Cambridge IGCSE · Computer Science 0478 · Notes

Text, sound and images

Read online at **pastopic.com/cambridge/computer-science-0478/notes/text-sound-and-images**

Last checked 30 September 2026

Read first: **Number systems**

What you need to know

By the end of this topic you should be able to:

1. **Explain how and why a computer represents text.** Text has to be turned into binary before a computer can process it. A **character set** gives every character its own binary code. Know the two character sets in the syllabus, **ASCII** and **Unicode**: Unicode can represent far more characters and symbols than ASCII (including other languages and emojis), but it needs more bits for each character.
2. **Explain how and why a computer represents sound.** A sound wave is **sampled** and each sample is turned into binary so the computer can process it. The **sample rate** is the number of samples taken each second, and the **sample resolution** is the number of bits used for each sample. Raising either one makes the recording more accurate and the file bigger.
3. **Explain how and why a computer represents an image.** An image is made of **pixels**, and each pixel's colour is turned into binary. The **resolution** is the number of pixels in the image, and the **colour depth** is the number of bits used for each colour. Raising either one makes the image better quality and the file bigger.

Working out a file size in bytes, KiB or MiB, and compressing files, are in the next topic, Data storage and compression. Exam questions often join the two, so this note shows the file size sums too.

Understand it

Why everything becomes binary

A computer is built from switches and logic gates that have only two states, on and off, so it can only store and process **binary** (1s and 0s). Letters, sounds and pictures are not numbers, so each needs a rule for turning it into binary. The three rules below are that step for text, sound and images.

Text: character sets

A **character set** is a list of every character the computer can use (letters, digits, punctuation, spaces and symbols), each with its own **unique** binary code. When you press a key, the computer looks up that character's code and stores the code. A word is stored as the codes of its letters, one after another, in order.

In **ASCII** the capital letters run in alphabetical order from A = 65, and the small letters from a = 97, so each small letter is exactly 32 more than its capital. Because the codes are in order, you can work out one code from another: if A is 65, then C is 67. The digit character "7" has its own code (55), which is not the number 7.

The word "Cat" is stored as three codes, one for each character:

Character	**C**	**a**	**t**
ASCII (denary)	67	97	116
Stored (binary)	01000011	01100001	01110100

To turn a code into binary or hexadecimal, use the methods in the Number systems note (see Read first).

ASCII uses 7 bits per character, which gives $2^7 = 128$ characters: enough for English letters, digits and punctuation. An extended 8-bit version gives 256. **Unicode** uses more bits per character (up to 32 bits, 4 bytes), so it has room for over a million characters: letters from many different languages and writing systems, many more symbols, and emojis. The first 128 Unicode codes are the same as ASCII.

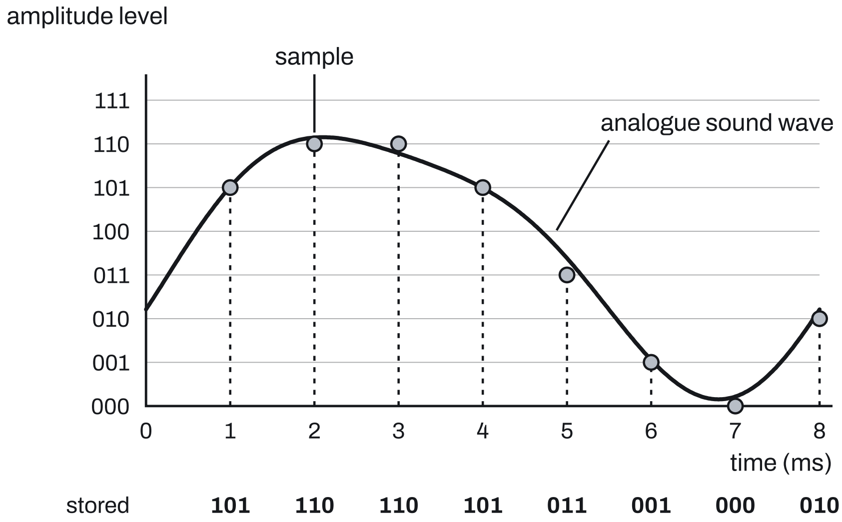
	ASCII	**Unicode**
Bits per character	7 (8 in extended ASCII)	more: up to 32
Characters	128 (256 extended)	far more, including other languages and emojis
Storage for the same text	less	more

The trade-off is simple: Unicode can show more characters, but every character takes more bits, so the same text takes up more storage space and takes longer to send.

