AQA GCSE **Computer Science** 8525

Clear**Revise**™

Illustrated revision and practice:

- Over 500 marks of examination style questions
- Answers provided for all questions within the book
- Illustrated topics to improve memory and recall
- Specification references for each topic
- Examination tips and techniques
- Free Python solutions pack

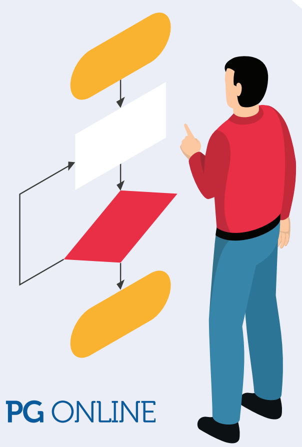
Experience + science + beautiful design = better results

Absolute clarity is the aim with a new generation of revision guide for the 2020s. This guide has been expertly compiled and edited by successful former teachers of Computer Science, highly experienced examiners and a good measure of scientific research into what makes revision most effective.

PG Online have a record of significantly raising and sustaining examination results at GCSE in schools using their award-winning teaching resources. This book aims to make an even greater difference.

Past examination questions are essential to good preparation, improving understanding and confidence. This guide has combined revision with tips and more practice questions than you could shake a stick at. All the essential ingredients for getting a grade you can be really proud of.

Each specification topic has been referenced and distilled into the key points to make in an examination for top marks. Questions on all topics assessing knowledge, application and analysis are all specifically and carefully devised throughout this book.



PG ONLINE

PG ONLINE

ClearRevise

AQA GCSE Computer Science 8525

PG ONLINE

Clear**Revise**

Illustrated revision and practice

AQA GCSE **Computer Science** 8525

ClearRevise™

AQA GCSE
Computer Science 8525

Illustrated revision and practice

Published by
PG Online Limited
The Old Coach House
35 Main Road
Tolpuddle
Dorset
DT2 7EW
United Kingdom

sales@pgonline.co.uk
www.pgonline.co.uk
2020

PG ONLINE

PREFACE

Absolute clarity! That's the aim.

This is everything you need to ace your exam and beam with pride. Each topic is laid out in a beautifully illustrated format that is clear, approachable and as concise and simple as possible.

Each section of the specification is clearly indicated to help you cross-reference your revision. The checklist on the contents pages will help you keep track of what you have already worked through and what's left before the big day.

We have included worked examination-style questions with answers for almost every topic. This helps you understand where marks are coming from and to see the theory at work for yourself in an examination situation. There is also a set of exam-style questions at the end of each section for you to practise writing answers for. You can check your answers against those given at the end of the book.

A free pack of over 30 Python solutions to accompany each of the programs listed in the book are available to download from pgonline.co.uk.

LEVELS OF LEARNING

Based on the degree to which you are able to truly understand a new topic, we recommend that you work in stages. Start by reading a short explanation of something, then try and recall what you've just read. This has limited effect if you stop there but it aids the next stage. Question everything. Write down your own summary and then complete and mark a related exam-style question. Cover up the answers if necessary, but learn from them once you've seen them. Lastly, teach someone else. Explain the topic in a way that they can understand. Have a go at the different practice questions – they offer an insight into how and where marks are awarded.

ACKNOWLEDGEMENTS

The questions in the ClearRevise textbook are the sole responsibility of the authors and have neither been provided nor approved by the examination board.

Every effort has been made to trace and acknowledge ownership of copyright. The publishers will be happy to make any future amendments with copyright owners that it has not been possible to contact. The publisher would like to thank the following companies and individuals who granted permission for the use of their images in this textbook.

Design and artwork: Jessica Webb / PG Online Ltd
Photographic images: © Shutterstock

First edition 2020
A catalogue entry for this book is available from the British Library
ISBN: 978-1-910523-25-4

Printed on FSC certified paper by Bell and Bain Ltd, Glasgow, UK.

MIX Paper from responsible sources FSC® C007785

THE SCIENCE OF REVISION

Illustrations and words

Research has shown that revising with words and pictures doubles the quality of responses by students.[1] This is known as 'dual-coding' because it provides two ways of fetching the information from our brain. The improvement in responses is particularly apparent in students when asked to apply their knowledge to different problems. Recall, application and judgement are all specifically and carefully assessed in public examination questions.

Retrieval of information

Retrieval practice encourages students to come up with answers to questions.[2] The closer the question is to one you might see in a real examination, the better. Also, the closer the environment in which a student revises is to the 'examination environment', the better. Students who had a test 2–7 days away did 30% better using retrieval practice than students who simply read, or repeatedly reread material. Students who were expected to teach the content to someone else after their revision period did better still.[3] What was found to be most interesting in other studies is that students using retrieval methods and testing for revision were also more resilient to the introduction of stress.[4]

Ebbinghaus' forgetting curve and spaced learning

Ebbinghaus' 140-year-old study examined the rate in which we forget things over time. The findings still hold power. However, the act of forgetting things and relearning them is what cements things into the brain.[5] Spacing out revision is more effective than cramming – we know that, but students should also know that the space between revisiting material should vary depending on how far away the examination is. A cyclical approach is required. An examination 12 months away necessitates revisiting covered material about once a month. A test in 30 days should have topics revisited every 3 days – intervals of roughly a tenth of the time available.[6]

Summary

Students: the more tests and past questions you do, in an environment as close to examination conditions as possible, the better you are likely to perform on the day. If you prefer to listen to music while you revise, tunes without lyrics will be far less detrimental to your memory and retention. Silence is most effective.[5] If you choose to study with friends, choose carefully – effort is contagious.[7]

1. Mayer, R. E., & Anderson, R. B. (1991). Animations need narrations: An experimental test of dual-coding hypothesis. *Journal of Education Psychology*, (83)4, 484-490.
2. Roediger III, H. L., & Karpicke, J.D. (2006). Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological Science*, 17(3), 249-255.
3. Nestojko, J., Bui, D., Kornell, N. & Bjork, E. (2014). Expecting to teach enhances learning and organisation of knowledge in free recall of text passages. *Memory and Cognition*, 42(7), 1038-1048.
4. Smith, A. M., Floerke, V. A., & Thomas, A. K. (2016) Retrieval practice protects memory against acute stress. *Science*, 354(6315), 1046-1048.
5. Perham, N., & Currie, H. (2014). Does listening to preferred music improve comprehension performance? *Applied Cognitive Psychology*, 28(2), 279-284.
6. Cepeda, N. J., Vul, E., Rohrer, D., Wixted, J. T. & Pashler, H. (2008). Spacing effects in learning a temporal ridgeline of optimal retention. *Psychological Science*, 19(11), 1095-1102.
7. Busch, B. & Watson, E. (2019), *The Science of Learning*, 1st ed. Routledge.

