



Cambridge
IGCSE[®]

Computer Science

**STUDY AND
REVISION GUIDE**



David Watson
Helen Williams



Cambridge
IGCSE®

Computer Science

STUDY AND REVISION GUIDE

David Watson
Helen Williams

Hachette UK's policy is to use papers that are natural, renewable and recyclable products and made from wood grown in sustainable forests. The logging and manufacturing processes are expected to conform to the environmental regulations of the country of origin.

Orders: please contact Bookpoint Ltd, 130 Milton Park, Abingdon, Oxon OX14 4SB. Telephone: (44) 01235 827720. Fax: (44) 01235 400454. Email education@bookpoint.co.uk Lines are open from 9 a.m. to 5 p.m., Monday to Saturday, with a 24-hour message answering service. You can also order through our website: www.hoddereducation.com.

ISBN: 9781471868689

® IGCSE is the registered trademark of Cambridge International Examinations. This text has not been through the endorsement process.

The questions, example answers, marks awarded and/or comments that appear in this book were written by the authors. In an examination, the way marks would be awarded to answers like these may be different.

© David Watson and Helen Williams 2016

First published in 2016 by
Hodder Education,
An Hachette UK Company
Carmelite House
50 Victoria Embankment
London EC4Y 0DZ

www.hoddereducation.com

Impression number 10 9 8 7 6 5 4 3 2 1

Year 2020 2019 2018 2017 2016















All rights reserved. Apart from any use permitted under UK copyright law, no part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or held within any information storage and retrieval system, without permission in writing from the publisher or under licence from the Copyright Licensing Agency Limited. Further details of such licences (for reprographic reproduction) may be obtained from the Copyright Licensing Agency Limited, Saffron House, 6–10 Kirby Street, London EC1N 8TS.

Cover photo © Scanrail – Fotolia

Illustrations by Aptara, Inc.
Typeset by Aptara, Inc.
Printed in India

A catalogue record for this title is available from the British Library.

Contents

	Introduction	iv	
Section 1	Theory of computer science		REVISED
	1 Binary systems and hexadecimal	1	
	2 Communication and internet technologies	12	
	3 Logic gates and logic circuits	21	
	4 Operating systems and computer architecture	31	
	5 Input and output devices	39	
	6 Memory and data storage	49	
	7 High- and low-level languages	56	
	8 Security and ethics	60	
Section 2	Practical problem-solving and programming		
	9 Problem-solving and design	69	
	10 Pseudocode and flowcharts	77	
	11 Programming concepts	82	
	12 Data structures: arrays and using pre-release material	87	
	13 Databases	90	
	Answers	94	
	Index	105	

Introduction

Welcome to the Cambridge IGCSE® Computer Science Study and Revision Guide. This book has been written to help you revise everything you need to know for your Computer Science examinations, alongside the Cambridge IGCSE Computer Science Student's Book. Following the Computer Science syllabus, it covers all the key content along with sample questions and answers, key definitions, and practice exam questions.

How to use this book

Key objectives

A summary of the main information

Key definitions

Definitions of the key terms you need to know

Examiner's tips

Tips that give you advice to help you give the perfect answer

Common errors
Mistakes that students often make, and how to avoid them

2 Communication and internet technologies

Key objectives

- explain half-duplex and full-duplex data transmission
- explain serial and parallel data transmission
- explain asynchronous and synchronous data transmission
- explain Universal Serial Bus (USB) connections
- explain data bus
- explain common network request (ARP)
- explain domain naming
- explain cloud computing
- explain internet technologies

Key definitions

Term	Definition
Simplex	Data transmission in one direction only
Half-duplex	Data transmission in two directions but not at the same time
Full-duplex	Data transmission in two directions at the same time
Serial transmission	Data is sent one bit at a time over a single wire or channel
Parallel transmission	Several bits of data are sent down several wires or channels at the same time (e.g. 1 byte of data sent down 8 wires/channels)
Asynchronous transmission	Data being transmitted in an agreed bit pattern
Synchronous transmission	A continuous stream of data which is accompanied by timing signals generated by an external clock

Data transmission

Simplex, half-duplex and full-duplex

Simplex – data transmission in one direction only
Half-duplex – data transmission in two directions but not at the same time
Full-duplex – data transmission in two directions at the same time

Serial and parallel data transmission

Serial – data is sent one bit at a time over a single wire or channel.
Parallel – several bits of data are sent down several wires or channels at the same time (e.g. 1 byte of data sent down 8 wires/channels).

