ASCII can represent 128 different characters.
- An extra bit (0) is added to the start of each binary code so each character uses 1 byte.
- The codes for numbers and letters are ordered (A comes before B before C…).

UNICODE
- Covers all major language, including ones that use different alphabets, like Greek, Russian and Chinese.
- Uses multiple bytes for each character.
- The first 128 characters in Unicode are the same as ASCII.

Character	Binary	Denary
A	0100 0001	65
B	0100 0010	66
C	0100 0011	67
a	0110 0001	97
b	0100 0010	98
c	0110 0011	99

J277 - 1.2 Memory & Storage




Text File Sizes

Use this formula to calculate file size:

File size (in bits)

= number of bits per character * number of characters

Example

How many bits would be needed to store “I’m a string, store me!” in 8 bit ASCII?

#Count the characters

I	‘	m		a		s	t	r	i	n	g	,		s	t	o	r	e		m	e	!

5 10 15 20 23

#Use the formula

File size = 8 x 23 = ***184 bits***

How many bits would be needed to store “My phone number is 07584 587458!” in 8 bit ASCII? Write your answer in Bytes.

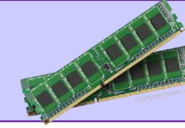
#Count the characters

32

#Use the formula

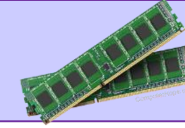
File size = 8 x 32 = ***32 bytes.***

J277 - 1.2 Memory & Storage




Sub topic	Guidance
Characters ☐ The use of binary codes to represent characters ☐ The term 'character set' ☐ The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ○ ASCII ○ Unicode	**Required** ✓ How characters are represented in binary ✓ How the number of characters stored is limited by the bits available ✓ The differences between and impact of each character set ✓ Understand how character sets are logically ordered, e.g. the code for 'B' will be one more than the code for 'A' ✓ Binary representation of ASCII in the exam will use 8 bits **Not required** ✗ Memorisation of character set codes
Images ☐ How an image is represented as a series of pixels, represented in binary ☐ Metadata ☐ The effect of colour depth and resolution on: ○ The quality of the image ○ The size of an image file	**Required** ✓ Each pixel has a specific colour, represented by a specific code ✓ The effect on image size and quality when changing colour depth and resolution ✓ Metadata stores additional image information (e.g. height, width, etc.)
Sound ☐ How sound can be sampled and stored in digital form ☐ The effect of sample rate, duration and bit depth on: ○ The playback quality ○ The size of a sound file	**Required** ✓ Analogue sounds must be stored in binary ✓ Sample rate – measured in Hertz (Hz) ✓ Duration – how many seconds of audio the sound file contains ✓ Bit depth – number of bits available to store each sample (e.g. 16-bit)
1.2.5 Compression	
☐ The need for compression ☐ Types of compression: ○ Lossy ○ Lossless	**Required** ✓ Common scenarios where compression may be needed ✓ Advantages and disadvantages of each type of compression ✓ Effects on the file for each type of compression **Not required** ✗ Ability to carry out specific compression algorithms

J277 - 1.2 Memory & Storage



The use of binary codes to represent characters

Bitmap
A type of image made up from lots of tiny dots, called **pixels.** The colour of each pixel is stored using binary code.

Image resolution
The number of pixels in a bitmap image. Often given as 'width x height.'

Metadata

Data stored in a file which contains information ab out the file. Helps the computer to recreate the image on screen from the binary data in each pixel.

Examples of Metadata include:

- Height and Width
- Colour depth
- Resolution
- File format
- Data created
- Date last edited

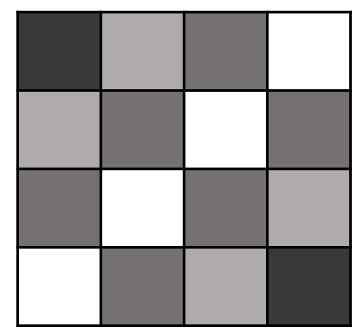
Colour Depth
The number of bits used to represent each pixel. The number of colours that can be used for a given colour depth follows this formula:

$$2^n \quad \text{(where n = colour depth)}$$

Example
Colour depth = 2 bits

- Number of colours = 2^2 = 4
- In this example: 00 = white, 01 = light grey, 10 dark grey & 11 = black.

11	01	10	00
01	10	00	10
10	00	10	01
00	10	01	11



J277 - 1.2 Memory & Storage

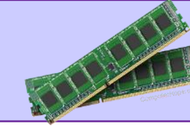


Image File Sizes

Use this formula to calculate file size:

File size (in bits)

= image resolution x colour depth
= width x height x colour depth

Increasing the image resolution or colour depth will usually give a high quality image, but larger file size.