CONTENTS

Computational thinking and programming skills - Paper 1

Section 1

☑

Section 2

☑

Computing concepts - Paper 2

Section 3

☑

Section 4

☑

Section 5

Section 6

Section 7

Section 8

MARK ALLOCATIONS

Green mark allocations[1] on answers to in-text questions through this guide help to indicate where marks are gained within the answers. A bracketed '1' e.g. [1] = one valid point worthy of a mark. There are often many more points to make than there are marks available so you have more opportunity to max out your answers than you may think.

TOPICS FOR PAPER 1
COMPUTATIONAL THINKING AND PROGRAMMING SKILLS

Information about Paper 1

Written exam: 2 hours
90 marks
50% of GCSE

Specification coverage

Computational thinking, code tracing, problem-solving, programming concepts including the design of effective algorithms and the designing, writing, testing and refining of code.

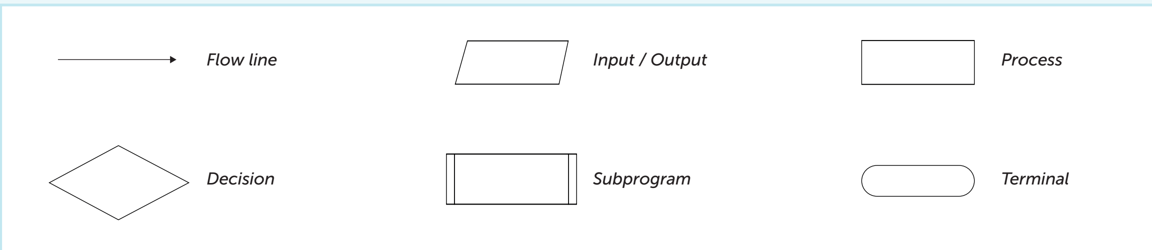
The content for this assessment will be drawn from subject content sections 3.1 and 3.2 of the specification.

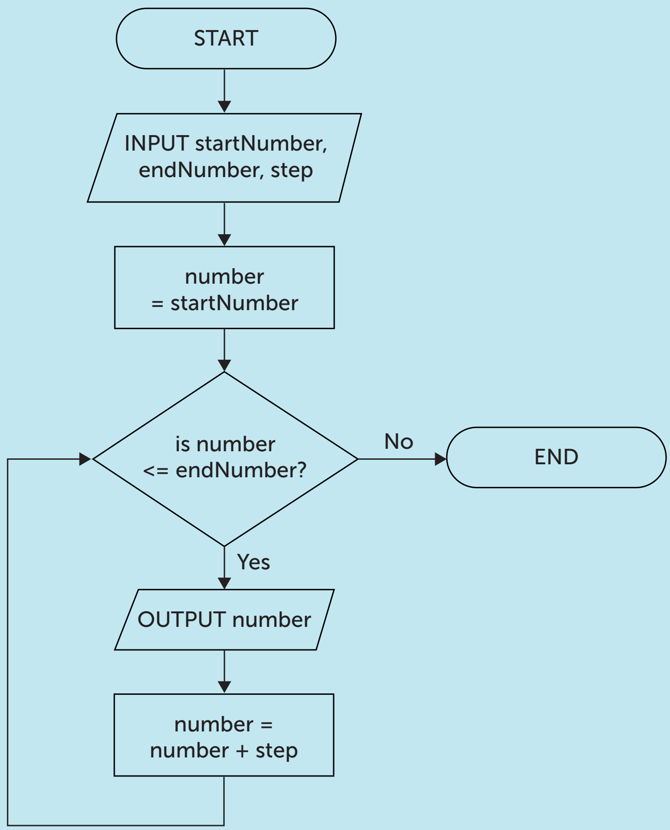
Questions

A mix of multiple choice, short answer and longer answer questions assessing programming, practical problem-solving and computational thinking skills.

USING FLOWCHARTS

Flowcharts are a useful tool that can be used to develop solutions to a problem. Standard flowchart symbols are shown below:





Look at the flowchart above.

(a) What will be output if the user enters 7, 50, 10 for the three values? [1]

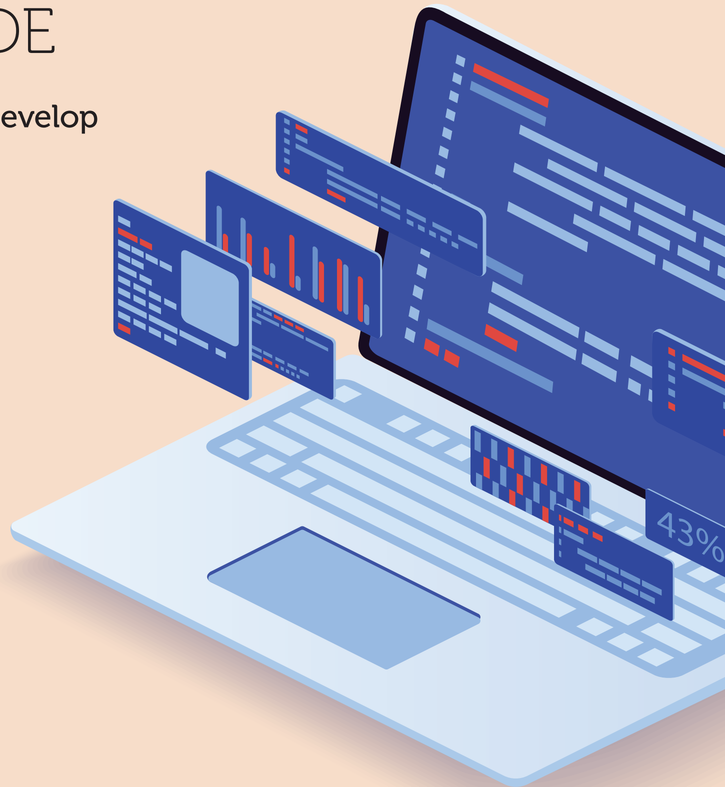
(b) What will be output if the user enters an end number which is less than the start number? [1]

(a) 7, 17, 27, 37, 47[1]

(b) Nothing will be output.[1]

3.1.1

USING PSEUDO-CODE



The problem with using a flowchart to develop an algorithm is that it does not usually translate very easily into program code.

Pseudo-code is useful for developing an algorithm using programming-style constructs, but it is not an actual programming language. This means that a programmer can concentrate on figuring out how to solve the problem without worrying about the details of how to write each statement in the programming language that will be used.

Using pseudocode, the algorithm shown in the flowchart above could be expressed like this:

```
input startNumber, endNumber, step
set number to startNumber
while number <= endNumber
    output(number)
    add step to number
endwhile
```

AQA standard pseudo-code

AQA has published a standard version of pseudo-code. This is defined in a file that can be downloaded from the AQA website. In an exam, where students are given pseudo-code, AQA will use the AQA standard version.

