Chapter 1 Student Book Answers

What you should already know

1 128 64 32 16 8 4 2 1

2 a) 01111101 (125)
   b) 10111011 (187)
   c) 01111101 (125)
   d) 10110110 (182)
   e) 11111000 (248)
   f) 101010100 (340)

3 65536 4096 256 16 1

4 a) 35E (862)
   b) C1F (3103)
   c) 1221 (4641)
   d) 9F6 (2550)
   e) 1058 (4184)
   f) 37B8 (14264)
   g) F4B5 (62645)
   h) 17780 (96128)
   i) AFFA (45050)
   j) D468 (54376)

Activity 1A

a) 51
b) 127
c) 153
d) 116
e) 255
f) 15
g) 143
h) 179
i) 112
j) 238

Activity 1B

a) 00101001
b) 01000011
c) 01010110
d) 01100100
e) 01101111
f) 01111111
g) 10010000
h) 10111101

i) 11001000
j) 11111111

Activity 1C

a) 01110010 (114)
b) 00111101 (61)
c) 01100000 (96)
d) 11110010 (−14)
e) 10001100 (−116)

Activity 1D

a) 01100010
b) 01101110
c) 1) 0000000
d) 1) 0110001
e) 00101011
f) 1) 00110011
g) 1) 00100101
h) 11110000
i) 10011111
j) 1) 00011110

Activity 1E

a)	1100 0011	C3
b)	1111 0111	F7
c)	0010 0111 1111	27F
d)	0100 1110 1110	4EE
e)	0001 1110 0001	1E1
f)	1000 1001 1110	89E
g)	0000 0100 1111 1110	(0)4FE
h)	0000 1110 1001 1100	(0)E9C
i)	1111 1111 0111 1101	FF7D
j)	0000 0110 0111 1010 1110	(0)67AE

Activity 1F

a) 0110 1100
b) 0101 1001
c) 1010 1010
d) 1010 0000 0000
e) 0100 0000 1110
f) 1011 1010 0110
g) 1001 1100 1100
h) 0100 0000 1010 1010
i) 1101 1010 0100 0111
j) 0001 1010 1011 0000

Activity 1G

1 a) 0010 0111 0001

b) 0101 0000 0000 0110

c) 0111 1001 1001 0000

2 a) 937

b) 7762

Activity 1H

a) 0000.01100110 0.66

b) 0001.00010111 1.17

c) 0001.01100011 1.63

Activity 1I

1 a) **Lossless:**

- All the data from the original file are reconstructed when the file is uncompressed.
- None of the original detail is lost – important for files where loss of data cannot be tolerated.

Lossy

- The file compression algorithm eliminates unnecessary data.
- The original file cannot be reconstructed following uncompression.
- A lossy algorithm has to make a decision about which parts of the file are less important and can be discarded.

b) Lossless – RLE (others exist)

Lossy – MPEG/JPEG, MP3/MP4 (and others exist)

2 a) Music is in analogue sound form initially. The microphone turns the sound into electrical signals. These signals are digitised and sent to computer for storage.

b) Music is stored in lossy (MP3) format. This reduces the size of the file, thus reducing memory requirements for storage and also allowing more tracks to be stored on a CD/MP3 device (for example). File compression uses algorithms that utilise perceptual music shaping – this essentially removes sounds the human ear can't hear properly. For example, if two sounds are played at the same time, only the louder one can be heard thus eliminating the softer sound; also certain sounds outside normal human range are removed – this allows considerable reduction in file size.

3 a) RLE – is a form of lossless file compression that reduces the size of a string of adjacent, identical data. For example, repeated colours in a string of pixels in an image.

b)

- reduces the size of a string of adjacent and identical data
- a repeated string is encoded into two values
- one of the values represents the number of identical characters in the run
- the second value represents the code of the character in the run
- only effective with long run of repeated bits e.g. aaaaabbbbccddddd (assuming ASCII coding used) is reduced to: 05 97 04 98 02 99 05 100 (8 bytes of data compared to 16 bytes in original string).

4 a) **Bit-map**

- made up of pixels (picture elements)
- image stored as an x-y two-dimensional matrix of pixels

- image may be scaled up or down but there may be loss of resolution (i.e. pixel density decreases to a level where picture quality isn't good).