Sound: sampling

Sound is an **analogue** wave: its height (**amplitude**) changes smoothly all the time. A computer cannot store a smooth wave, so it measures it.

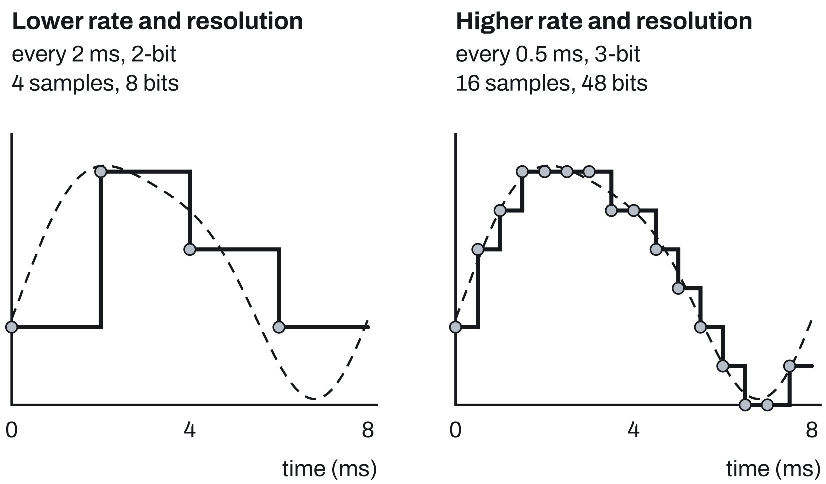
1. A microphone picks up the sound wave.
2. At regular time intervals, the height (amplitude) of the wave is measured. Each measurement is a **sample**.
3. Each amplitude is rounded to the nearest value that the number of bits can hold, and stored as that binary number.
4. The samples are stored in order, and played back one after another to rebuild the sound.



Two settings decide how good the recording is:

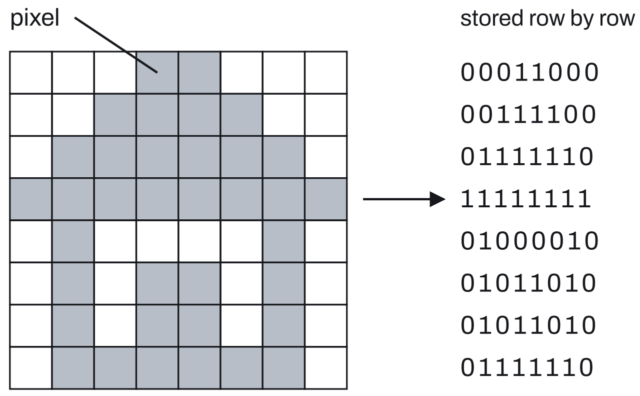
- **Sample rate**: the number of samples taken **per second**, measured in hertz (Hz). The diagram takes one sample every millisecond, which is 1000 samples per second (1000 Hz). Music is often recorded at 44 100 Hz.
- **Sample resolution** (also called bit depth): the number of **bits used for each sample**. With n bits there are 2^n different amplitude values: the diagram's 3 bits give $2^3 = 8$ levels; 8 bits give 256; 16 bits give 65 536.

A **higher sample rate** measures the wave more often, so fewer changes are missed between samples. A **higher sample resolution** gives more amplitude levels, so each measurement is rounded less and is more precise, and a wider range of amplitudes (quieter and louder sounds) can be recorded. Either way the recording is **more accurate**, closer to the original sound. The price is size: more samples, or more bits per sample, means **more bits to store**, so the file is **larger**.



Images: pixels

A digital (bitmap) image is a grid of tiny squares called **pixels**. A pixel is the smallest part of the image, and it shows **one colour**. Each colour has its own binary code, and the image is stored as the code for each pixel's colour, in order, row by row.



shaded = 1, blank = 0

8 × 8 = 64 pixels × 1 bit = 64 bits = 8 bytes

- **Resolution** is the **number of pixels** in the image, usually given as width × height. A 1920 × 1080 image has $1920 \times 1080 = 2\,073\,600$ pixels. More pixels means more detail, so a sharper image, but more pixels to store.
- **Colour depth** is the **number of bits used to represent each colour** (each pixel's colour). With n bits there are 2^n possible colours:

Colour depth	1-bit	2-bit	8-bit	16-bit	24-bit	32-bit
Possible colours	2	4	256	65 536	16 777 216	4 294 967 296

A higher colour depth means a **wider range of colours** can be used, so the image looks closer to real life, but every pixel needs more bits. The house above in 24-bit colour would need $64 \times 24 = 1536$ bits, 192 bytes, instead of 8 bytes.