You do not have to use the AQA style of pseudo-code in your own work, when answering questions or describing algorithms. You will be awarded marks as long as your code is clear and consistent. **You should not use plain English or bullet points when describing algorithms.**

Some questions in the exam specify that you must use either a flowchart, pseudo-code or a high-level programming language you have studied to write or complete a program. Marks are awarded for correctly using syntax to represent programming constructs, whichever language you use. Answers written in pseudo-code, natural English or bullet points will not be awarded marks.

The algorithm shown above written in AQA standard pseudo-code would be written:

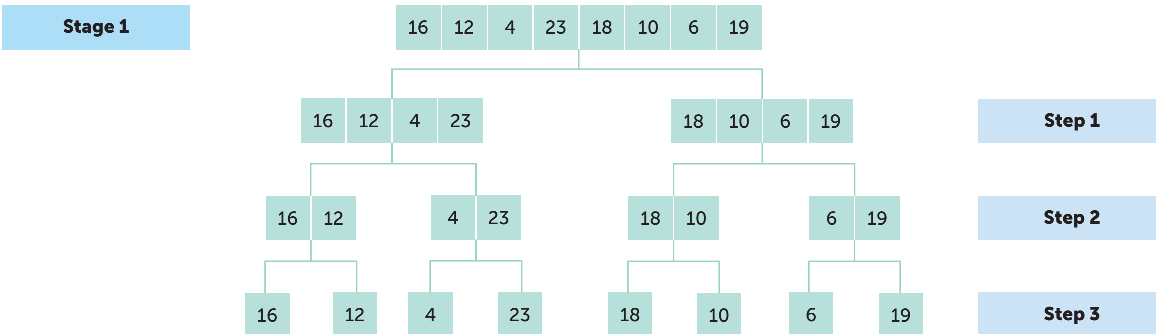
```
startNumber ← USERINPUT
endNumber ← USERINPUT
step ← USERINPUT
number ← startNumber
WHILE number ≤ endNumber
    OUTPUT number
    number ← number + step
ENDWHILE
```

Note that if there are three values to be input, when writing your own pseudo-code you should write three separate INPUT statements. Each INPUT statement is used to input a single value and assign it to a variable.

3.1.4

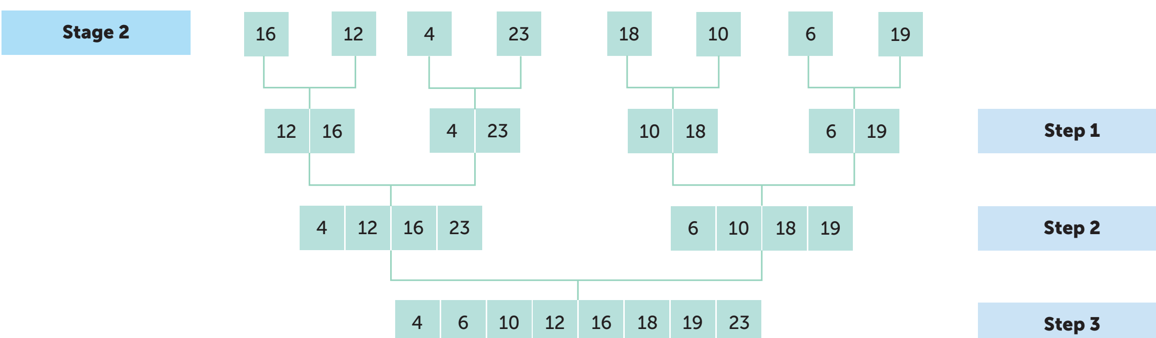
MERGE SORT

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



At the end of stage 1, 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. This is the sorted list.



1. Write the list that results from merging the two lists 2, 5, 17, 38, 56 and 3, 4, 15, 19, 36 [1]
2. The following list is to be sorted using a merge sort algorithm.

Giraffe	Zebra	Monkey	Leopard	Hippo	Warthog	Rhino

(a) Describe the two stages of a merge sort algorithm. [4]

(b) Write out the list after Step 2 of the Stage 2 process. [2]

1. The list would be: 2, 3, 4, 5, 15, 17, 19, 36, 38, 56[1]

2. (a) Stage 1: The list is successively divided in half[1], forming two sublists[1], until each sublist is of length one[1].

Stage 2: Each pair of sublists[1] is repeatedly merged[1] to produce new sorted sublists[1] until there is only one sublist remaining[1].

(b) Giraffe, Leopard, Monkey, Zebra,[1] Hippo, Rhino, Warthog[1]

EXAMINATION PRACTICE

1. A pseudocode algorithm is given below.

```
01  aList ← [3,6,7,9,13,15,16,19,20,24,26,29,36]
02  found ← False
03  n ← 0
04  x ← USERINPUT
05  WHILE found = False AND n < len(aList)
06      OUTPUT (aList[n])
07      IF aList[n] = x THEN
08      found ← True
09      ELSE
10          n ← n + 1
11      ENDIF
12  ENDWHILE
13  IF found = True THEN
14      OUTPUT(x, n)
15  ELSE
16      OUTPUT("invalid number")
17  ENDIF
```

(a) At line 05, what is the value of len(aList)? [1]

(b) The user enters 9 at line 04. What is printed at line 06 the first 3 times the while…endwhile loop is performed? [3]

(c) State what will be printed at line 14 if the user enters the number 9. [1]

(d) Explain the purpose of this algorithm. [2]

2. (a) An array names holds n items. An algorithm for a bubble sort is given below. [2]

```
01  swapMade ← True
02  while swapMade = True
03      swapMade ← False
04      for index ← 0 to n - 1
05          if names[index] > names[index + 1] then
06              swap the names
07              swapMade ← True
08          endif
09      next index
10  endwhile
```

(a) Explain the purpose of the variable swapMade in the algorithm. [2]

(b) Write the code for "swap the names" in line 06. [2]

(c) The list names contains the following:

Edna	Adam	Victor	Charlie	Jack	Ken	Maria

Write the contents of the list after each of the first two times the while…endwhile loop is executed. [2]

(d) How many times will the while loop be executed before the program terminates? Explain your answer. [2]

TWO-DIMENSIONAL ARRAYS

An array may have two or more dimensions. A 2-dimensional array named sales could hold the number of properties sold each quarter (Jan–March, April–June, July–September, October–December) by three different branches of an estate agent.

	Index	0	1	2	3
	0	56	87	92	43
Three branches	1	167	206	387	54
	2	22	61	52	14

The index for both row and column of the array starts at 0. The array is defined with the statement `array sales[3,4]`. The number of properties sold in Quarter 4 by Agent 1 is held in `sales[0,3]` and has the value 43.

