Objective:
- Show understanding of the need for and examples of the use of compression
- Show understanding of lossy and lossless compression and justify the use of a method in a given situation
- Show understanding of how a text file, bitmap image, vector graphic and sound file can be compressed. Including the use of run-length encoding (RLE)

Compression Techniques

Larger files require larger **storage** capacity and lower transmission or **download rates**. Compression techniques are used to reduce **file size**.

Two categories of compression.

Lossless Compression > **Lossless compression** where file size is **reduced** but no information is lost and when necessary, process can be reversed to **re-create original** file.

This is important for files where loss of any data would be disastrous (text file)

Types of Lossless Compression Algorithms:

i. **Huffman Coding** is a **lossless data** compression algorithm. Instead of having each character coded in one byte an **analysis** is carried out to find **most often used** characters. These are then given **shorter codes**. Original stream of bytes becomes a **bit stream**. Most **frequent** character gets **smallest code** and **least frequent** character gets **largest code**.

A possible set of codes if a text contained only eight different letters.

Code	Character
10	e
01	t
111	o
110	h
0001	l
0000	p
0011	w
0010	z

ii. **Run-length Encoding (RLE):**
RLE can be used to compress a number of different file formats. It is a form of lossless file compression that reduces size of a string of adjacent, identical data such as repeated colours in an image.
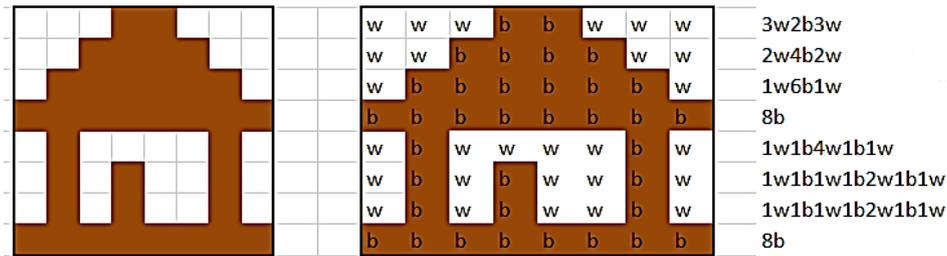
A repeating string is encoded into two values.

- **First value** represents **number** of identical data items (such as characters) in run.
- **Second value** represents code of data item (such as ASCII code).

Note: RLE is only effective where there is a long run of repeated units/bits.

Example: Using RLE on Text Data:

Consider text string **'aaaaabbbbccddddd'**.

Assuming each character requires 1 byte, then this string needs **16** bytes. If we assume ASCII code is being used, then string can be coded as follows:

a	a	a	a	a	b	b	b	b	c	c	d	d	d	d	d
05 97					04 98				02 99		05 100				

Assuming each number in second row requires **1 byte** of memory, RLE code will need 8 bytes. This is half the original file size.

Example: Using RLE with images

Black and white images Figure shows letter F in a grid where each square requires 1 byte of storage. White square has value 1 and black square a value of 0.

8 × 8 grid would need 64 bytes; compressed RLE format has 30 values, and therefore needs only 30 bytes to store image.

iii.   **Example:**

Consider a screen containing plain black text on a solid white background. There will be many long runs of white pixels in the blank space, and many short runs of black pixels within the text. **B** representing a black pixel and **W** representing white.

WWWWWWWWWWWWBWWWWWWWWWWWWBBBWWWWWWWWWWWWWWWW WWWWWWWWWWWWWBWWWWWWWWWWWWWW

With a **run-length encoding (RLE)** data compression algorithm applied to above line, it can be rendered as follows:

12W1B12W3B24W1B14W

This can be interpreted as a sequence of twelve Ws, one B, twelve Ws, three Bs, etc. Run-length code represents original 67 characters in only 18.

Consider following picture with brown pixels(b) on white background(w)

w	w	w	b	b	w	w	w	3w2b3w
w	w	b	b	b	b	w	w	2w4b2w
w	b	b	b	b	b	b	w	1w6b1w
b	b	b	b	b	b	b	b	8b
w	b	w	w	w	w	b	w	1w1b4w1b1w
w	b	w	b	w	w	b	w	1w1b1w1b2w1b1w
w	b	w	b	w	w	b	w	1w1b1w1b2w1b1w
b	b	b	b	b	b	b	b	8b

> **Lossy Compression**

Lossy compression where the file size is reduced with some loss of information and the original file can never be recovered.