So, like sound: **higher resolution or higher colour depth → better quality and a larger file.** Lower either one and the file gets smaller, but the image loses detail (it may look blocky) or has fewer colours.

An image file also stores **metadata**: data about the image, kept with it in the file. It tells the computer how to display the image (its width and height in pixels and its colour depth) and can hold other facts such as the date and time the photo was taken. JPEG, PNG, GIF and BMP are common image file formats.

Where this meets file size and compression

Every rule above ends with the same idea: more bits for better quality. That is why the file size sums and compression (next topic) use exactly these settings:

- image file size in bits = resolution (width × height) × colour depth;
- sound file size in bits = sample rate × sample resolution × length in seconds.

Divide by 8 to get bytes, then by 1024 for each step up to KiB, MiB, GiB. **Lossy** compression makes images and sound files smaller by permanently lowering these settings (fewer colours, a lower resolution or sample rate) or removing detail people barely notice; JPEG and MP3 files are lossy. **Lossless** compression keeps every bit of the original, so it can be rebuilt exactly.

Key facts and formulas

There is no formula list for this subject, so you must learn all of these.

Formula	Status
Character set: every character has a unique binary code; text is stored as the codes of its characters, in order	Learn it
ASCII: 7 bits per character, 128 characters (extended ASCII: 8 bits, 256); A = 65, a = 97, codes in alphabetical order	Learn it
Unicode: more bits per character (up to 32), far more characters (other languages, symbols, emojis), so more storage	Learn it
Sample rate = number of samples per second (Hz); sample resolution = number of bits per sample	Learn it
Resolution = number of pixels (width × height); colour depth = number of bits per colour	Learn it
n bits give 2^n different values (colours, or amplitude levels)	Learn it
Higher sample rate / sample resolution / resolution / colour depth → more accurate or better quality, and a larger file	Learn it
Image file size (bits) $=$ width $\times$ height $\times$ colour depth	Learn it
Sound file size (bits) $=$ sample rate $\times$ sample resolution $\times$ seconds	Learn it
1 nibble = 4 bits; 1 byte = 8 bits; 1 KiB = 1024 bytes; 1 MiB = 1024 KiB; 1 GiB = 1024 MiB; 1 TiB = 1024 GiB	Learn it

How to do it

In the Paper 1 questions we have tagged for this topic (21 questions, 2021–2025), the types came up this often. 10 of the 21 questions mix types, so the counts add up to more than 21. Parts about other topics in the same questions (packets, storage devices, compression methods) are not counted.

Question type	Questions
Define or complete image terms	7
Define sound terms, or explain how sound is converted to binary	5
Effect of changing resolution or colour depth	4
Explain how text is converted to binary	4

Question type	Questions
Calculate an image or sound file size	4
Compare ASCII and Unicode	3
Effect of sample rate and sample resolution	3
Work with character codes	2
Describe how an image is stored; image file formats	2
Explain why data is converted to binary	1

1. Define or complete image terms (7 questions)

Often a table with a term or a description missing. Learn one precise sentence for each term:

- **pixel**: the smallest part (component) of an image; one square of one colour.
- **resolution**: the number of pixels in the image, or the number of pixels wide × the number of pixels high.
- **colour depth** (bit depth): the number of bits used to represent each colour in the image.
- **metadata**: data stored with the image about the image, such as its width, height, colour depth, and the date and time it was taken.

Variations you will meet:

- **"State what is meant by a 16-bit (or 32-bit) colour depth"**: 16 bits are used to represent each colour (each pixel). You can add that this gives $2^{16} = 65\,536$ possible colours.
- **"Give one benefit of increasing the colour depth"**: a wider range of colours can be used, so the image has more detail and looks closer to real life.
- A mixed table can also ask for **character set** (stores every character with its unique binary code) or **sample rate**, so learn the sound and text terms too.

2. Define sound terms, or explain how sound is converted to binary (5 questions)

- **sampling**: measuring the height (amplitude) of the sound wave at regular time intervals.
- **sample rate**: the number of samples taken per second.
- **sample resolution**: the number of bits used for each sample, which sets how many different amplitudes can be stored (for example 16-bit).

