CHAPTER 1: ACTIVITIES

1.1.2 Binary number system

ACTIVITY 1A

Convert these binary numbers into denary.

a) 0 0 1 1 0 0 1 1
b) 0 1 1 1 1 1 1 1
c) 1 0 0 1 1 0 0 1
d) 0 1 1 1 0 1 0 0
e) 1 1 1 1 1 1 1 1
f) 0 0 0 0 1 1 1 1
g) 1 0 0 0 1 1 1 1
h) 0 0 1 1 0 0 1 1
i) 0 1 1 1 0 0 0 0
j) 1 1 1 0 1 1 1 0

Activity 1A

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

ACTIVITY 1B

Convert these denary numbers into binary (using either method).

a) 4 1
b) 6 7
c) 8 6
d) 1 0 0
e) 1 1 1
f) 1 2 7
g) 1 4 4
h) 1 8 9
i) 2 0 0
j) 2 5 5

Activity 1B

a) 00101001
b) 01000011
c) 01010110
d) 01100100
e) 01101111
f) 01111111
g) 10010000
h) 10111101
i) 11001000
j) 11111111

ACTIVITY 1C

Convert these denary numbers into 8-bit binary numbers using two's complement where necessary. Use these binary column weightings:

−128	64	32	16	8	4	2	1

a) +114
b) +61
c) +96
d) −14
e) −116

Activity 1C

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

ACTIVITY 1D

Carry out these binary additions and subtractions using these 8-bit column weightings:

−128 64 32 16 8 4 2 1

a) 00111001 + 00101001
b) 01001011 + 00100011
c) 01011000 + 00101000
d) 01110011 + 00111110
e) 00001111 + 00011100
f) 01100011 − 00110000
g) 01111111 − 01011010
h) 00110100 − 01000100
i) 00000011 − 01100100
j) 11011111 − 11000011

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

1.1.3 Hexadecimal number system

ACTIVITY 1E

Convert these binary numbers into hexadecimal.

a) 11000011
b) 11110111
c) 1001111111
d) 100111011110
e) 000111100001
f) 100010011110
g) 0010011111110
h) 0111010011100
i) 1111111101111101
j) 0011001111010111 0

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

Convert these hexadecimal numbers into binary.

a) 6 C
b) 5 9
c) A A
d) A 0 0
e) 4 0 E
f) B A 6
g) 9 C C
h) 4 0 A A
i) D A 4 7
j) 1 A B 0

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	

1.1.4 Binary-coded decimal (BCD) system

ACTIVITY 1G

1 Convert these denary numbers into BCD format.
   a) 2 7 1
   b) 5 0 0 6
   c) 7 9 9 0
2 Convert these BCD numbers into denary numbers.
   a) 1 0 0 1 0 0 1 1 0 1 1 1
   b) 0 1 1 1 0 1 1 1 0 1 1 0 0 0 1 0

Activity 1G

1 a) 0010 0111 0001
   b) 0101 0000 0000 0110
   c) 0111 1001 1001 0000
2 a) 937
   b) 7762

ACTIVITY 1H

Carry out these BCD additions.
a) 0.45 + 0.21
b) 0.66 + 0.51
c) 0.88 + 0.75

Activity 1H

a) 0000.01100110 0.66
b) 0001.00010111 1.17
c) 0001.01100011 1.63

1.3.2 General methods of compressing files

ACTIVITY 1I

1 a) What is meant by *lossless* and *lossy* file compression?
   b) Give an example of a lossless file format and an example of a lossy file format.
2 a) Describe how music picked up by a microphone is turned into a digitised music file in a computer.
   b) Explain why it is often necessary to compress stored music files. Describe how the music quality is essentially retained.
3 a) What is meant by *run length encoding*?
   b) Describe how RLE compresses a file. Give an example in your description.
4 a) Describe the differences between bit-map images and vector graphics.
   b) A software designer needs to incorporate images into her software to add realism. Explain what she needs to consider when deciding between using bit-map images and vector graphics in her software.

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

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.

CHAPTER 1: QUESTIONS

1 a) The following bytes represent binary integers using the two's complement form. State the equivalent denary values.

i) 0 1 0 0 1 1 1 1

[1]

ii) 1 0 0 1 1 0 1 0

[1]

iii) Write the integer −53 in two's complement form.

[1]

iv) Write the maximum possible range of numbers using the two's complement form of an 8-bit binary number.
Give your answers in denary.

[2]

b) i) Write the denary integer 798 in binary-coded decimal (BCD) format.

[1]

ii) Write the denary number that is represented by the following BCD number.

1	0	0	1	0	1	1	1	0	1	1	1	0	1	1	0

[2]

c) Give **one** use of binary-coded decimal system.

[1]

2 A software developer is using a microphone and a sound editing app to collect and edit sounds for his new game.
When collecting sounds, the software developer can decide on the sampling resolution he wishes to use.

a) i) State what is meant by *sampling resolution*.

[1]

ii) Describe how sampling resolution will affect how accurate the stored digitised sound will be.

[2]

b) The software developer will include images in his new game.

i) Explain the term *image resolution*.

[1]

ii) The software developer is using 16-colour bit-map images.
State the number of bits required to encode data for one pixel of his image.

[1]

iii) One of the images is 16 384 pixels wide and 512 pixels high.
The developer decides to save it as a 256-colour bit-map image.
Calculate the size of the image file in gibibytes.

[3]

iv) The bit-map image will contain a *header*.
State **two** items you would expect to see in the header.

[2]

v) Give **three** features you would expect to see in the sound editing app.

[3]

3 The editor of a movie is finalising the music score. They will send the final version of the score to the movie producer by email attachment.

a) Describe how *sampling* is used to record the music sound clips.

[3]

b) The music sound clips need to undergo some form of data compression before the music editor can send them via email.
Identify the type of compression, lossy or lossless, they should use.
Give a justification for your answer.

[3]

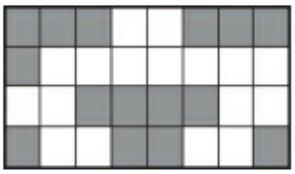
c) One method of data compression is known as *run length encoding (RLE)*.

i) Explain what is meant by RLE.

[3]

ii) Show how RLE would be used to produce a compressed file for the image below. Write down the data you would expect to see in the RLE compressed format (you may assume that the grey squares have a code value of 85 and the white squares have a code value of 255).

[4]



4 a) Write the denary numbers 60, 27 and −27 in 8-bit binary two's complement form.

[3]

b) Show the result of the addition 60 + 27 using 8-bit binary two's complement form. Show all of your working.

[2]

c) Show the result of the subtraction 60 − 27 using 8-bit binary two's complement form.

[2]

d) Give the result of the following addition.

0 1 0 1 1 0 0 1

+

0 1 1 0 0 0 0 1

Explain why the expected result is not obtained.

[2]

5 a) Carry out 0.52 + 0.83 using binary-coded decimal (BCD). Show all of your working.

[4]

b) i) Define the term *hexadecimal*.

[1]

ii) Give **two** uses of the hexadecimal system.

[2]

iii) Convert the following binary number into hexadecimal.

0 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0

[2]

6 a) Convert the denary number 95 into binary coded decimal (BCD).

[1]

b) Using two's complement, carry out the binary subtraction:
0 0 1 0 0 0 1 1 – 0 1 0 0 0 1 0 0
and convert your answer into denary.

[3]

c) Convert the denary number 506 into hexadecimal.

[1]

CHAPTER 1: ANSWERS

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