Example

Calculate the size in kB of a 100x100 pixel image with a colour depth of 16 bits.

#Calculating the size
File size = 100x100x16
= 160,000 bits

#Calculating in KB
160,000 bits
= 160,000 / 8 = 20,000 bytes
= 20, 000 / 1000 = 20 KB

Example

Calculate the size in MB of a 800 pixels (width) × 600 pixels (height) with a colour depth of 24 bits.

#Calculating the size
File size = 800x600x24
= 11,520,000 bits

#Calculating in MB
11,520,000 bits
= 11,520,000 bits / 8 = 1,440,000 bytes
= 1,440,000 / 1000 = 1,440 KB
1,400KB / 1000 = 1.44MB

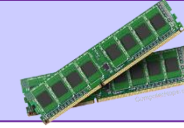
Example

Calculate the size in MB of a 750 pixels (width) × 900 pixels (height) with a colour depth of 32 bits.

#Calculating the size
File size = 750x900x32
= 21,600,000 bits

#Calculating in GB
21,600,000 bits
= 21,600,000 bits / 8 = 2,700,000 bytes
= 2,700,000 / 1000 = 2,700 KB
2,700 KB / 1000 = 27MB

J277 - 1.2 Memory & Storage




Sub topic	Guidance

Characters
☐ The use of binary codes to represent characters
☐ The term 'character set'
☐ The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.:
  ○ ASCII
  ○ Unicode
	Required
	✓ How characters are represented in binary
	✓ How the number of characters stored is limited by the bits available
	✓ The differences between and impact of each character set
	✓ Understand how character sets are logically ordered, e.g. the code for 'B' will be one more than the code for 'A'
	✓ Binary representation of ASCII in the exam will use 8 bits
	Not required
	✗ Memorisation of character set codes

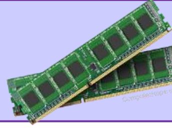
Images
☐ How an image is represented as a series of pixels, represented in binary
☐ Metadata
☐ The effect of colour depth and resolution on:
  ○ The quality of the image
  ○ The size of an image file
	Required
	✓ Each pixel has a specific colour, represented by a specific code
	✓ The effect on image size and quality when changing colour depth and resolution
	✓ Metadata stores additional image information (e.g. height, width, etc.)

Sound
☐ How sound can be sampled and stored in digital form
☐ The effect of sample rate, duration and bit depth on:
  ○ The playback quality
  ○ The size of a sound file
	Required
	✓ Analogue sounds must be stored in binary
	✓ Sample rate – measured in Hertz (Hz)
	✓ Duration – how many seconds of audio the sound file contains
	✓ Bit depth – number of bits available to store each sample (e.g. 16-bit)

1.2.5 Compression	

☐ The need for compression
☐ Types of compression:
  ○ Lossy
  ○ Lossless
	Required
	✓ Common scenarios where compression may be needed
	✓ Advantages and disadvantages of each type of compression
	✓ Effects on the file for each type of compression
	Not required
	✗ Ability to carry out specific compression algorithms

J277 - 1.2 Memory & Storage



Sampling

Key definitions

We hear sound as ‘analogue’ – tones and rhythms blend together smoothly.

If sound has to be handled by the computer – needs to be converted to binary.

‘Digitalise’ using an input device.

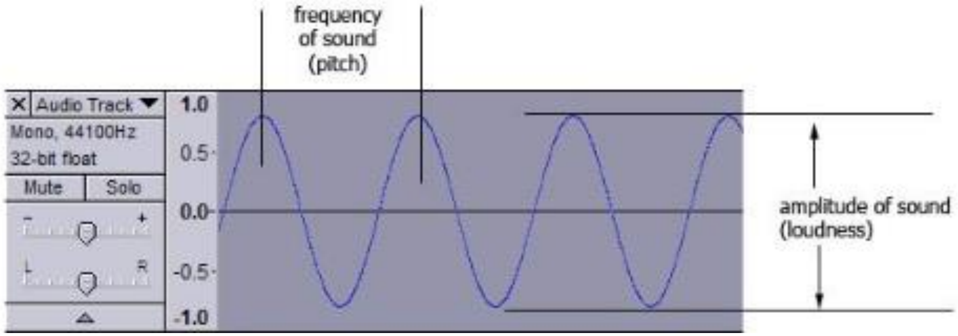
Sampling – Converting analogue sound wave into digital data that can be read and stored by a computer.