Lossy compression can be used in circumstances where a sound file or an image file can have some of the detailed coding removed or modified when it is likely that the human ear or eye will hardly notice any difference.

*One **example** would be to reduce the colour depth for the coding of a bitmap.*

In many applications a combination of lossless and lossy methods may be used.

Photographic (bit-map) images :

When a photographic file is compressed, both the file size and quality of image are reduced. A common file format for images is JPEG, which uses **lossy** file compression.

Once the image is subjected to the JPEG compression algorithm, a new file is formed and the original file can no longer be constructed. A JPEG will reduce the **raw bit-map** image by a factor of between 5 and 15, depending on the quality of the original.

Vector graphics can undergo file compression. Scalable vector graphics (.svg) are defined in text files which allows them to be compressed.

File Compression in MPEG-3 (MP3) and MPEG-4 (MP4)

MPEG-3 (MP3) uses technology known as **audio compression** to convert music and other sounds into an MP3 file format. This compression technology will reduce size of a normal music file by about 90%. **For example**, 80 MB music file on a CD can be reduced to 8 MB using MP3 technology.

MP3 files use what is known as a **lossy** format, since part of original file is lost following compression algorithm. This means that original file cannot be put back together again.

TYK: When storing music tracks in a computer, MP3 format is used. This reduces file size by about 90%. Explain how music quality is apparently retained.

This is done using file compression algorithms that use **perceptual music shaping**.
Perceptual music shaping removes certain sounds. **For example**
- **frequencies** that are outside the human hearing range
- if two sounds are played at the same time, only **louder** one can be heard by ear, so softer sound is eliminated.

  This means that certain parts of music can be removed without affecting quality too much.

MPEG-4 (MP4) files are slightly different to MP3 files. This format allows storage of multimedia files rather than just sound. Music, videos, photos and animation can all be stored in the MP4 format. Videos could be streamed over internet using MP4 format without losing any real noticeable quality.
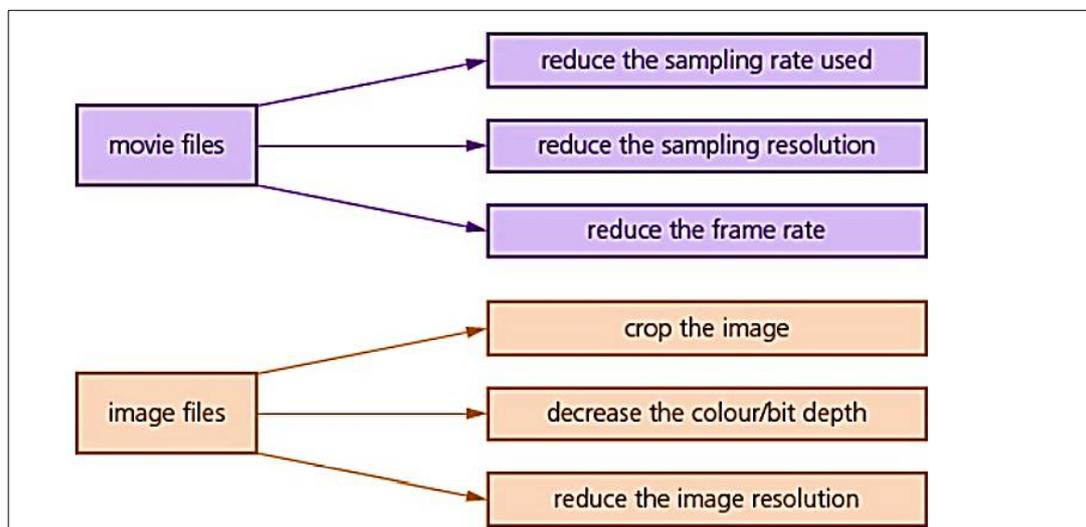
EXAM STYLE QUESTIONS

ESQ#1 Photographs are compressed before they are uploaded to a web server.

Customers download photographs from this web server.

(i) Explain reasons why compressing photographs will benefit customers.

Ans: Customers will be able to download photographs in **less time** and they will take less of **bandwidth.** Photographs will take up **less space** on customer's storage medium therefore customers can store **more images** and will have **more space** for other files.

(ii) An image can be compressed using RLE. Explain reasons why RLE may not reduce the file size of a bitmap image. Give one example in your answer. [3]

Ans: RLE stores a colour and number of times it occurs consecutively. An image may not have many sequences of the same colour. It would need to store each colour and then the count/number 1 which adds data.

Example: Red-Green-Blue would become Red 1 Green 1 Blue 1

****