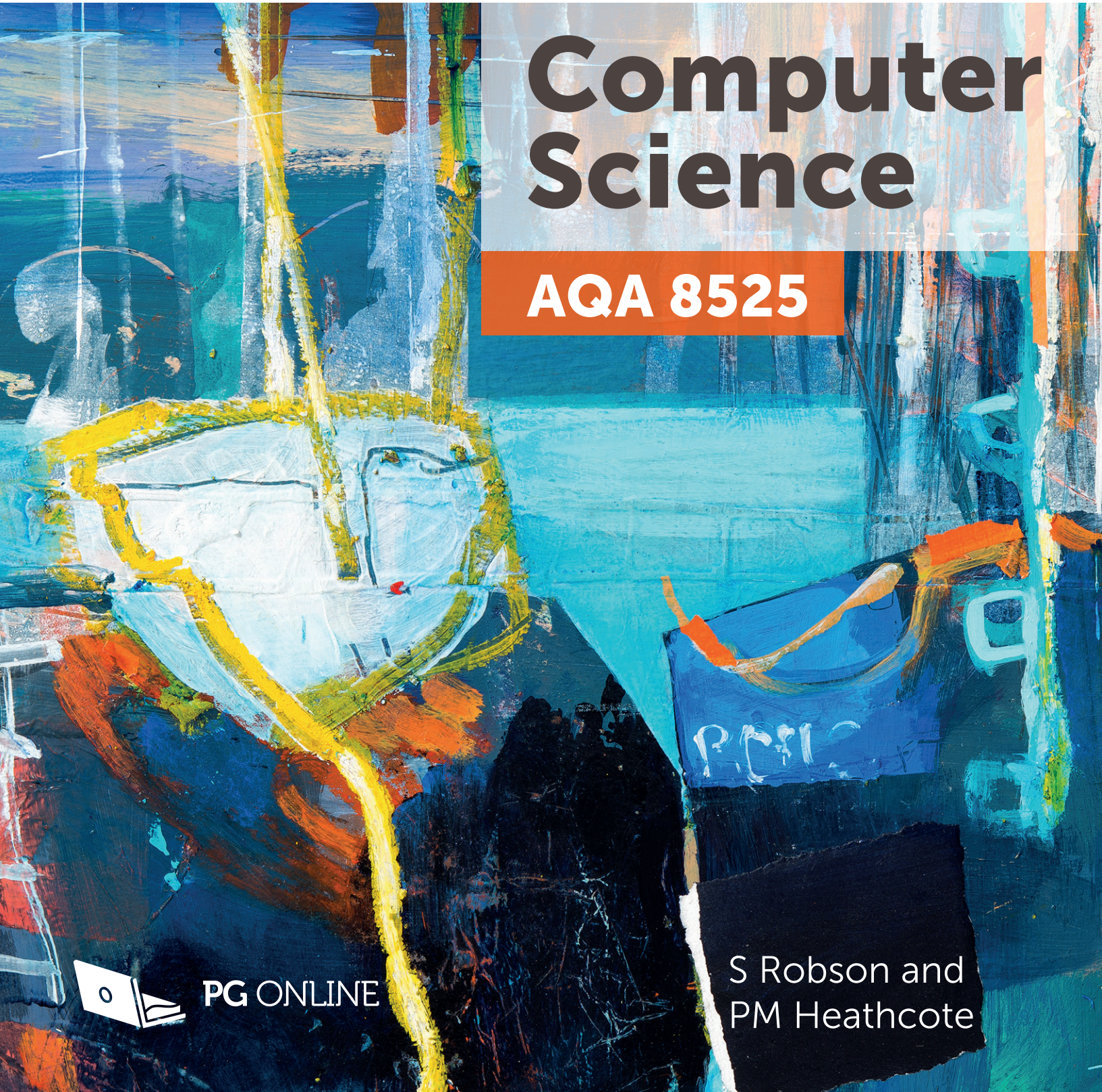
AQA Approved

GCSE (9-1)

