# Lossless and lossy compression (incl. RLE)

2.3.2 · Data · revision notes

Why data is compressed, the difference between lossless and lossy compression, and when each is suitable. Run-length encoding (RLE) is a simple example of lossless compression: when a question uses it, it shows you the format to follow. Expect definitions, 'explain why this method is suitable' questions and short RLE tasks.

## Key facts

- Compression reduces file size, saving storage and transfer time
- Lossy: data permanently removed; the original cannot be restored
- Use lossy for photos, music and video
- RLE stores each run as a count and a value, e.g. 5A 3B
- Lossless: no data lost; the original is restored exactly
- Use lossless for text, program code and spreadsheets
- Lossy usually gives smaller files than lossless
- RLE only helps when there are long runs of repeated values

## Notes

### Why compress data?

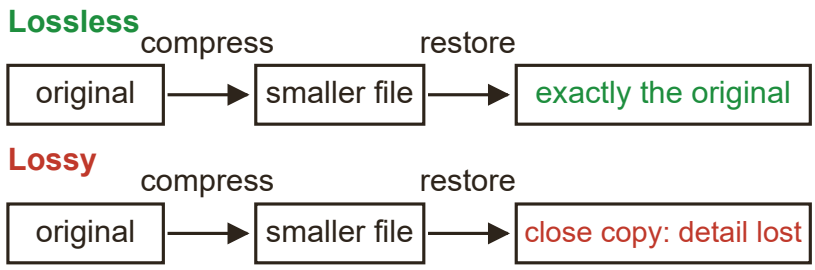
- **Compression** reduces the size of a file.
- Smaller files take up less storage space.
- They transfer faster over a network and use less bandwidth, so they are quicker to download, email or stream.

### Lossless compression

- **Lossless** compression reduces the file size without losing any data. When the file is decompressed, the original is restored exactly.
- It works by finding patterns and repeated data and storing them more efficiently.
- It must be used when every bit matters: text documents, program code and executable files, spreadsheets and databases.
- It usually reduces the file size less than lossy compression does.

## Lossy compression

- **Lossy** compression permanently removes some data, usually detail that people are unlikely to notice, such as sounds humans cannot hear or tiny differences in colour.
- The original file cannot be restored exactly.

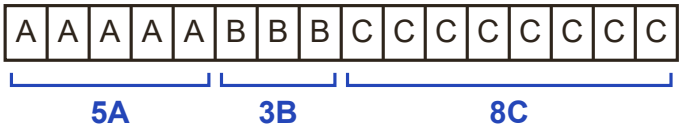



Only lossless compression gives back exactly the file you started with

- It gives much smaller files, so it is used for photos, music and video, especially for streaming and web pages.
- Too much compression causes a noticeable loss of quality.

## Run-length encoding (RLE)

- RLE is a simple lossless method. It replaces each **run** (the same value repeated one after another) with the number of repeats and the value.
- Example: AAAAABBBCCCCCCCC becomes 5A 3B 8C.




16 characters stored as 3 runs

Each run of repeated values is stored once, with a count

- A row of pixels in a black-and-white image: WWWWBBBBBBWW becomes 4W 6B 2W.
- RLE works well on data with long runs, such as simple images with large areas of one colour. It works badly when the values keep changing: ABCD becomes 1A 1B 1C 1D, which is bigger than the original.

## How to answer each type of question

### Explain which type of compression is suitable

2 marks | Grade 5

1. Name the type (lossless or lossy).
2. Give a reason linked to the file: must it be exact, or is a small loss of quality acceptable?
3. Add the consequence: what would go wrong with the other method, or the benefit of a smaller file.

**Example.** A software company lets customers download its app installer from its website.
Explain why the installer should be compressed using lossless compression.

---

**Model answer.** Lossless compression means no data is lost, so the file can be restored exactly to the original (1). A program must be exactly correct: if any data were lost, the app would not install or run properly (1).