Asynchronous and synchronous data transmission

Asynchronous – this refers to data being transmitted in an agreed bit pattern, data is grouped together and sent with control bits known as the start/stop bits.
Synchronous – this is a continuous stream of data which is accompanied by timing signals generated by an external clock; the timing signals ensure the sender and receiver are synchronized with each other to ensure the bits are assembled into the correct bit patterns at the receiving end.

Universal Serial Bus (USB)

USB is an asynchronous serial data transmission method which has become an industry standard. When a device is plugged into a computer using one of the USB ports:

- the device is automatically detected
- a browser device is automatically recognized and a driver device is loaded
- if a new device is plugged in, the computer looks for the correct device driver or the user will be prompted to download the driver.

Common errors

- Students often confuse half-duplex and full-duplex data transmission.
- Students do not realise that serial and parallel is a very common error when referring to data transmission.

Sample question

1. Name the direction and method of data transmission being described below:

- data sent in one direction only, one bit at a time, along a single wire
- bytes of data being sent in two directions at the same time using eight channels

2. Indicate which of the following are forms of serial, data transmission or parallel data transmission (put a tick (✓) in the appropriate column):

Feature	Serial	Parallel
uses one wire or channel to transmit data	✓	
bytes will not be lost		✓
multiple channels between sender and receiver		✓
USB is an example of it	✓	
works for use with devices		✓

Student answer

1. a) (1) Simplex, serial
 (2) Full-duplex, parallel

2. Feature

Feature	Serial	Parallel
uses one wire or channel to transmit data	✓	
bytes will not be lost		✓
multiple channels between sender and receiver		✓
USB is an example of it	✓	
works for use with devices		✓

Examiner's comments

1. The above answer is correct but it is necessary to analyse the sentence carefully to split out the two parts of the data transmission.

2. Data transmission in one direction only – this is serial
 one bit at a time using a single wire – this is serial
 data being sent in two directions at the same time – this is full-duplex
 bytes of data – using eight channels – this is parallel

3. Questions where students need to be ticked should be treated with caution to avoid any errors. Read each statement carefully and check which category it falls into – it is usually any questions of common answers will average. Always remember to place only one tick for each answer unless told to do otherwise – if two ticks are given in any of the above rows, no marks will be awarded for that row.

IGCSE Computer Science Study and Revision Guide © David Watson and Helen Williams 2016

IGCSE Computer Science Study and Revision Guide © David Watson and Helen Williams 2016

It is illegal to photocopy this page

IT PROGRAMMING CONCEPTS

Programming techniques

Sequence

Statements are followed in sequence in the order of the statements in a program's response.

Selection

Selects an error method to change a program's flow to be performed according to given criteria. For example, using the figure, when the student fails, allowing them to re-attempt the question.

Repetition

Most programming languages support three types of repetition:

- a fixed number of repetitions
- as long as a certain condition is met
- until a certain condition is met

Totalling

Totalling is used with repetition with the total updated every time the loop repeats. Keeping a running total is one of the most frequently used programming techniques in many computer systems.

Counting

Counting is used with repetition with the counter increased by 1 every time the loop repeats. Counting the number of times a loop repeats is one of the most frequently used programming techniques in many computer systems.

Common errors

- Students often use 'while' for counting and 'do-while' for counting the number of times a loop repeats.
- Students often use 'while' for counting and 'do-while' for counting the number of times a loop repeats.