Computer Science

AQA 8525



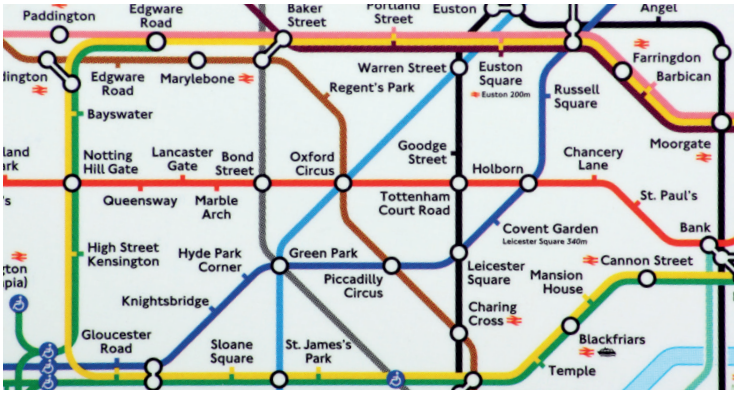
PG ONLINE

S Robson and
PM Heathcote

Contents

Abstraction allows us to separate the 'logical' from the 'physical'. A good example of this is the map of the London Underground – all we need to know is what stations are on which line, and the best route to get from A to B. There is no need to get bogged down in details of the exact distance between stations or even in which direction the route actually takes us at any given moment.

Similarly, we are all quite happy to use a computer or drive a car without having much idea of how it works. A driver, a child in the back seat and a mechanic all have a very different view of a car. We abstract away everything we don't need to know about and concentrate on the essentials.



1.2 Developing algorithms using flowcharts

1

In computing we write programs or create computer systems to 'solve a problem'. The **problem** is the need or requirement we have to meet. The solution could be a simple program but is more likely to be a complex suite of hardware and software in a real-world scenario, which will need to be broken down into many programs and subroutines.

Understanding how to solve the problem is important. You cannot just start coding at line 1 and hope to get a working solution straight away. The first step is to write an **algorithm** – that is, the series of steps needed to solve the problem.

This section will consider how algorithms are developed with the aid of **flowcharts** and **pseudocode**. Flowcharts are diagrams which use certain symbols to show the flow of data, processing and input/output taking place in a program or task.

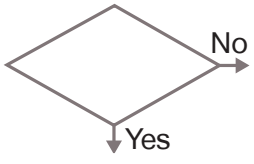
Standard flowchart symbols

This is used to START and END the flowchart.

This is a process box, for example:
`count ← count + 1` or `total ← (a * b) + 3`

This is an input/output box, for example:

`INPUT number` or `OUTPUT total`



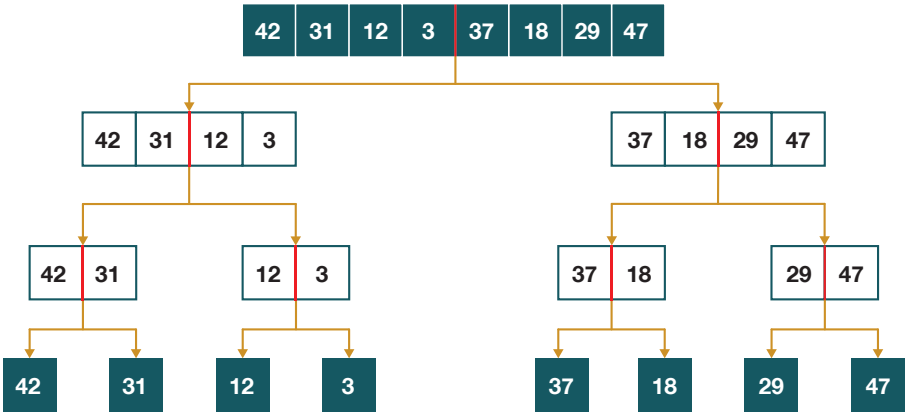
This is a decision box that can only accept "yes" or "no" answers, for example: `"Is number less than 0?"`

Merge sort

This is a two stage sort. In the first stage, the list is successively divided in half, forming two sublists, until each sublist is of length one.