1. The three branches of the estate agency are known as Branch A, Branch B and Branch C.
   (a) Write code to output sales figure for Branch C for the period April–June. [1]
   (b) What will be output? [1]
2. Write a program to ask a user to enter the name and five race times in seconds for each of 3 competitors, and display the average time for each competitor. [8]

1. (a) `print(sales[2,1])`[1] *(b) 61*[1]

2.

```
name = ["","",""][1]
totalTime = [0,0,0][1]
averageTime = [0,0,0][1]
raceTime = [ [0,0,0,0,0],
             [0,0,0,0,0],
             [0,0,0,0,0] ][1]
for c in range(3):[1]
  name[c] = input("Enter competitor name: ")[1]
  for race in range(5):[1]
     raceTime[c][race] = int(input("Enter race time: "))[1]
     totalTime[c] = totalTime[c] + raceTime[c][race][1]
     averageTime[c] = totalTime[c] / 5[1]
  print("Average race Time for ",name[c], averageTime[c])[1]
```

THE STRUCTURED APPROACH

Decomposition of a problem involves breaking down a problem into subroutines or **modules**. This helps to produce **structured code**.

The structured approach includes **modularised programming** using **parameters and local variables**. **Clear**, **well-documented code** should include **comments** to explain what the code is intended to do.

Using subroutines in programs has many advantages

- Makes debugging and maintaining the program easier as subroutines are usually no more than a page of code and are separate from the main program
- Subroutines can be tested separately and shown to be correct
- A particular subroutine can be called several times in the same program, and may also be saved in a subroutine library to be used in other programs

Python has a library of useful modules which can be imported into a program.

A subroutine may declare its own **local variables**, which exist only while the subroutine is executing. They are only accessible within the subroutine. This is important because if the value of a variable having an identical name in the main program is changed, this will not affect the local variable in the subroutine.

Generating a random number

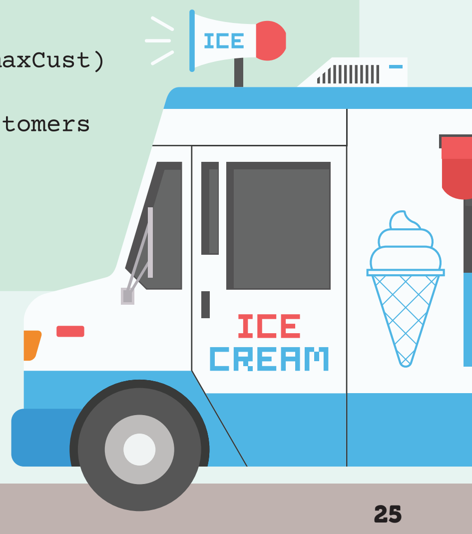
The randint() function generates a random number. To use it, you must first import the Python library module random by writing the statement import random at the start of the program. Then, to generate a random number num between a range of integers a and b:

```
num = random.randint(a,b)
```

Random numbers are often used in modelling. For example, suppose an ice cream van is visited by between 100 and 500 people each day during a given period. The owner wants to model the total number of customers, assuming a random number of customers in that range each day.

```
import random                 #import a library module
def totalFootfall(minCust, maxCust, days):
    totalCustomers = 0
    for day in range(days):
        dailyCustomers = random.randint(minCust, maxCust)
        print(dailyCustomers)
        totalCustomers = totalCustomers + dailyCustomers
    return totalCustomers

customers = ..................................
print ("Total customers for period: ",customers)
```



Look at the code above. Complete the statement to call the function `totalFootfall()` for a 30-day period. [2]

totalFootfall(100,500,30)[2]

TOPICS FOR PAPER 2
COMPUTING CONCEPTS

Information about Paper 2

Written exam: 1 hour 45 minutes
90 marks
50% of GCSE

Specification coverage

Fundamentals of data representation, computer systems, fundamentals of computer networks, cyber security, relational databases and structured query language (SQL).

Ethical, legal and environmental impacts of digital technology on wider society, including issues of privacy.

The content for this assessment will be drawn from subject content sections 3.3 to 3.8 of the specification.

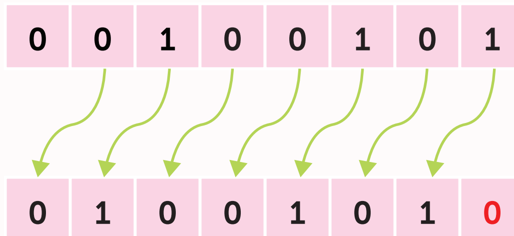
Questions

A mix of multiple choice, short answer, longer answer and extended response questions assessing SQL programming skills and theoretical knowledge.

BINARY SHIFTS

A binary shift moves all of the bits in a given binary number either to the left or the right by a given number of places. All of the empty spaces are then filled with zeros.

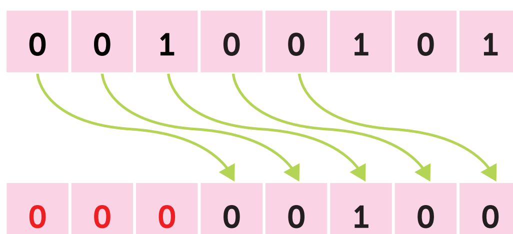
A shift of one place to the left will have the following effect:



Effects of shifts

A shift to the left will multiply a binary number by 2. Two shifts left would therefore multiply a number by 4. Each shift right would divide a number by 2. Similarly, a shift left in decimal of the number 17 becomes 170 and has therefore been multiplied by its base of 10.

An issue with precision occurs where odd numbers are divided since a standard byte cannot represent fractional numbers. Consider the following shift of three places to the right:



The original binary value was equal to decimal 37. A right shift should divide this by 8 (or divide by 2, three times). 37 / 8 = 4.625. However, the resulting binary converted to decimal is 4.

1. Complete a 2-place shift to the right on the binary number 11010110. [1]
2. Explain the effect of performing a right shift of two places on the binary number 11010110. [2]
3. Explain the effect of performing a left shift of 1 place on the binary number 11010110. [2]

1. 0011 0101[1]

2. Each shift right will divide the number by 2, so a two-place shift right will divide the number by 4[1]. However, if the shift results in a 1 being lost at the right hand end, the results will lose precision[1]. This is demonstrated in this question. 1101 0110 is 214 in decimal. Dividing that by 4 = 53.5. The shifted result 0011 0101 however is only 53 in decimal.

3. Shifting one place left multiplies the number by 2.[1] However this will cause overflow[1] for the given number, as 9 bits would be needed for the result[1], which is greater than 255[1].

REPRESENTING IMAGES

Similar to a mosaic, a bitmap image is made up of picture elements or pixels. A pixel represents the smallest identifiable area of an image, each appearing as a square with a single colour.

Image size

The size of an image is expressed as width × height of the image in pixels, for example 600 × 400px.

Number of colours		Colour depth
2 colours	2^1 colours	1 bit per pixel required
4 colours	2^2 colours	2 bit per pixel required
8 colours	2^3 colours	3 bit per pixel required
16 colours	2^4 colours	4 bit per pixel required

Colour depth

The first symbol below is represented in black and white using a series of binary codes. 0 = black and 1 = white.