"Explain how the analogue sound is recorded and converted to digital" (up to 5 marks). Give a chain of separate points:

1. The sound is recorded with a microphone.
2. The sound wave is sampled: its height (amplitude) is measured at regular intervals.
3. The sample rate is set: the number of samples taken each second.
4. The sample resolution is set: the number of bits used for each sample.

5. Each sample's amplitude is given a binary value and stored.

Don't describe sample resolution in terms of pixels or images; examiners saw that mix-up often.

3. Effect of changing resolution or colour depth (4 questions)

1. Say which way the **file size** goes: higher resolution or colour depth → larger; lower → smaller.
2. Give the **reason**: more (or fewer) pixels to store, or more (or fewer) bits for each colour, so more (or fewer) bits in total.
3. If asked about quality: more pixels → more detail; more bits per colour → more colours available.

For example, "a 1000 × 500 image is changed to 2000 × 1000": the number of pixels is 4 times bigger, so the file size increases (it is 4 times as big at the same colour depth). "Describe the relationship" (2 marks) wants the link in steps: the higher the resolution, the more pixels; the more pixels, the more bits to store; so the larger the file.

4. Explain how text is converted to binary (4 questions)

Three points earn the marks:

1. A **character set** is used,
2. such as **ASCII or Unicode**;
3. each character has a **unique binary value** (code), and that value is what is stored.

To explain how a **word** such as "SUN" is stored, add: the code for S is stored, then the code for U, then the code for N, in sequence. Write about the character set, not about how the keyboard works; the examiners say describing key presses earns nothing here.

5. Calculate an image or sound file size (4 questions)

Image:

1. Pixels = width × height.
2. Bits = pixels × colour depth.
3. Bytes = bits ÷ 8, then ÷ 1024 for KiB, ÷ 1024 again for MiB.

For example, a 400 × 300 image with an 8-bit colour depth: $400 \times 300 \times 8 \div 8 = 120\,000$ bytes, and $120\,000 \div 1024 = 117.19$ KiB. If the colour depth is given in **bytes** (such as "2 bytes"), multiply by it straight away to get bytes.

Sound:

1. Bits = sample rate × sample resolution × length in seconds.
2. Bytes = bits ÷ 8, then ÷ 1024 for each step up.

For example, 5 seconds at 10 000 Hz with 8-bit resolution: $10\,000 \times 8 \times 5 \div 8 = 50\,000$ bytes.

Variations you will meet:

- **Several files**: work out one, then multiply by how many there are.
- **Unit questions** in the same part: bits in a byte (8), nibbles in a byte (2), bits in a KiB ($1024 \times 8 = 8192$), the unit equal to 1024 GiB (a TiB). Multiply by 1024 going down a unit and divide going up.
- If the question says you may use 1000 or 1024, either is fine, but use the same one all the way through.

Write each stage down: most of the marks are for the working, and the last mark is for the answer in the unit asked for.

6. Compare ASCII and Unicode (3 questions)

- **Advantages of Unicode**: it can represent more characters; it covers many more languages; it includes emojis and more symbols.
- **Disadvantage of Unicode**: each character needs more bits, so text takes more storage space (and longer to transmit).
- **"Explain the differences"** (4 marks): make comparisons, each naming both sides: ASCII has fewer characters (128, or 256 extended) while Unicode has far more; ASCII covers a limited set of (mostly English) characters while Unicode covers many languages and emojis; ASCII uses 7 or 8 bits per character while Unicode uses up to 16 or 32.
- **"Unicode is an example of what?"** A character set. **"Name one other character set"** when the question gives ASCII: Unicode (and the other way round).

7. Effect of sample rate and sample resolution (3 questions)

- **"Give two ways to make the recording more accurate"**: increase the sample rate; increase the sample resolution.
- **Why a higher sample resolution is more accurate**: more bits for each sample, so there are more possible amplitude values; each amplitude is stored more precisely, and a wider range of amplitudes can be recorded.
- **Why a higher sample rate is more accurate**: more samples each second, so the recording is closer to the original sound.
- **Drawback of either**: the file size increases, so it needs more storage space (and takes longer to send).
- A compression method that **permanently lowers** the sample rate, sample resolution or colour depth is **lossy**; one that removes no data permanently is **lossless**.

8. Work with character codes (2 questions)

1. **A code from another code**: codes run in alphabetical order, so count along. If G is 71, then H is 72, I is 73 and J is 74. Capitals and small letters are separate runs: j is $74 + 32 = 106$.
2. **Convert a code** to binary or hexadecimal with the Number systems methods; give 8 bits if the question shows 8 (for example 74 is $64 + 8 + 2$, so 01001010).
3. A question may go on to add, shift or convert these binary codes; those parts are Number systems skills.