Sample question

2. Study the code for the program below and state the average of 10 random integers, and the number of the program executed.

```

10 TOTAL = 0
20 COUNT = 0
30 WHILE COUNT <= 10
40 INPUT NUMBER
50 TOTAL = TOTAL + NUMBER
60 COUNT = COUNT + 1
70 PRINT TOTAL/COUNT
80 END

```

Student answer

```

10 TOTAL = 0
20 COUNT = 0
30 WHILE COUNT <= 10
40 INPUT NUMBER
50 TOTAL = TOTAL + NUMBER
60 COUNT = COUNT + 1
70 PRINT TOTAL/COUNT
80 END

```

Examiner's comments

The student has used the correct programming technique for this question. The program is correct for all marks for marks 1 to 10 and is correct for marks 11 to 12. The student has used the correct programming technique for this question. The program is correct for all marks for marks 1 to 10 and is correct for marks 11 to 12.

Examiner's comments

1. Explain, using numbers, the difference between a variable and a constant when a program is being executed. (2 marks)

2. Write a program that checks the number of letters in the input program and a value of -1 for the number of letters should be 0. (8 marks)

3. Identify the errors in the following program code to print the highest value of 10 random input, and then counts the program correctly. (8 marks)

```

10 INPUT N
20 COUNT = 1
30 WHILE COUNT <= 10
40 INPUT NUMBER
50 IF NUMBER > NUMBER THEN N = NUMBER
60 COUNT = COUNT + 1
70 PRINT NUMBER
80 END

```

Student's answer
Model student answers to see how the question might be answered

Examiner's comments

Feedback from an examiner showing what was good, and what could be improved

Exam-style questions

Exam questions for you to try to see what you have learned

Sample question

Exam-style questions for you to think about

iv

IGCSE Computer Science Study and Revision Guide © David Watson and Helen Williams 2016

● Exam tips

Know what the examiners can ask you questions about.

Find a copy of the IGCSE Computer Science syllabus that you have been studying and are going to be taking the exam for and use your revision guide and textbook. The list of topics will make an excellent checklist for your revision. If you find something that you don't understand or haven't made any notes about, find out about it. *Anything* that is mentioned in the syllabus could be used in an examination question.

Read and understand examination questions.

What are you being asked to do?

- 1 Read the question.
- 2 Understand the type of instruction you are being given: Complete, Describe, Draw, Explain, Give and State all require different actions.
- 3 If the question makes use of a specific scenario then make sure that all your answers are relevant to that scenario. For example if the question is security measures for a bank's website then suggesting taking it offline outside banking hours would not be appropriate!
- 4 Decide on the information required but remember that you are sitting an IGCSE examination and most answers will require more than just a single word. If you have finished your examination well before the time allotted, you may have fallen into this trap.
- 5 Always use correct technical terms and avoid the use of trade names. Write about using an operating system rather than using 'Windows'.
- 6 Decide how much information is required.
- 7 Use the instructions e.g. Describe two benefits of using USB connections...
- 8 ...or look at the number of marks e.g. Write an algorithm to complete Task 2, [5 marks]: at least five statements needed here.

Know your subject.

- Make sure that you understand computer science terms and that you can spell them correctly e.g. validation, phishing, pharming etc.
- Learn the definitions and be able to provide examples to show your understanding. There are certain items which need learning so that you can apply them e.g. validation checks, types of translator.
- Don't ignore items in the syllabus that you don't understand or haven't got notes on. Do some research and ask.
- Beware of cramming facts into your brain without understanding them or being able to apply them. You need to revise but try and make it interesting rather than just reading and re-reading your notes. Practise examination questions, watch any relevant YouTube videos, use interactive revision programs etc.
- Don't try to learn too much at the last minute. Use the best revision techniques that suit you. If you understand something you can remember it far better than if you don't.
- Finally, complete the programming tasks for paper 2 and be able to explain what you have done and why you have done it.

Remember what an exam paper is for.

The examiners are trying to test your knowledge and understanding of the subject and your analytical skills.