Example 5: Sorting a list in ascending order

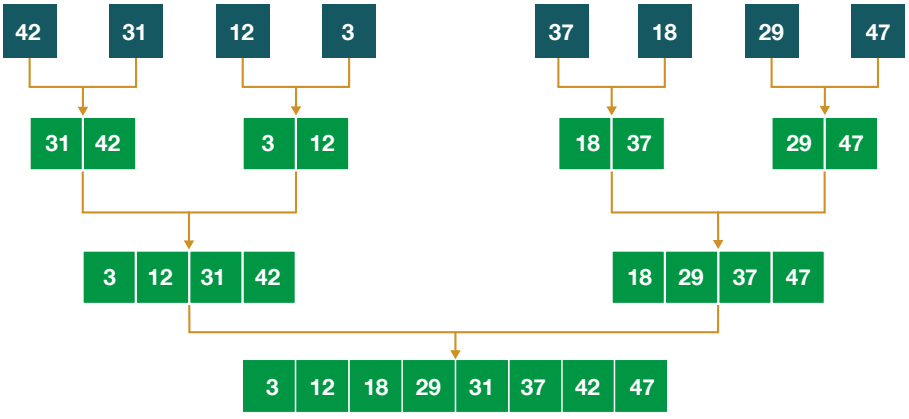
Stage 1



This is the end of stage 1 where all the elements have been separated out.

In the second stage, each pair of sublists is repeatedly merged to produce new sorted sublists until there is only one sublist remaining. As each pair of lists is merged, they are merged in order. Merging the final two sublists results in the sorted list.

Stage 2



This is the end of stage 2, with all the items recombined in sorted order.

Q22 Carry out a merge sort on the following set of numbers. The numbers are to be sorted in ascending order.

6	8	1	17	27	11	15	3

(a) Write out the four sorted sublists after the first phase of Stage 2 (the merge process).

(b) Write out the two sorted sublists after the second phase of the merge process.

(c) Write out the complete list after the third phase of the merge process.

Exercises

1. **Abstraction** and **decomposition** are two aspects of computational thinking.

   (a) Sienna is designing a program to control a cat-flap which will open only when a cat belonging to the owner approaches.

   Describe two ways in which she may use abstraction in reaching a solution to this problem. [2]

   (b) A program is required to enter a set of students' examination marks, count the number of students who obtained each mark and output the counts for each mark. Examination marks entered must be in the range 0 to 100.

   Explain how **decomposition** might be used in designing a solution to this problem. [3]

2. (a) (i) A bubble sort is used to sort the following numbers in ascending order:

   34, 56, 89, 23, 12, 77, 49, 44

   Write the order that the numbers be in after the first pass. [2]

   (ii) State the number of passesrequired to sort the items? (No flag is used to indicate a sorted list.) [1]

   (b) A **merge sort** is to be used to sort the same numbers. During the merge phase, the following four pairs of numbers need to be merged into two groups of four.

   (34, 56), (23, 89), (12, 77), (44, 49)

   Write the contents of each group of four numbers after the next phase of the merge? [2]

3. A list of surnames is held in sorted order. The names are:

   Beck, Coe, Ford, Grey, Hill, Kerr, Lunn, Pugh, Ross, Shaw, Taft, Ward

   (a) State which names would be examined when searching for the name **Grey** using

   (i) a linear search [1]

   (ii) a binary search [1]

   (b) State which names would be examined when searching for the name **James** using

   (i) a linear search [1]

   (ii) a binary search [1]

   (c) In a list of 1000 items, state the maximum number of names that would have to be searched to find a particular name using

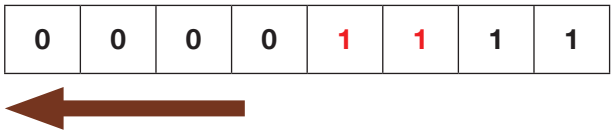
   (i) a linear search [1]

   (ii) a binary search [1]

Binary shifts

If a binary number is shifted to the left this is equivalent to multiplying the number by 2 for each shift to the left.

For example: If we shift:

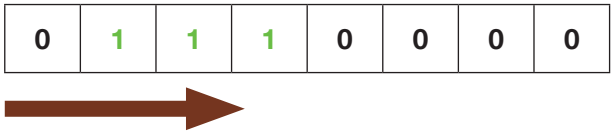


TWO places to the left we get the binary number:

(NOTE: Fill empty binary positions with 0s as you shift to the left.)

The original binary number has a value of 15 (i.e. 8+4+2+1 = 15); the number after shifting two places to the left has the value 60 (i.e. 32+16+8+4 = 60). It is multiplied by 4, or 2^2.