9. Describe how an image is stored; image file formats (2 questions)

"Describe how a digital image is stored" (3 marks). Build the answer the way the house diagram in "Understand it" works: the picture is split into a grid of pixels, so many across and so many down (its resolution). Each pixel holds a single colour, and that colour is written as a binary code whose length is the colour depth. The computer keeps these codes one pixel after another, row by row. Alongside them the file keeps metadata, such as the width, height and colour depth, so the computer knows how to rebuild the grid on screen. Always say what the binary stands for (the colour of each pixel), not just "it is stored in binary".

Image formats: JPEG, PNG, GIF, BMP, TIFF. A **JPEG** file holds an **image** and is a **lossy** compressed file.

10. Explain why data is converted to binary (1 question)

A computer is made of switches and logic gates that can only be on or off (two states), so all data, including text, sound and images, must be converted to binary (1s and 0s) to be processed.

Worked examples

The bold codes show where marks are earned, as in Cambridge mark schemes. The number is how many marks.

- **M1** method mark: for a correct method, even if the numbers slip.
- **A1** accuracy mark: for a correct answer; it needs the M mark before it.
- **B1** mark for a correct result on its own, with no method needed.

Example 1

A messaging app stores text using ASCII. The ASCII denary code for "D" is 68.

(a) Give the ASCII denary code for "H". **[1]**

(b) Convert your answer to (a) to 8-bit binary and to hexadecimal. **[2]**

(c) Explain how the word "HEAD" is stored using a character set. **[2]**

(d) The app is changed to use Unicode. Give two advantages and one disadvantage of this change. **[3]**

(a) The letters are in order: D 68, E 69, F 70, G 71, H **72**. **B1**

(b) $72 = 64 + 8$, so **01001000**. **B1** Nibbles 0100 and 1000 are 4 and 8, so **48**. **B1**

(c) Each character in the character set has its own unique binary code. **B1** The code for H (72) is stored, then E (69), then A (65), then D (68), in sequence: 01001000 01000101 01000001 01000100. **B1**

(d) Advantages: Unicode can represent many more characters, **B1** including other languages and emojis. **B1** Disadvantage: each character needs more bits, so messages take up more storage space. **B1**

Example 2

A photo has a resolution of 1200 × 800 pixels and a colour depth of 24 bits.

(a) State what is meant by a colour depth of 24 bits, and how many different colours this allows. **[2]**

(b) Calculate the file size of the photo in mebibytes (MiB), to 2 decimal places. Show your working. **[3]**

(c) The resolution is changed to 600 × 400. Explain the effect on the file size and on the image. **[2]**

(a) 24 bits are used to represent each colour (each pixel). **B1** $2^{24} = 16\,777\,216$ colours are possible. **B1**

(b) $1200 \times 800 = 960\,000$ pixels; $960\,000 \times 24 = 23\,040\,000$ bits. **M1**

$23\,040\,000 \div 8 = 2\,880\,000$ bytes; $\div 1024 = 2812.5$ KiB; $\div 1024 = 2.75$ MiB. **M1** for dividing by 8 and by 1024 twice, **A1** for **2.75 MiB**.

(c) There are fewer pixels ($240\,000$, a quarter as many), so fewer bits to store and the file size decreases (to $720\,000$ bytes, a quarter). **B1** The image has less detail, so it is lower quality and may look blocky if enlarged. **B1**

Example 3

A podcaster records a 30-second advert. The sound is sampled at 44 100 Hz with a sample resolution of 16 bits.

(a) Explain how the sound is converted to binary. **[4]**

(b) Calculate the file size in kibibytes (KiB), to the nearest KiB. Show all your working. **[3]**

(c) The podcaster lowers the sample rate to 22 050 Hz. Describe the effect of this change. **[2]**

(a) Any four of: the sound is picked up by a microphone; **B1** the sound wave is sampled, measuring its amplitude at regular time intervals; **B1** 44 100 samples are taken every second (the sample rate); **B1** each sample is stored using 16 bits (the sample resolution); **B1** each amplitude is converted to its binary value and the samples are stored in order.