- Examiners rarely put unnecessary words in exam questions so make sure that you read all of the question carefully.
- Make sure that you understand the work that you have done; if you only try to remember it then you will not be able to apply it.
- Mark allocations are there to help you to decide on how to use your time sensibly – so don't spend half an hour on a two mark question.

Help the examiner and yourself.

- Get a good night's sleep or take a decent lunch break before the exam! The most important thing that goes into the exam room is your brain and it doesn't function well if you haven't had enough rest. Think of exam preparation in the same way as preparation for taking part in a sporting event.
- Stay calm! Everyone gets nervous in some way so work out how best to relax yourself before you start writing.
- Read the whole paper through first. This is not a waste of time. It helps you to collect your thoughts and to get an idea of how long you can spend on each question. It also helps you to sort out which questions are on which topics.
- Read each question through completely before you start to answer it. This helps you to avoid problems like writing the answer needed for part b) in part a) of a question.
- Use clear English and good handwriting. If the examiner can't read or understand your answer then no marks!
- Don't use correction fluid, cross out neatly any answers that you don't want to include and show the examiner where your answer is e.g. answer on page 10.
- Don't cramp your answers – the better set out that they are the easier they are to mark.
- Read through your answers if you have time when you have finished. You just might think of something else to add in that could make the difference of a grade.
- Answer all of the questions. There is no point in leaving blank spaces. You just might get something down that is correct!

1

Binary systems and hexadecimal

Key objectives

Objectives of this chapter are to revise:

- binary systems
- memory size measurement
- use of binary registers
- hexadecimal system

● Key definitions □

Term	Definition
Denary system	Number system based on the number 10
Binary system	Number system based on the number 2
Hexadecimal system	Number system based on the number 16. Used for memory dumps, HTML, MAC addresses, web addresses and assembly code/machine code

● Binary system □

Converting binary numbers to denary numbers

Example

Convert 0 1 1 0 1 1 0 1 to denary.

Add up binary values where 1s appear:

128	64	32	16	8	4	2	1
0	1	1	0	1	1	0	1
	64 + 32			+ 8	+ 4		+ 1 = 109

Converting denary numbers to binary numbers

Example

Convert 147 to binary.

Method 1

Place 1-values in the correct positions until it adds up to 147:

128	64	32	16	8	4	2	1
1	0	0	1	0	0	1	1

Method 2

Successive division by 2:

2	147		↑
2	73	remainder: 1	
2	36	remainder: 1	
2	18	remainder: 0	
2	9	remainder: 0	
2	4	remainder: 1	
2	2	remainder: 0	
2	1	remainder: 0	
	0	remainder: 1	

read the remainder from bottom to top to get the binary number:
1 0 0 1 0 0 1 1

Measurement of memory size

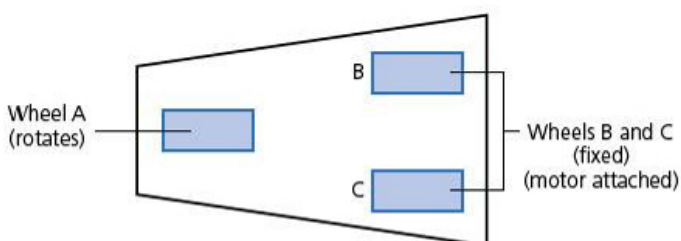
Name of memory size	Number of bits	Equivalent denary value
1 kilobyte (1 KB)	2^{10}	1 024 bytes
1 megabyte (1 MB)	2^{20}	1 048 576 bytes
1 gigabyte (1 GB)	2^{30}	1 073 741 824 bytes
1 terabyte (1 TB)	2^{40}	1 099 511 627 776 bytes
1 petabyte (1 PB)	2^{50}	1 125 899 906 842 624 bytes

NOTE: $1024 \times 1024 = 1\,048\,576$ and so on.