Shifting binary numbers to the right has the opposite effect i.e. each shift to the right has the effect of dividing by 2. Thus if we shift:



THREE places to the right we get the binary number:

The original binary value was 112 (i.e. 64 + 32 + 16 = 112) and the value after shifting three places to the right is 14 (i.e. 8 + 4 + 2 = 14). The number was divided by 8, and becomes 2^3. (NOTE: Fill empty binary positions with 0s as you shift to the right.)

Multiplication/division by powers of 2

This gives an easy way to multiply and divide binary numbers by powers of 2, but can come at the expense of accuracy. For example 00000110 shifted right twice to divide by 4 would be 00000001. This is the equivalent of decimal 1, but 6 / 4 = 1.5.

- Shifting right one place divides the number by 2
- Shifting left one place multiplies the number by 2

This is equivalent to shifting a decimal number right or left – for example shifting 12300 right gives 1230, i.e. it divides the number by 10. Shifting left multiplies a decimal number by 10.

Q7 Write down the results after the following shift operations and write down the decimal values before and after the shifts:

(a) The number 11001100 is shifted TWO places to the right

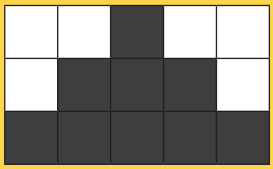
(b) The number 00011001 is shifted TWO places to the left

(c) The number 11001000 is shifted THREE places to the right

(d) The number 00000111 is shifted FOUR places to the left

(e) The number 10000000 is shifted FIVE places to the right

Q17 Using RLE, show how the image below would be coded, if black is encoded as 0 and white as 1.



Huffman coding

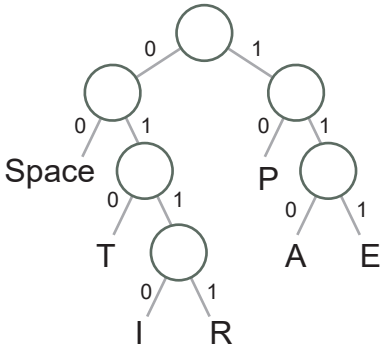
Huffman coding is a compression technique used to reduce the number of bits used to represent each letter. The more frequently a letter appears in the text, the fewer bits are used to represent it in a text file.

Example 3

Consider the sentence PIPPA ATE A PEPPER. A table showing the frequency of each character, including spaces is created as the first step in building the Huffman tree. For example, there is one "I", one "R", and six "P"s in the sentence.

Character	I	R	T	A	E	SPACE	P
Frequency	1	1	1	3	3	3	6

You will only be required to interpret the tree, not build it. A Huffman tree for this sentence is shown below. It is a binary tree in which characters that occur most frequently are nearer the top and therefore require fewer characters to encode them, as described below.



Using this Huffman tree, the coding for each character is derived from the path taken from the root node to the character. Branching left at a node is coded as 0, branching right is coded as 1.

Thus the character ‘A’ would be represented by the bit pattern 110 because from the top of the tree, you go right, right, left to reach A. The encoding for ‘T’ would be 001 and for ‘E’, 111.

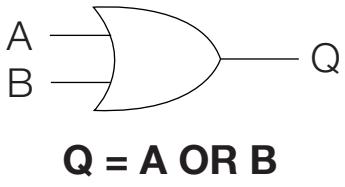
The total number of bits needed to represent the word “ATE” would be 3 + 3 + 3 = 9. In 7-bit ASCII, the number of bits required would be 3 x 7 = 21, representing a saving of 12 bits in the compressed format, with a 57% reduction in size.

Q18
(a) What would be the coding for the word PIT?
(b) How many bits would these three letters take using the Huffman code?
(c) The sentence PIPPA ATE A PEPPER is represented in a total of 47 bits. How many bits would be required to represent the sentence in ASCII?
(d) How many bits are saved by compressing PIT using Huffman coding?

The Boolean expression for AND is written: Q = A • B where • represents AND.

The truth table reflects the fundamental property of the AND gate: the output of A AND B is 1 (True) only if input A and input B are both 1 (True).

OR gate

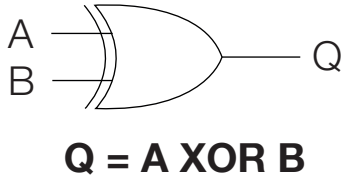


Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	1

The Boolean expression for OR is written: Q = A + B where + represents OR.