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

11	11	11	00
10	10	10	10
10	10	01	10
10	10	01	10

Given that only 1 bit per pixel is available, only two colours, black and white, can be represented. The full image would have a size of 16 bits or 2 bytes. If the number of bits per pixel is increased, more colours can be represented. In the second example, four colours can be represented as the **colour depth** (also known as **bit depth**), or bits per pixel has been doubled to two. This will also double the file size.

1. Study the coloured bitmap images above.
   (a) Give the binary representation for the top row of the second example. [2]
   (b) State the colour depth of an image if a palette of 256 colours per pixel was required. [1]
   (c) State the effect on file size on the first 4x4 pixel symbol above of increasing the numbers of available colours to 256. [1]

(a) 11 11[1] 11 00[1]. One mark per correct pair.

(b) 8 bits per pixel.[1] ($2^8 = 256$)

(c) The file size would increase[1] to 1 byte per pixel, or 16 bytes for the whole icon, from 16 bits or 2 bytes[1].

HUFFMAN CODING

Huffman coding is a technique used to reduce the number of bits to represent letter in a body of text. The more frequently a character appears, the fewer bits are used to represent it and the higher up the tree it will appear.

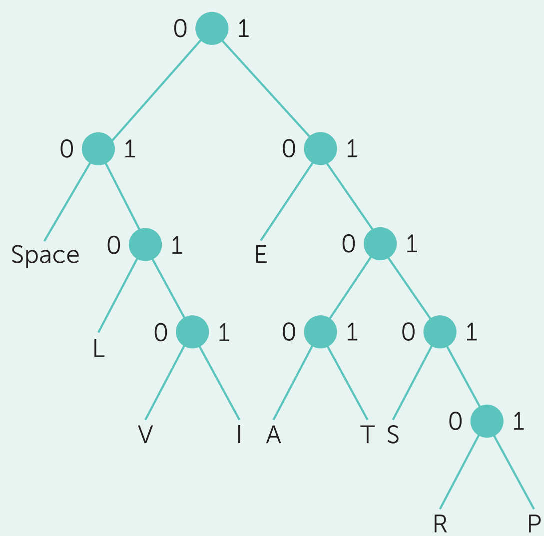
Example

Consider the sentence: **EVIL RATS STEAL LIVE PET**. The frequency that each letter in the sentence appears is first recorded in a table.

Character	Space	E	L	V	I	A	T	S	R	P
Frequency	4	4	3	2	2	2	2	2	1	1
	00	10	010	0110	0111	1100	1101	1110	11110	11111

A tree is then formed using the most frequent letters nearer the top of the tree. You will only be required to interpret a tree. You will not need to build one.

Using this tree, each character code is derived from the route taken to it. Left is 0, right is 1. 'A' is therefore represented as right, right, left, left or 1100. The word 'RAIL' would be encoded as 11110 1100 0111 010. This is 16 bits or 2 bytes. In ASCII, 'RAIL' would be stored as 4 bytes representing a 50% reduction in size.



This question uses the Huffman tree given above.

(a) State the coding for the word L,E,T. [3]

(b) Calculate how many bits these three letters would require using the Huffman code. [1]

(c) The sentence EVIL RATS STEAL LIVE PET is represented in a total of 79 bits.
Calculate how many bytes would be required to represent the sentence using ASCII. [1]

(d) Calculate how many bits are saved by compressing the sentence using Huffman coding instead of ASCII. [1]

(a) 10, 01, 10[1], 11, 01, 00[1]

(b) $2^8 = 256$[1]

(c) 3 Hz[1] (3 samples per second.)

(d) 8,000 samples per second, taken at 8 bits each = 64 kilobits / 8 = 8 kB x 3 seconds duration = 24 kB.[1]

3.4.1

HARDWARE AND SOFTWARE

A computer system is made up of hardware and software. Hardware is any physical component that makes up a computer. Software is any program that runs on the computer. You can touch hardware. You cannot touch software.

Computer systems are all around us. They are not just the PCs on the desk but include mobile phones, cash machines, supermarket tills and the engine management systems in a modern-day car.

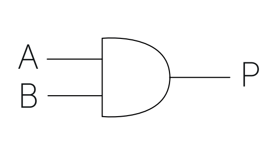
3.4.2

BOOLEAN LOGIC

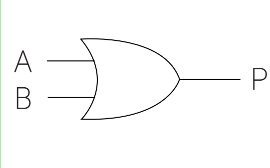
Simple logic gates

Computers are made of **logic gates**, **transistors** and **switches** which can be in one of two states: on or off, 1 or 0. You need to know about four simple logic gates that are used in electronics. Each is represented by a diagram and a truth table showing the possible outputs for each possible input.

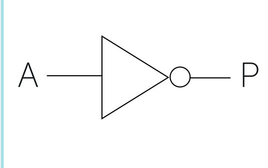
AND gate



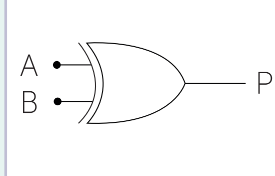
OR gate



NOT gate



XOR gate



AND			OR			NOT		XOR		
A	B	P = A AND B	A	B	P = A OR B	A	P = NOT A	A	B	P = A XOR B
0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	1	1	1	0	0	1	1
1	0	0	1	0	1			1	0	1
1	1	1	1	1	1			1	1	0

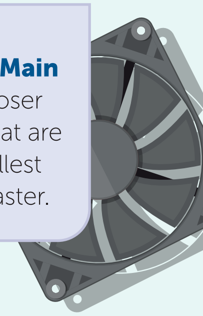
An XOR gate represents the **exclusive OR**. The output is True, or 1, when either input is 1, but not when both inputs are 1.

COMMON CPU COMPONENTS AND THEIR FUNCTION

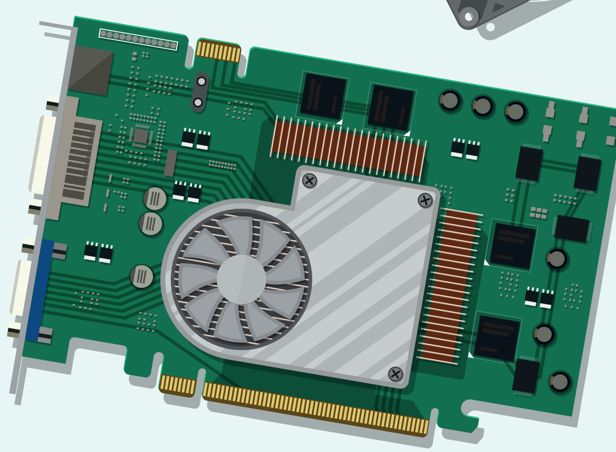
CPU component	Function
ALU (Arithmetic Logic Unit)	Carries out mathematical and logical operations including AND, OR and NOT, and binary shifts. It compares values held in registers
CU (Control Unit)	Coordinates all of the CPU's actions in the fetch-decode-execute cycle and decodes instructions. Sends and receives control signals to and fetch and write data
Clock	The clock regulates the speed and timing of all signals and computer functions
Registers	Even smaller and faster than cache memory, registers are built into the CPU chip to temporarily store memory addresses, instructions or data. They are used in the fetch-execute cycle for specific purposes
Bus	A collection of wires used to transfer data and instructions from one component to another

Types of memory

There are three types of memory. They all lose their contents when the power is switched off. **Main memory**, (or **RAM**), is largest. **Cache** is expensive, relatively small, super-fast memory that is closer to the CPU than RAM. It is used to hold recently used blocks of data or program instructions that are likely to be needed again. Some CPUs have different levels of cache with level 1 being the smallest and fastest. **Registers** are a special memory location on the CPU. They are even smaller and faster.