Sample frequency – The number of audio samples captured every second. Measured in Hertz (Hz)

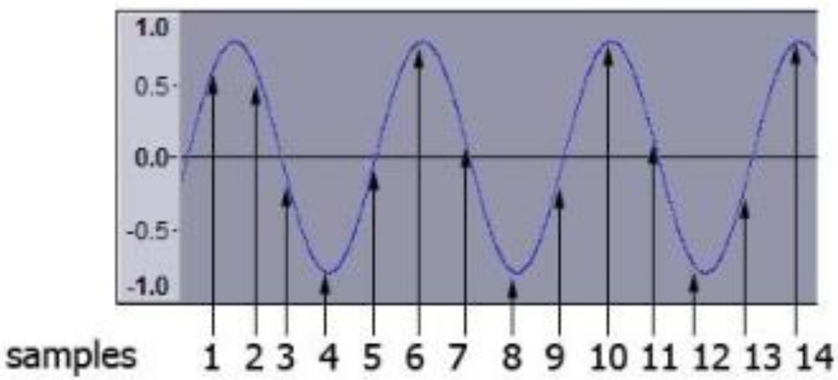
Bit depth – Number of bits available for each sample.

Sound Sampling process

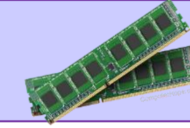
Typical sound wave



The height of the wave is measured (sampled) at regular intervals.
This is then turned into binary and stored.



J277 - 1.2 Memory & Storage




Sound File Sizes

Use this formula to calculate file size:

File size (in bits)

= Sample rate (in Hz) x bit depth x length (in Secs)

A higher sample rate or bit depth will give a higher quality sound file, but will increase the file size.

Example

Calculate the file size in MB of a 50 second audio recording with a sample rate of 40kHz and a bit depth of 8 bits.

#Calculating the size
File size = 40kHz = 40,000 Hz
=50x40,000x8
= 16,000,000 bits

#Calculating in MB
16,000,000 bits
= 16,000,000 / 8 = 2,000,000 bytes
= 2,000,000 / 1000 = 2000 KB
= 2000 KB / 1000 = 2GB

Example

Calculate the file size in MB for an audio quality recording with a 16-bit depth, sample rate of 44,100 Hz (samples per second) and duration of 3 minutes.

#Calculating the size
File size = 16x44,100x180
= 127,008,000 bits

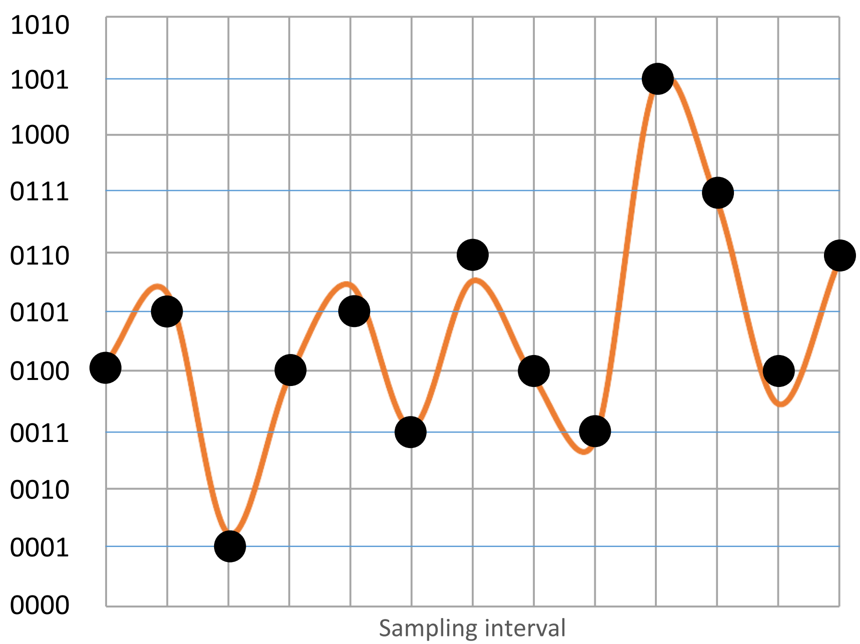
#Calculating in MB
127,008,000 bits
= 127,008,000 / 8 = 15,876,000 bytes
= 15,876,000 / 1000 = 15,876 KB
15,876 / 1000 = 15.876 MB

Example

Calculate the file size in MB for an audio quality recording with a 24-bit depth, sample rate of 48kHz (samples per second) and duration of 5 minutes.