Vector

- images are made up of 2D points that describe lines and curves and are then grouped into geometric shapes
- properties such as line colour and style are part of image (these form part of a drawing list)
- easy to scale up with no loss of quality since dimensions of each object in the graphic are not defined.

b)

Bit-map image	• requires less processing power • individual elements cannot be grouped together • bit-map files are larger than vector graphic images • most suitable for photos and scanned in images • at least 8 bits per pixel needed to code a colour image • resolution needs to be considered (number of pixels per row and per column) • possible to scale image up or down but pixel density may be reduced resulting in loss of quality (pixilation) • they rely on certain properties of the eyes; thus, a certain amount of lossy file compression can be tolerated.
Vector graphics image	• contain a drawing list which contains attributes such as line colour, line type, in-fill colours and so on • dimensions of each object not stored (only defined in relation to each other; thus, scale up has no loss in quality) • to print out vector graphic image, it first needs to be converted into bit-map image • they are most suitable for geometric shapes • very difficult to compress the file size.

Extension activity 1A

–32768, 16384, 8192, 4096, 2048,1024, 512, 256, 128, 64, 32, 16, 8, 4, 2, 1

End of chapter questions

1 a) i) 0100 1111 = 79

ii) 1001 1010 = –102

iii) –53 = 11001011

iv) range is: 10000000 (–128) to 01111111 (+127)

b) i) 798 = 0111 1001 1000 in BCD

ii) 9776

c) storage of digital displays on calculators (accept other valid uses)

2 a) i) Sampling Resolution – is a number of values available to encode each sample. It is specified by number of bits per sample (bit depth).

ii) A larger sampling resolution leads to more values available improving accuracy of sound digitised.

b) i) number of pixels per unit

ii) 16-colour bit map image requires 4 bits (½ byte) per pixel

iii) number of pixels = 16384 × 512

1 pixel = 1 byte ⇨ (16384 × 512)/1024 = 8192/1024 ≃ 8 GiB storage

iv)

- file type (e.g. .bmp)
- file size
- image resolution
- colour depth (bits per pixel, e.g. 1, 4, 8, 16, 24, 32)
- type of compression being used.

v)

- edit start/stop time and duration of sound clip
- extract/delete/save part of a sound clip
- ability to alter frequency, amplitude and pitch of the sound clip
- fade in/fade out facility
- mix/merge multiple soundtracks or sound sources
- combine various sources at different volume levels
- removal of noise, for example, to enhance one particular sound in a clip
- conversion between audio file formats.

3 a)

- The amplitude of the sound wave is first determined at set time intervals (the sampling rate).
- This gives an approximate representation of the sound wave.
- The sound wave is then encoded as a series of binary digits.
- Using a higher sampling rate or larger resolution will result in a more faithful representation of the original sound source.

b)

- music compression algorithm uses lossy format
- perceptual music shaping is used therefore loss of sound quality not noticed
- music files are large therefore compression needed and lossy also gives greater compression than lossless.

c) (i) **run length encoding**

- reduces size of a string of adjacent and identical data
- repeated string encoded into two values
- one value represents number of identical characters in a run
- second value represents code for each character in run.

(ii) assume grey = 85 and white = 255 then we have the RLE code:

3, 85, 2, 255, 4, 85, 9, 255, 4, 85, 2, 255, 1, 85, 2, 255, 2, 85, 2, 255, 1, 85

need 1 byte per pixel ⇨ number of bytes = 1 x 8 x 4 = 32 for original diagram; RLE needs 22 bytes only

4 a) 60 = 00111100

27 = 00011011

–27 = 11100101

b
```
   00111100
 + 00011011
 = 01010111
```

c
```
      00111100
  +   11100101
  = 1) 00100001
```

d
```
   01011001
 + 01100001
 = 10111010
```
– gives negative result which is not possible when adding two positive numbers

5 **a** 0.52 = 00000000 . 0101 0010

0.83 = 00000000 . 1000 0011

add .02 and .03 together gives: 0101

now add 0.5 and 0.8 together and this gives 1101 (which doesn't have a denary value)

thus we add 0110 to 1101 and this gives: 1) 0011

therefore we get 0011 with a carry of 1 giving final answer:

00000001 . 0011 1101 = 1.35

b **(i)** Hexadecimal – a number system using base 16.

(ii)

- memory dumps
- HTML
- assembly code instructions

(iii) 0111 1110 1111 0010

7 E F 2

6 **a** 95 = 1001 0101

b using two's complement this becomes 00100011 + 10111100 = 11011111

(i.e. 35 – 68 = -33)

c 506 = 1 F A