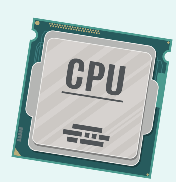


John von Neumann developed the **stored program computer**. The von Neumann architecture involves storing both programs and the data they use in memory.



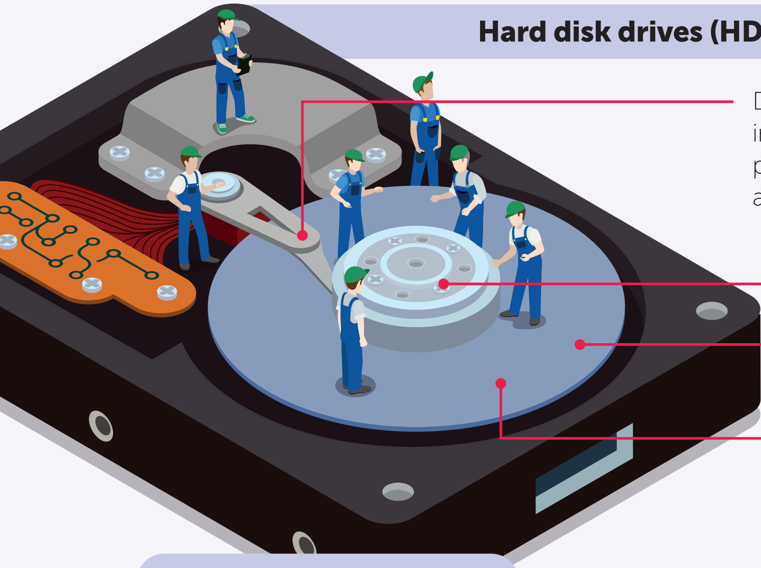
Explain how increased cache memory can improve the performance of the Central Processing Unit (CPU). [2]

Frequently used data or instructions are stored in cache[1] *so that the CPU does not need to fetch them from RAM*[1]*. Cache is quicker for the CPU to access than RAM.*[1]



DEVICE OPERATION

Hard disk drives (HDD)



Drive read/write head moves into position, like a record player. This movement takes additional time.

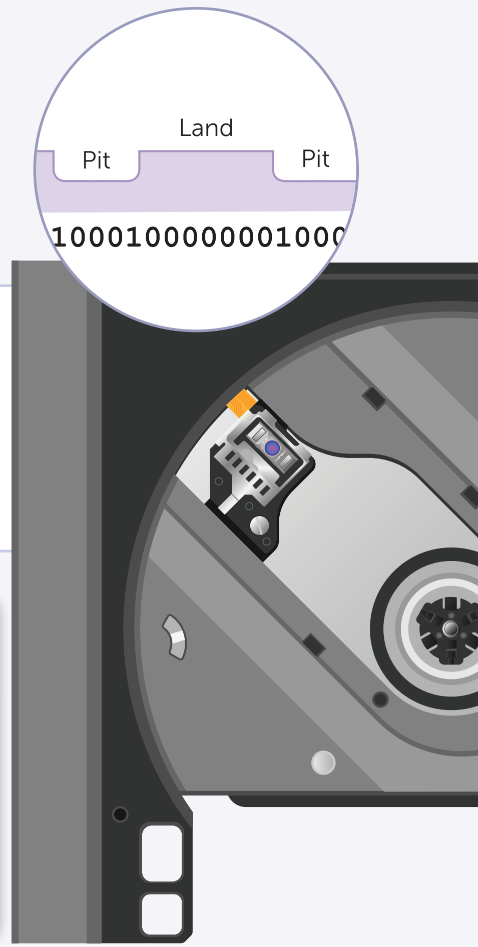
Drive spindle rotates disk. Moving parts cause issues if dropped.

Magnetic platter stores data. Affected by heat and magnetic fields.

Iron particles on the disk are magnetised to be either north or south, representing 0 or 1.

Solid state disks (SSD)

SSDs look like a standard circuit board.
They use electrical circuits to persistently store data. These use microscopic transistors to control the flow of current. One that allows current to flow is a 1. Where current is block, a 0 is represented.



Optical drives

An optical drive uses a laser to reflect light off the surface of the disk. Pits are burned into one long spiral track on the surface of the disk. When the laser light hits the curved end of a pit, the light is refracted and a 1 is recorded. Where light is reflected back directly from the flat bottom of a pit, or from an area of the track with no pit (a land) a 0 is recorded.

Explain why hard disk drives have been largely replaced by solid state drives in portable devices. [4]

Hard disk drives have lots of moving parts[1] *which can cause problems if dropped or shaken*[1]*. The read/write head moves across the disk and can scratch the disk irreparably if accidentally moved too violently whilst in operation.*[1] *Moving the head across the disk to read or write data reduces the access speed*[1] *that can be achieved with solid state devices that have no moving parts. The cost and capacity of solid-state storage is improving.*[1]

NETWORK SECURITY

Networks require security measures to prevent unauthorised access. This ensures the privacy of data that is transferred within the network. Using a combination of methods provides greater protection against threats.

Authentication

Authentication of an individual is used to make sure that a person is who they say they are.

Most commonly, this is done by asking a user to enter their **ID and password** or **PIN**. This is then compared with the stored password on a database to authenticate it. A simple authentication routine is used when you log into a school network, or an online shopping site.

Email confirmation is frequently used to confirm that a user has access to the email address they may have used to register with a website. An email with an activation link or code is sent to the email address that users can click to confirm they have access to the email inbox.

Biometric methods of authentication include optical, facial or fingerprint recognition. These use a person's physical features to confirm their identity.

Firewall

A firewall is a software or hardware device that monitors all incoming and outgoing network traffic. Using a set of rules, it decides whether to block or allow specific data packets. By opening and closing ports, it can block traffic from disallowed connections from accessing the network, as well as blocking communications from the network, to make sure that only authorised traffic is permitted.