If A = 0 (False) and B = 0 (False) then A OR B = 0 (False), otherwise A OR B = 1 (True).

XOR gate



Input A	Input B	Output Q
0	0	0
0	1	1
1	0	1
1	1	0

The Boolean expression for XOR is written: Q = A ⊕ B where ⊕ represents XOR.

If either A or B, but not both, = 1 (True) then A ⊕ B = 1 otherwise A ⊕ B = 0 (False)

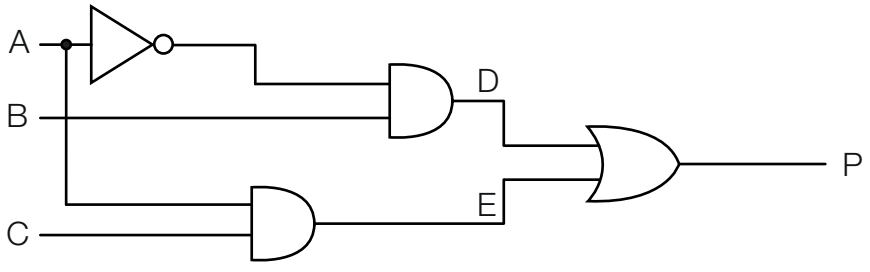
The XOR gate is known as the **exclusive OR** gate.

Combining logic gates into logic circuits

These logic gates can be combined to form more complex logic circuits which can carry out a number of functions. They are the basic building blocks of many electronic circuits found in computer memories, household devices, computer management systems in cars, and so on. Two examples are shown below. You should look at each logic circuit and follow the accompanying truth table which represents the logic circuit.

Example 1

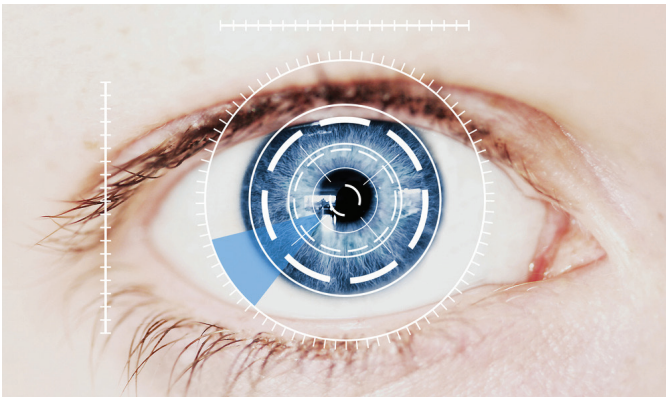
The logic circuit below represents the Boolean condition (NOT A AND B) OR (A AND C). The outputs at D and E have been labelled so that they can be referred to in the truth table overleaf.



Automatic software updates

Some popular software is a common target for malware. Browsers, PDF readers and other software can be automatically updated by selecting options to *Automatically update and install* either from the operating system or from the software. This will remove any harmful code that has been planted in the software by a hacker, or potential vulnerabilities that could be exploited in the future.

Biometric methods



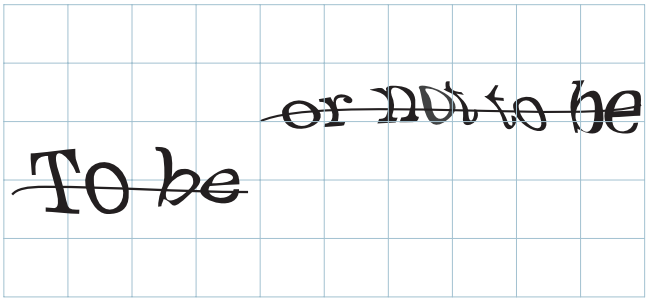
Employees of an organisation or members of the public passing through airport security, for example, may be asked to identify themselves by using a **biometric method** to prove to the system that they are who they claim to be. Biometric methods include a fingerprint scan, voice pattern sample or retinal scan. The probability of two people having identical biological characteristics is infinitesimally small, and so these methods can be used to positively identify a person.

Biometric methods are often used on mobile devices. The advantage of these methods over password entry are that it is not possible to steal or forget a biometric characteristic.

CAPTCHA

CAPTCHA is an acronym for “Completely Automated Public Turing test to tell Computers and Humans Apart", and is a type of test to determine whether or not the user is human.