#Calculating the size
File size = 48kHz = 48,000 Hz
= 5 mins = 300 secs
24x48,000x300
= 345,600,000 bits
#Calculating in MB
= 345,600,000 / 8 = 43,200,000 bytes
= 43,200,000 / 1000 = 43,200 KB
43,200KB / 1000 = 43.2 MB

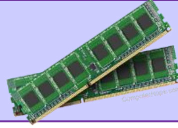
The effect of doubling the bit rate on the quality of the sound and file size



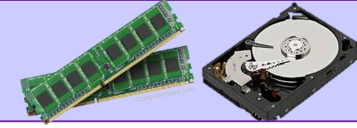
Binary for the sound could be: 0100 0101 0001 0100 0101 0011 0110 0100 0011 1001 0111 0100 0110 — 52 bits.

The quality has: Increased because the digital value is closer to the original wave.

J277 - 1.2 Memory & Storage




Sub topic	Guidance
Characters ☐ The use of binary codes to represent characters ☐ The term 'character set' ☐ The relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.: ○ ASCII ○ Unicode	**Required** ✓ How characters are represented in binary ✓ How the number of characters stored is limited by the bits available ✓ The differences between and impact of each character set ✓ Understand how character sets are logically ordered, e.g. the code for 'B' will be one more than the code for 'A' ✓ Binary representation of ASCII in the exam will use 8 bits **Not required** ✗ Memorisation of character set codes
Images ☐ How an image is represented as a series of pixels, represented in binary ☐ Metadata ☐ The effect of colour depth and resolution on: ○ The quality of the image ○ The size of an image file	**Required** ✓ Each pixel has a specific colour, represented by a specific code ✓ The effect on image size and quality when changing colour depth and resolution ✓ Metadata stores additional image information (e.g. height, width, etc.)
Sound ☐ How sound can be sampled and stored in digital form ☐ The effect of sample rate, duration and bit depth on: ○ The playback quality ○ The size of a sound file	**Required** ✓ Analogue sounds must be stored in binary ✓ Sample rate – measured in Hertz (Hz) ✓ Duration – how many seconds of audio the sound file contains ✓ Bit depth – number of bits available to store each sample (e.g. 16-bit)
1.2.5 Compression	
☐ The need for compression ☐ Types of compression: ○ Lossy ○ Lossless	**Required** ✓ Common scenarios where compression may be needed ✓ Advantages and disadvantages of each type of compression ✓ Effects on the file for each type of compression **Not required** ✗ Ability to carry out specific compression algorithms



Data Compression

What is Data Compression?

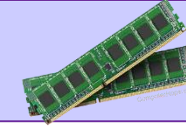
Making file sizes smaller, whilst trying to stay as true to the original as possible.

*Less **storage** space used.*

*Streaming/Downloading compressed files takes less **bandwidth**.*

Some services (eg. Email) have file size limits, compression can get below limits.

	Pros	Cons
<u>Lossy</u> Permanently removes data from the file.	• Small file sizes. • Ideal for web use due to file size. • Commonly used – Lots of software and read lossy files.	• Looses data – Can't be reversed to the original. • Quality degrades. • Can't be used on text / software.
<u>Lossless</u> Temporarily removes data to store the file and restores it to the original state when opened.	• No loss in quality. • File can be turned back to the original. • Can be used of text / software.	• Larger than lossy files.



Units of data storage

Case study:

Elizabeth is looking to upgrade her phone. She is choosing between model A and model B.

	Model A	**Model B**
Storage capacity	64GB	256GB
Camera	8 mega-pixel, 24bit colour	12 mega-pixel, 24 bit colour
Video	1080p (2 mega-pixel) at 24 fps, 30 fps, or 60 fps	4K HD (8 mega-pixel) at 24 fps, 30 fps, or 60 fps

Consider that 32GB of data storage will be used for apps. Assume 50% compression.

Maximum number of photographs that could be stored on the phone:	8,000,000 bit resolution. x 24 bit = 192,000,000 bits. Divide by 8 for bytes = 24,000,000 bytes or 24MB Per photograph. 32,000 (32GB) divided by 24 = 1,333 uncompressed photos. x 2 due to 50% compression = 2,666 photos.	12,000,000 bit resolution. X 24 bit = 288.000.000 bits. Divide by 8 for bytes = 36,000,000 bytes or 36MB Per photograph. 224,000 (224GB) divided by 36 = 6,222 uncompressed photos. x 2 due to 50% compression = 12,444 photos.