### Apply run-length encoding

1 to 3 marks | Grade 6

1. Work from left to right, counting how many times each value repeats in a row.
2. Write the count and then the value for each run, in the format the question uses.
3. To decode, write each value out the number of times shown.
4. To compare sizes, count the runs and multiply by the bytes used for each run.

**Example.** A row of pixels in a two-colour image is: B B B W W W W W W W B B
(a) Give the RLE for this row.
(b) Each run is stored in 2 bytes. Without compression, each pixel is stored in 1 byte.
Calculate the number of bytes saved by using RLE for this row.

---

**Model answer.**

(a) 3B 7W 2B (1)
(b) Without RLE: 12 pixels × 1 byte = 12 bytes; with RLE: 3 runs × 2 bytes = 6 bytes (1)
12 − 6 = 6 bytes saved (1)

**Explain the differences between lossless and lossy compression** 4 marks **Grade 6**

1. Make each difference a comparison: 'lossless… whereas lossy…'.
2. Give one mark's worth of fact and one of explanation or example for each difference.
3. Good differences: whether the original can be restored; how small the file becomes; what each is used for.

**Example.** Explain **two** differences between lossless and lossy compression.

**Model answer.** Lossless compression keeps all the data, so the original file can be restored exactly (1), whereas lossy compression permanently removes some data, so the original cannot be restored (1). Lossy compression usually makes files much smaller than lossless compression (1), because it can remove detail that people are unlikely to notice (1).

**Discuss the choice of compression for a scenario** 4 marks **Grade 8**

1. Say what the chosen method does to the file size and why that matters in the scenario.
2. Say what it does to quality, and whether that matters here.
3. Compare with the other method.
4. Finish with a justified conclusion.

**Example.** A news website shows hundreds of photos. Many of its readers use mobile phones. Discuss whether the website should use lossy or lossless compression for its photos.

**Model answer.** Lossy compression makes the photos much smaller (1), so pages load faster and use less of readers' mobile data (1). The quality is slightly lower, but on a phone screen the difference is unlikely to be noticed (1). Lossless compression would keep perfect quality, but the files would be larger and pages slower to load, so lossy is the better choice (1).

## Don’t lose marks

- Saying lossless compression 'doesn't reduce quality' without saying the original can be restored exactly.
- Saying lossy compression 'removes quality'. Say it permanently removes data.
- Choosing lossy compression for text or program files.
- Writing RLE in a different format from the one in the question (e.g. value first when the example puts the count first).
- Assuming RLE always makes files smaller.

## More tips

### Memory tricks

- Lossy loses data for good; lossless loses nothing.
- Ask 'would one wrong bit matter?' Yes: lossless (text, code). No: lossy is fine (photos, music, video).
- RLE check: add up all the counts. The total must equal the number of values in the original.
- RLE saves space only when runs are long: a run of 1 takes more space than the single value did.

### Exam technique

- Always link your choice to the scenario: say why this file must be exact, or why a small loss of quality is acceptable.
- For 'explain two differences', make each difference a comparison (lossless… whereas lossy…).
- In RLE questions, follow the exact format of the example in the question, and show your run counts.

## What each grade needs

What you need to be able to do, from the first marks up to the top grade.

**Grade 3** — **Give reasons for compressing files**
Compressed files use less storage space and transfer faster, using less bandwidth.

**Grade 4** — **Define lossless compression**
File size is reduced without losing any data, so the original can be restored exactly.

**Grade 4** — **Define lossy compression**
Some data is permanently removed, so the original cannot be restored exactly.

**Grade 5** — **Choose the right method for a file**
Lossless for text, program code and spreadsheets; lossy for photos, music and video.

**Grade 6** — **Encode and decode data using RLE**
Replace each run of a repeated value with a count and the value, e.g. 5A.

**Grade 7** — **Explain when RLE works well**
RLE only saves space when the data has long runs of the same value.

**Grade 8** — **Evaluate compression methods for a scenario**
Weigh file size and speed against quality and whether the data must be exact.

Free notes, Full packs, practice papers and mock exams at chhetriacademy.com. Written for this site; not taken from past papers. Stuck? 1-to-1 tutoring with a free 30-minute trial lesson.