A piece of text is displayed on screen in a format indecipherable by text recognition software. Context is critical; a **t** might look like an **l** or **i**, and it is only in context that a human can identify it as a **t**.

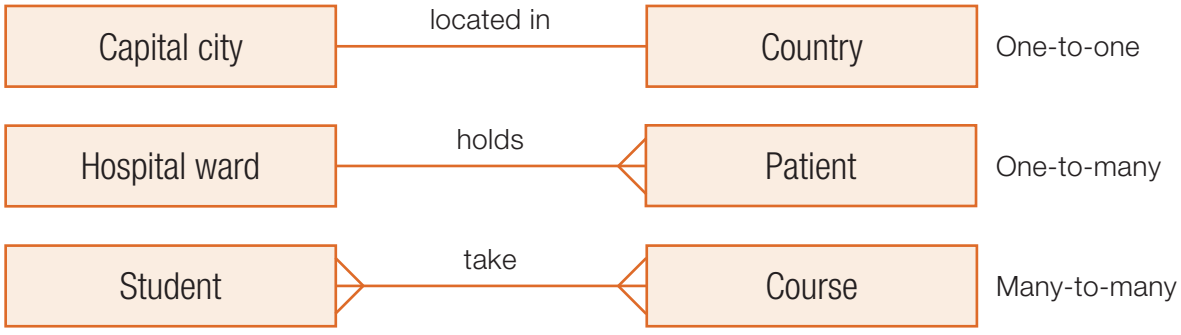


Q6 Even perfectly sighted individuals sometimes find CAPTCHA text very difficult or impossible to read. Does the use of CAPTCHA images discriminate against any computer users?

Q3 Looking at the Owner and Animal tables, who owns Lottie the rabbit? Which animals does Mrs Feinnes own?

Relationships between tables in a database

There are three possible relationships between database tables. Entity-relationship diagrams can be drawn to show what the relationship is between two tables. For example:



The 'many' end of a relationship is shown by the "crow's feet" in the relationship diagram.

Q4 What type of relationship is there between the Owner and Animal tables?

Q5 Draw diagrams to show the relationship between

(a) pet and owner

(b) magazine and contributor

(c) school and head

You may have to state what assumptions you are making before you can draw the diagram.

7.3 Structured query language

Structured Query Language, or **SQL**, is a language used for querying and updating database tables in a relational database. The **Owner** and **Animal** table in the Vet database will be used to demonstrate some SQL statements.

SELECT .. FROM .. WHERE

The SELECT statement is used to extract a collection of fields from one or more tables in a database. The syntax of the statement is

SELECT	list of fields to be displayed
FROM	list the table or tables where the data will come from
WHERE	list of search criteria
ORDER BY	list the fields that the results are to be sorted on (default is Ascending)

Q6 Think of some ways in which computers have helped to decrease environmental pollution, by monitoring the environment or by replacing old polluting industries with more environmentally friendly products, for example in the energy industry.

Monitoring the environment

Computers are widely used for monitoring the environment. Water quality in rivers and oceans, air quality in cities, pollen levels in the countryside which affect people with hay fever and asthma, radiation in nuclear plants, can all be monitored and warnings given.

Data which helps to predict earthquakes, tsunamis, hurricanes and other natural disasters is collected and analysed continuously all over the world. Weather data can be collected from the top of a mountain or from inside a volcano without a scientist having to go into these dangerous situations every day to collect it

Using data collected about weather, volcanic activity, and movement beneath the earth's surface, people can be warned about impending disasters such as hurricanes and earthquakes, and moved to safety.

Environmental impacts of cloud storage

Cloud storage uses vast amounts of energy to keep its storage equipment running, and cooling systems to combat the heat they generate also uses massive amounts of water and energy. However, cloud storage is generally much more energy efficient than storing all the data for an organisation in its own data storage centre. An organisation's data centre needs to be up and running 24/7, while its actual usage may be closer to 15%.

Large-scale cloud providers allow organisations to use fewer servers and therefore, less energy. Specially designed cloud data centres are more efficient than in-house hardware, using optimised equipment and sophisticated cooling equipment. This can result in very significant energy savings.

In addition, cloud storage enables many employees to work from home for at least part of the week, saving on the carbon emissions arising from commuting daily to work.