MAC address filtering

A (**Media Access Control**) **MAC address** is a **unique hexadecimal identification number** assigned to every **Network Interface Card** used in networked devices. Whilst an IP address can only get data to the Internet-facing router on a network, the router can then forward the data on to a specific device within a network using its MAC address.

MAC address: 30-A5-BD-6F-C4-63

A **MAC address filter** allows access to, or blocks specific devices from, accessing a network.

Explain how MAC address filtering is used to restrict access to a network. [2]

The address of each device attempting to connect to a network is looked up in a predetermined list of addresses allowed or denied by the router[1]*. The router then makes a decision.*[1]

RELATIONAL DATABASES

A **relational database** consists of two or more tables, connected to each other through the use of a common field.

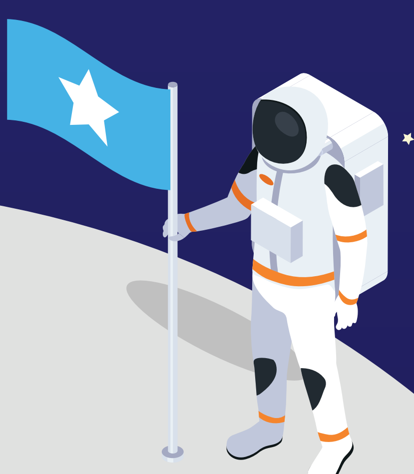
Each table will have a **primary key** field that uniquely identifies each record in the table.

Example

In the Mission table below, the primary key is **missionID**.

Mission

missionID	commander	launchDate	days	moonLanding	landingSite
Apollo 10	Thomas Stafford, Air Force	18/5/1969	8	No	No landing
Apollo 11	Neil Armstrong, Civilian	16/7/1969	5	Yes	Mare Tranquilitatis
Apollo 12	Charles Conrad, Navy	14/11/1969	5	Yes	Ocean of Storms
Apollo 13	Jim Lovell, Navy	11/4/1970	6	No	Aborted
Apollo 14	Alan Shepard, Navy	31/1/1971	6	Yes	Littrow crater
Apollo 15	David Scott, Air Force	2/8/1971	12	Yes	Censorinus crater
Apollo 16	John Young, Navy	16/4/1972	11	Yes	Descartes Highlands
Apollo 17	Gene Cernan, Navy	7/12/1972	12	Yes	Marius Hills
Skylab 2	Thomas Stafford, Air Force	25/5/1973	28	No	No landing



Relationships

A mission can have only one commander, but a commander can fly on many missions. This one-to-many relationship can be represented using an entity relationship diagram:

Other possible examples of relationships are many-to-one, one-to-one and many-to-many:

Data inconsistency and data redundancy

The data in this database contains **repeating fields** or **redundant data**. Some astronauts (Thomas Stafford) feature more than once in the table and therefore their details are unnecessarily repeated. This repeated data is redundant data. The individual instances of this repeating data could also result in **inconsistency** over time by updating different records with conflicting data for the same person, for example.

A further problem with this table is that the name held in the commander field is not in a format that can be easily searched. It actually holds three pieces of information. Each field in a database should be **atomic**, holding only one piece of data, e.g. surname, firstname, military service.

To avoid data redundancy and inconsistency, each data item should be held just once. This involves splitting the table into two – one for each **entity** or thing. In this case, the two entities are mission and astronaut. A link is then created between them both.

The new tables would look like this. The primary key from the Astronaut table (astronautID) is inserted into the Mission table. Here it is referred to as a **foreign key**.

Mission

missionID	astronautID	launchDate	days	moonLanding	landingSite
Apollo 10	1001	18/5/1969	8	No	No landing
Apollo 11	1002	16/7/1969	5	Yes	Mare Tranquilitatis
Apollo 12	1003	14/11/1969	5	Yes	Ocean of Storms
Apollo 13	1004	11/4/1970	6	No	Aborted
Apollo 14	1005	31/1/1971	6	Yes	Littrow crater
Apollo 15	1006	2/8/1971	12	Yes	Censorinus crater
Apollo 16	1007	16/4/1972	11	Yes	Descartes Highlands
Apollo 17	1008	7/12/1972	12	Yes	Marius Hills
Skylab 2	1001	25/5/1973	28	No	No landing

Astronaut

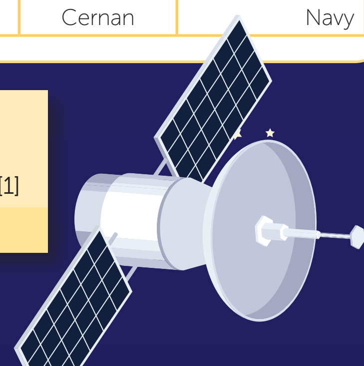
astronautID	firstname	surname	militaryService
1001	Thomas	Stafford	Air Force
1002	Neil	Armstrong	Civilian
1003	Charles	Conrad	Navy
1004	Jim	Lovell	Navy
1005	Alan	Shepard	Navy
1006	David	Scott	Air Force
1007	John	Young	Navy
1008	Gene	Cernan	Navy

1. The table **Misson** records Apollo moon missions.
   (a) Suggest a suitable data type for the data in the moonLanding field. [1]
   (b) Explain why missionID is the only suitable field for the primary key. [2]

(a) Boolean.[1] (b) missionID is the only field which contains data that will always be unique.[1] It would be possible for all data in other fields to be repeated[1] in other missions to the same sites, on the same days or using the same astronauts.

2. Suggest a suitable relationship type for the entities car and owner. [1]

Many to one.[1]



ETHICAL, LEGAL AND ENVIRONMENTAL IMPACTS OF DIGITAL TECHNOLOGY ON WIDER SOCIETY

Any advancement or implementation of a technology may introduce new ethical, legal, environmental and privacy issues.

Cyber security and hacking

Hacking means unauthorised access to programs or data. The growing use of communications technologies has vastly increased the volume of sensitive data that is sent electronically. People need to be increasingly careful with the data they send, and think about how they can avoid cyber threats. Software and hardware can be put in place to prevent hacking (unauthorised access).

Encryption is commonly used to secure data, but cyber criminals are also using the same security methods to protect their data from governments and law enforcement agencies. Should security services have access to everyone's encrypted data in order to protect the majority from the few?

Some governments argue that they should have access to all encrypted data sent to or from their citizens. Discuss how this might impact governments and citizens. [4]

Governments would have greater control over the protection of their citizens[1] *if they are able to analyse communications that may relate to planned criminal activity or acts of terrorism*[1]*, for example. Citizens value their privacy*[1] *and may not like the fact that governments may be 'watching' or 'reading' their messages, emails and other communications.*[1]

Mobile technologies

Mobile technologies such as smartphones and laptops switch from one network to another almost seamlessly as they roam through various regions. Any data sent through these networks needs to be secure. An unprotected network connection can allow a **hacker** to intercept any data, including passwords, online shopping data and bank details. **Phishing** scams are increasingly happening via SMS messaging instead of standard email.