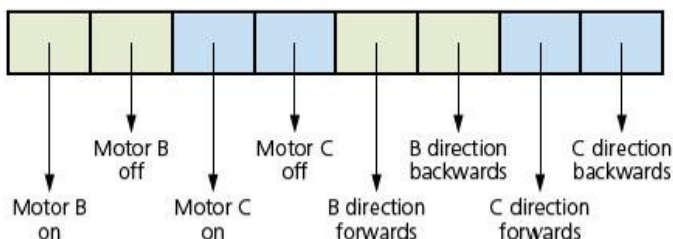
Binary registers

A register is a group of bits.

A robot vacuum cleaner has three wheels, A, B and C. A can turn through 360° ; B and C are fixed but have an electric motor attached:



An 8-bit register is used to control the movement of the robot vacuum cleaner:



If the register contains 1 0 1 0 1 0 1 0 this means 'motor B is ON and motor C is ON and both motors are turning to produce FORWARDS motion'. Effectively, the vacuum cleaner is moving forwards.

Common error

- Incorrect position values used in the number base systems leading to errors during number conversion.

Sample question

1 a Convert the following binary numbers into denary numbers:

(i) 0 0 1 1 0 1 1 1

(ii) 1 1 1 0 0 1 1 0

b Convert the following denary numbers into 8-bit binary numbers:

(i) 6 8

(ii) 2 4 0

Student answer

1 a (i) $32 + 16 + 4 + 2 + 1 = 55$

b (i) $68 = 01000100$

(ii) $128 + 64 + 32 + 4 + 2 = 230$

(ii) $240 = 11110000$

Examiner's comments

The above answers are correct, but setting out the answers like this could lead to errors since not all of the working is shown. It would be better to set it out as follows showing all bit positions:

a (i)

128	64	32	16	8	4	2	1	
0	0	1	1	0	1	1	1	
		32 + 16			+ 4	+ 2	+ 1 = 55	

(ii)

128	64	32	16	8	4	2	1	
1	1	1	0	0	1	1	0	
128	+ 64	+ 32			+ 4	+ 2 = 230		

b (i) 68

128	64	32	16	8	4	2	1	
0	1	0	0	0	1	0	0	

(ii) 240

128	64	32	16	8	4	2	1	
1	1	1	1	0	0	0	0	

Sample question

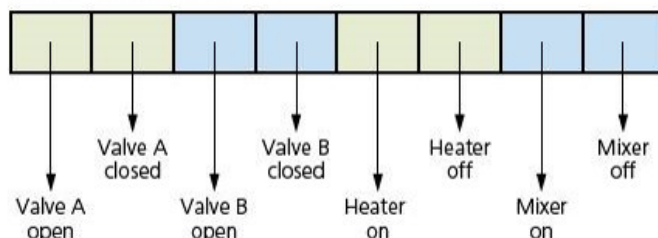


2 a (i) Convert 4096 kilobytes (KB) to megabytes (MB).

(ii) Convert 0.2 terabytes (TB) to gigabytes (GB).

b A chemical process is controlled by a microprocessor. The process utilises two valves, A and B, to control the chemical addition. There is also a mixer and a heater which can be turned on or off as required.

An 8-bit register is used to show the condition of the two valves and heater and mixer. A 1-value means the condition is true; a 0-value means the condition is false.



(i) What is indicated by: 1 0 1 0 0 1 1 0?

(ii) Give the register contents if valve A is open, valve B is closed, and the heater and mixer are both on.

(iii) What would the following register contents indicate: 0 1 0 1 0 1 0 1?

Student answer

- 2 a (i) 4 MB
 (ii) 204.8 GB
 b (i) valve A is open, valve B is open, the heater is off and the mixer is on
 (ii) 1 0 0 1 1 0 1 0
 (iii) everything is in the OFF mode

Examiner's comments

While both answers are correct, it would be better to show your working and set your answer out as follows:

- a (i) $4096/1024 = 4 \text{ MB}$ (since $1024 \text{ KB} = 1 \text{ MB}$)
 (ii) $0.2 \times 1024 = 204.8 \text{ GB}$ (since $1024 \text{ GB} = 1 \text{ TB}$)
- b (i) Because there is a 1-bit in positions one and three, this indicates that these two conditions (BOTH valves open) are true. Positions six and seven also have 1-bits, indicating that the heater is off and the mixer is on. Such detail is not necessary, but it is included here to explain to the reader how the answer to the question was arrived at.
- (ii) Valve A is open means that bit position one should contain a 1-bit and valve B is closed, which means bit position four also contains a 1-bit. The heater is on, which means bit position five contains a 1-bit and the mixer is on meaning bit position seven must contain a 1-bit. The other four bit positions will contain a 0-bit. This gives: 1 0 0 1 1 0 1 0. Again, this level of detail is not usually needed, but it explains to the reader how the 8-bit register is made up.

Examiner's tips

Not writing out the position values above each digit can lead to errors in the calculation (see question 1).

It is always a good idea to show all your working in any question involving a calculation; if an error is made in the final calculation then all your marks will be lost; but if your working is shown, marks can often still be awarded for showing how you attempted to do the calculation.

● Hexadecimal system

This is based on a 16-digit number system that uses the digits 0 to 9 and the letters A to F to represent each hexadecimal digit (A = 10, B = 11, C = 12, D = 13, E = 14, and F = 15).

Converting binary numbers to hexadecimal numbers

Example 1

Convert 1 1 0 0 0 0 1 1 1 0 1 0 to hexadecimal.

Convert each group of 4 binary digits to a hex value:

C 3 A

Example 2

Convert 1 1 1 1 1 1 1 0 1 1 0 0 0 0 to hexadecimal.

This time we don't have groups of four binary digits (14 bits in total). Split up into groups of four from the right going left and fill in with 0s at the very end where necessary. This gives:

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

Now convert each group to its hex value:

3 F B 0

Converting hexadecimal numbers to binary numbers

Example

Convert A 9 E to binary.

Convert each hex digit into a 4-bit binary group to give:

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

Converting hexadecimal numbers to denary numbers

Example

Convert 2 A F 4 into denary.

Remember the hex values of 4096, 256, 16 and 1 which gives:

$(2 \times 4096) + (10 \times 256) + (15 \times 16) + (4 \times 1)$

i.e. $8192 + 2560 + 240 + 4 = 10996$

Converting denary numbers into hexadecimal numbers

Example

Convert 7079 into hex.

Method 1

Place hex values in the correct positions until it adds up to 7079:

4096	256	16	1
1	B	A	7

Method 2

Successive division by 16:

16	7079		
16	442	remainder: 7	↑ read the remainder from bottom to top to get the hexadecimal number: 1 B A 7
16	27	remainder: 10 (A)	
16	1	remainder: 11 (B)	
16	0	remainder: 1	

Uses of the hexadecimal system

The hexadecimal system is used for:

- Memory dumps – when developing new software, this allows the contents of the memory to be seen by the writer, thus enabling errors to be detected; also used in diagnostics when a computer malfunctions; hex is used since it is easier to use than a long string of binary values.
- HTML – hypertext mark-up language which is used when writing and developing web pages; uses tags in the definitions, e.g. <h1>
- MAC address – a media access control address is used to uniquely identify a device on the internet (NIC card used to identify device); MAC addresses can be UAA or LAA.
- Web addresses – ASCII code (hex values) can be used to replace the URL, e.g. %77%77%77%2E%68%6F%64%65%72%20%63%6F%2E%75%6B can be used to replace: www.hodder.co.uk.
- Assembly code/machine code – using hex makes it easier, faster to write and less error-prone than writing the code in binary.



Common error

- Incorrect position values used in the number base systems can lead to errors during number conversion.