Many cloud providers are committed to reducing the environmental impact of using their technology. Data centres may be powered by wind and solar power alongside non-renewable energy sources. Google and Amazon purchase almost 100% renewable energy. Microsoft has reduced the huge amount of water used in cooling systems by designing air-cooling systems.

Index

Index continued

AQA GCSE 8525 (9-1) Specification map

		Unit 1	Unit 2A	Unit 2B	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8
3.1	**Fundamentals of algorithms**									
3.1.1	Representing algorithms	✓		✓						
3.1.2	Efficiency of algorithms	✓								
3.1.3	Searching algorithms	✓								
3.1.4	Sorting algorithms	✓								
3.2	**Programming**									
3.2.1	Data types		✓							
3.2.2	Programming concepts		✓							
3.2.3	Arithmetic operations in a programming language		✓							
3.2.4	Relational operations in a programming language		✓							
3.2.5	Boolean operations in a programming language		✓							
3.2.6	Data structures		✓							
3.2.7	Input / output		✓							
3.2.8	String handling operations in a programming language		✓	✓						
3.2.9	Random number generation in a programming language		✓							
3.2.10	Structured programming and subroutines			✓						
3.2.11	Robust and secure programming			✓						
3.3	**Fundamentals of data representation**									
3.3.1	Number bases				✓					
3.3.2	Converting between number bases				✓					
3.3.3	Units of information				✓					
3.3.4	Binary arithmetic				✓					
3.3.5	Character encoding				✓					
3.3.6	Representing images				✓					
3.3.7	Representing sound				✓					
3.3.8	Data compression				✓					

AQA GCSE 8525 (9-1) Specification map

		Unit 1	Unit 2A	Unit 2B	Unit 3	Unit 4	Unit 5	Unit6	Unit 7	
3.4	**Computer systems**									
3.4.1	Hardware and software					✓				
3.4.2	Boolean logic					✓				
3.4.3	Software classification					✓				
3.4.4	Classification of programming languages and translators					✓				
3.4.5	Systems architecture					✓				
3.5	**Fundamentals of computer networks**									
3.5	Computer networks						✓			
3.5	Network topologies						✓			
3.5	Network security						✓			
3.5	Protocols and layers						✓			
3.6	**Cyber security**									
3.6.1	Fundamentals of cyber security							✓		
3.6.2	Cyber security threats							✓		
3.6.1.1	Social engineering							✓		
3.6.1.2	Malicious code (malware)							✓		
3.6.3	Methods to detect and prevent cyber security threats							✓		
3.7	**Relational databases and SQL**									
3.7.1	Relational databases								✓	
3.7.2	Structured query language (SQL)								✓	
3.6.1.1	Social engineering								✓	
3.8	**Impacts of digital technology on wider society**									
3.8	Ethical, legal and environmental impacts									✓

AQA GCSE (9-1) 8525

Computer Science

The aim of this book is to provide an accessible text for students, covering the AQA GCSE (9-1) 8525 Computer Science specification. It can be used both as a course text and as a revision guide for students nearing the end of their course. It is divided into nine sections, each broken down into manageable chapters of roughly one lesson.

Sections 1, 2A and 2B of the textbook cover algorithms and programming concepts with a theoretical approach to provide students with experience of writing, tracing and debugging pseudocode solutions without the aid of a computer. These sections would complement practical programming experience.

Each section contains in-text questions and practice exercises, which can be set as homework. Answers to all these are available to teachers only, in a free Teachers' Supplement, which can be ordered from our website **www.pgonline.co.uk**

About the authors

Susan Robson worked for International Computers Ltd after graduating from Manchester University with a degree in Computer Science. She spent the following 12 years in technical pre-sales for ECI Telecom, before moving into teaching. As a Head of Computer Science, she gained years of experience teaching GCSE and A Level Computing and has written successful textbooks and teaching materials. She is currently teaching Computer Science at King Alfred's Academy in Wantage.

Pat Heathcote is a well-known and successful author of Computer Science textbooks. She has spent many years as a teacher of A Level Computing courses with significant examining experience. She has also worked as a programmer and systems analyst, and was Managing Director of Payne-Gallway Publishers until 2005.

Cover picture:

'Side by Side'
Mixed media on board,
88 x 64 cm
© Karen Stamper
www.karenstampercollage.com

This book has been approved by AQA.

ISBN 978-1-910523-22-3
9 781910 523223

PG ONLINE