Police have access to mobile phone cell data. This can effectively be used to track the movement of a phone. Like a breadcrumb trail, each time a phone makes contact with the nearest mobile mast it gives away its location. This data, and the communications to and from a device, are commonly used in crime prevention and as evidence in **legal** cases.

Wireless networking

Users of an unsecured connection should be very wary of what data they are sending. There may be **eavesdroppers** within the network looking to capitalise on sensitive data that may be sent.

EXAMINATION PRACTICE ANSWERS

Section 1

1. (a) 13 (the number of items in the list) [1]
   (b) 3 6 7 [3]
   (c) 9 found at position 3 (item 9 is the 4th in the list, counting from 0) [1]
   (d) It performs a linear search on the list for an item entered by the user. If the item is not found, it prints "Invalid number). [2]

   Download the program solutions in Python from www.pgonline.co.uk

2. (a) It acts as a 'flag' which is set to False when a pass through the list is made and no items are swapped, meaning that the list is now sorted. [2]
   (b)
   ```
   temp = names[n]
   names[n] = names[n+1]
   names[n+1] = temp
   ```
   [3]
   (c) Adam Edna Charlie Jack Ken Maria Victor
   Adam Charlie Edna Jack Ken Maria Victor [2]
   (d) 3 passes. Swaps are made on the first two passes. The list will be sorted after the second pass, and on the third pass, no swaps are made, so swapMade is set to False and the while loop terminates. [2]

3. (a) Algorithmic thinking [1]
   (b) Decomposition [1]
   (c) Abstraction [1]

4. (a) [4]

num	a	b	ans
	0	0	0
3	3	1	0
8	11	2	0
2	13	3	0
5	18	4	0
–1			4.5

   (b) It calculates the average of the numbers input by the user. [1]

Section 2

1. (a) String (b) Boolean (c) integer (d) real/float [4]

2. (a) line 02. [1]
   (b) Should be: `height = float(input())` [2]
   (c) Line 03 [1]
   (d) `if height < 1.2:` [1]

3. (a) False (n.b. do not accept "False" – this is a Boolean variable, not a string) [1]
   (b) 1 [1]
   (c) 5 [1]

4. (a) `not("7" in str(count))` (***Tip:*** *you must convert the integer 7 and the integer count to strings*) [3]
   (b)
   ```
   for count in range (100):
       if not("7" in str(count))and not(count mod 5 == 0):
           print (count)
   (or, if not("7" in str(count))and (count mod 5 != 0):
   ```
   [3]

BAND DESCRIPTIONS AND LEVELS OF RESPONSE GUIDANCE FOR EXTENDED RESPONSE QUESTIONS

Questions that require extended writing use mark bands. The whole answer will be marked together to determine which mark band it fits into and which mark should be awarded within the mark band.

Mark Band 3 – High Level (6–8 marks)

- Technical terms have been used precisely
- The answer is logical and shows an extensive understanding of Computer Science concepts, and principles
- The answer is almost always detailed and accurate
- All parts of the answer are consistent with each other
- Knowledge and ideas are applied to the context in the question
- Where examples are used, they help with understanding the answer
- Arguments and points are developed throughout the answer with a range of different perspectives. Different sides of a discussion are considered against each other

Mark Band 2 – Mid Level (3–5 marks)

- The meaning of technical terms in the question has been understood
- The answer shows an understanding of Computer Science concepts
- Arguments and points are developed in the answer, but sometimes useful examples or related knowledge to the context have not been included
- Some structure has been given to the answer with at least one line of reasoning
- Sound knowledge has been effectively shown

Mark Band 1 – Low Level (1–2 marks)

- The answer shows that technical terms used in the question have not been understood
- Key Computer science concepts have not been understood and have not been related to the context of the question
- The answer is only loosely related to the question and some inaccuracies are present
- Gaps are shown in Computer Science knowledge
- The answer only considers a narrow viewpoint or one angle
- The answer is unstructured
- Examples used are mostly irrelevant to the question or have no evidence to support them

0 marks

- No answer has been given or the answer given is not worth any marks

The above descriptors have been written in simple language to give an indication of the expectations of each mark band. See the OCR website at **www.ocr.org.uk** for the official mark schemes used.

EXAMINATION TIPS

With your examination practice, use a boundary approximation using the following table. Be aware that boundaries are usually a few percentage points either side of this.

Grade	9	8	7	6	5	4	3	2	1
Boundary	90%	80%	75%	65%	60%	50%	35%	25%	15%

1. Read questions carefully as some students give answers to questions they think are appearing rather than the actual question.
2. In calculation questions, marks are often given for working. Students should make sure to show their working in case they make a mistake and the answer is incorrect.
3. Algorithms can be given as pseudo-code or flowcharts unless the question explicitly states otherwise. If you make a mistake when drawing flowchart symbols, you are unlikely to be penalised unless you make the algorithm unclear.
4. Arrows coming out of decision symbols must be labelled to make an algorithm clear.
5. If candidates need to produce pseudo-code, then string ← USERINPUT will count as two statements – one for collecting user input, one for the assignment – this line of code may therefore be worth two marks.
6. Students may be asked to explain why one algorithm is better than another – for example, for a sorting algorithm. 'Quicker' and 'faster' are not acceptable answers. They must explain why the algorithm is quicker or faster to gain a mark.
7. This is also the case in programming code. If a more efficient code change is made, it is not acceptable to describe the improvement as 'faster' or 'uses less storage' unless there is an explanation of why this is the case.
8. When drawing logic gate diagrams, students often use the incorrect symbols for gates.
9. If students are asked for an explanation of issues as they affect an organisation, many will give how they affect an individual rather than the organisation which limits marks available. Again, care needs to be taken with reading the question closely.
10. When performing binary arithmetic, students can use any method they wish – for example, if they wish, convert to decimal, perform the addition, then convert back, alternatively they can do the addition directly in binary which is often faster.
    The same applies for converting numbers between hexadecimal to binary where students, if they wish, can convert from hex to decimal and then to binary.
11. Common misconceptions about ROM are that it is usually used to store application software or that there is typically more ROM than RAM. Both these are incorrect.
12. Students should be careful with vague answers. For cloud storage benefits it is not acceptable to write that it 'has more space' or 'costs less'. Correct answers would be 'it allows access to a larger amount of storage capacity' or 'it allows the purchase of a cheaper computer with less storage capacity'.
13. When giving the differences of WANs and LANs many students will say that 'WANs are larger'. This isn't acceptable. They need to say that a WAN links one remote geographical site/location to another.
14. Many students, especially weaker ones, do not have their answers match the context of the question. For example – a travel agent that stores customer and business data electronically and needs to prevent infections from malware. Some students would mention keeping records on paper or disconnecting computers from the network. These would help to prevent infections but are not appropriate for the type of business and therefore are not given marks.