(b) $44\,100 \times 16 = 705\,600$ bits per second; $\times 30 = 21\,168\,000$ bits. **M1**

$21\,168\,000 \div 8 = 2\,646\,000$ bytes; $\div 1024 = 2583.98\ldots$ **M1**, so **2584 KiB**. **A1**

(c) Half as many samples are taken each second, so the file size halves (to $1\,323\,000$ bytes). **B1** The recording is less accurate, less close to the original sound. **B1**

Common mistakes

- **Mixing up the two sound terms.** Sample rate is samples per **second**; sample resolution is **bits per sample**. The June 2023 report says few candidates could define both, and some described sample resolution as if it were about images.
- **Vague colour depth answers.** It is the number of bits for **each** colour (or each pixel), not "the number of colours in the image", not "the number of pixels per bit", and not bits for all the colours together (March and June 2025 reports).
- **Resolution as "quality".** "The quality of the image" earns nothing; say **the number of pixels** (or pixels wide × pixels high).
- **Defining a pixel loosely.** "All the squares in the image" is wrong: a pixel is **one** square of **one** colour, the smallest part of the image (March 2023 report).
- **Describing the keyboard instead of the character set.** For "how is text converted to binary", key presses and switches earn nothing; the character set and unique codes do (June 2023 report).
- **Saying "it's stored in binary" without saying what the binary means.** For an image, say each pixel's colour has a binary code, and that value is stored (March 2022 report).
- **Forgetting metadata.** Few candidates knew the term for the extra data stored with an image, such as the date and time (June 2025 report).
- **File size method errors.** Most slips are in the method, not the arithmetic: multiply by the colour depth (or sample resolution), then divide by 8 before dividing by 1024 (November 2021 report). A byte has **2** nibbles, not 4.

How to get the marks

- **"State" / "Give"**: one precise fact is enough, but it must be precise: "the number of bits per sample", not "how good the sound is".
- **"Describe" or "Explain" a process** (text or sound to binary): write separate points in a logical order. Each correct point is a mark, so a 5-mark answer needs five different points: microphone, sampling, amplitude, sample rate, sample resolution, binary value.
- **"Explain the effect"**: state the change **and** the reason: "the file size increases, because more bits are stored for each colour".
- **"Explain the differences"**: every point must compare both sides, or be clearly about one of them. Answers only about ASCII (or only about Unicode) are capped.
- **"Describe the relationship"**: give the chain, not just the end: higher resolution → more pixels → more bits → larger file.
- **Calculations**: "Show all your working" means each stage earns a mark (pixels, bits, ÷ 8, ÷ 1024). Put the answer in the unit asked for.
- **Tables of terms**: use the exact term (colour depth, sample rate, metadata). "Hertz" or "frequency" is not the name for sampling.
- **Use the context and don't repeat it.** If the question already says "it is quicker to transmit", give a different reason.

Quick check

1. State what is meant by sample resolution. How many different amplitude values can a 4-bit sample resolution store?
2. The ASCII denary code for "k" is 107. Give the ASCII code for "p" in denary, 8-bit binary and hexadecimal.
3. An icon is 256 × 128 pixels with an 8-bit colour depth. Calculate its file size in KiB, and state how many colours it can use.
4. A website changes from ASCII to Unicode. Give two advantages and one disadvantage.
5. A voice memo is recorded for 1 minute at 8000 Hz with an 8-bit sample resolution. Calculate its size in bytes. What would it be with a 16-bit sample resolution?

Answers

1. The number of bits used to store each sample. $2^4 = 16$ different values.
2. k, l, m, n, o, p: $107 + 5 = 112$. $112 = 64 + 32 + 16$, so **01110000**; nibbles 0111 and 0000 give **70** in hexadecimal.
3. $256 \times 128 \times 8 = 262\,144$ bits; $\div 8 = 32\,768$ bytes; $\div 1024 =$ **32 KiB**. $2^8 = 256$ colours.
4. Advantages: it can represent more characters; it covers other languages and emojis. Disadvantage: each character needs more bits, so pages take more storage and take longer to send.
5. $8000 \times 8 \times 60 = 3\,840\,000$ bits $= 480\,000$ bytes. With 16 bits per sample it doubles, to $960\,000$ bytes.

Past questions to try

- **0478/11/O/N/23 Q5**: how analogue sound is converted, making it more accurate, and ASCII compared with Unicode.
- **0478/12/O/N/24 Q2**: the file size of a sound effect in KiB from its sample rate and resolution.
- **0478/12/F/M/23 Q4**: pixel and resolution defined, then an image's file size in bytes.
- **0478/11/O/N/24 Q3**: ASCII codes converted to binary and hexadecimal.