Sample question

3 a Convert the following binary numbers into hexadecimal numbers:

(i) 111100101011

(ii) 101011111101010

b Convert the following hexadecimal numbers into binary numbers:

(i) A14A

(ii) F9A8

c Convert the following hexadecimal numbers into denary numbers:

(i) 4A2F

(ii) 2DF5

d Convert the following denary numbers into hexadecimal numbers:

(i) 6940

(ii) 15120

Student answer

3 a (i) F2B	c (i) 18991
(ii) 57EA	(ii) 11765
b (i) 1010 0001 0100 1010	d (i) 1B1C
(ii) 1111 1001 1010 1000	(ii) 3B10

Examiner's comments

a While the two answers given are correct, you are strongly advised to show that the binary numbers are first split into groups of four bits giving:

(i) 1111 0010 1011

(ii) 0101 0111 1110 1010

You should also explain the need to add the additional 0 in the first 4-bit pattern, since only three bits were given initially. By doing this, you will indicate you understand the process of conversion from binary to hex.

b As in part a, it is a good idea to mention that each hex digit is equivalent to four binary digits; so the four hex digits in both parts will create a 16-bit binary value giving the two answers shown above.

c It is worth showing the hex headings here and each of the steps in the conversion. By doing this, an examiner will be able to see your thought processes and award marks even if the final answer is incorrect:

(i)

4096	256	16	1
4	A	2	F

gives $(4 \times 4096) + (10 \times 256) + (2 \times 16) + (15 \times 1)$
 $= 16384 + 2560 + 32 + 15 = 18991$

(ii)

4096	256	16	1
2	D	F	5

$$\begin{aligned} &\text{gives } (2 \times 4096) + (13 \times 256) + (15 \times 16) + (5 \times 1) \\ &= 8192 + 3328 + 240 + 5 = 11765 \end{aligned}$$

d As with parts a to c it is advisable to show your working. We use Method 2 here:

(i)

16	6940	
16	433	remainder: 12
16	27	remainder: 1
16	1	remainder: 11
	0	remainder: 1

↑
Read the remainder from bottom to top
to get the hexadecimal number:
1 B 1 C

(ii)

16	15120	
16	945	remainder: 0
16	59	remainder: 1
16	3	remainder: 11
	0	remainder: 3

↑
Read the remainder from bottom to top
to get the hexadecimal number:
3 B 1 0

● Sample question



4 Describe three applications of the hexadecimal system.

Student answer

1 Memory dumps

Since it is much easier to work with: **B5A41AFC**
rather than: **10111001101001000001101011111100**
hexadecimal is often used when developing new software or when trying to trace errors in programs. The contents of part of the computer memory can hold the key to help solve many problems. When the memory contents are output to a printer or monitor, this is known as a memory dump.

A program developer can look at each of the hexadecimal codes and determine where the error lies. The value on the far left shows the memory location so that it is possible to find out exactly where in memory the fault occurs. This is clearly much more manageable using hexadecimal rather than using binary. It's a very powerful fault tracing tool, but requires considerable knowledge of computer architecture to be able to interpret the results.

2 HyperText Mark-up Language (HTML)

HTML is used when writing and developing web pages. HTML isn't a programming language but is simply a mark-up language. A mark-up language is used in the processing, definition and presentation of text (for example, to specify the colour of the text).

HTML uses <tags> which are used to bracket a piece of code; for example, <td> starts a standard cell in an HTML table, and </td> ends it. Whatever is between the two tags has been defined.

3 Media Access Control (MAC) address

MAC address refers to a number which uniquely identifies a device on the internet. The MAC address refers to the network interface card (NIC) which is part of the device. The MAC address is rarely changed so that a particular device can always be identified no matter where it is.

A MAC address is usually made up of 48 bits which are shown as 6 groups of hexadecimal digits (although 64-bit addresses are also known):

$NN - NN - NN - DD - DD - DD$ or $NN:NN:NN:DD:DD:DD$

where the first half ($NN - NN - NN$) is the identity number of the manufacturer of the device and the second half ($DD - DD - DD$) is the serial number of the device.

4 Web addresses

Each character used on a keyboard has what is known as an ASCII code (American Standard Code for Information Interchange). These codes can be represented using hexadecimal values or decimal values.

A good example of the use of ASCII codes is the representation of a web address (or URL) such as:

`www.hodder.co.uk` which becomes (using hexadecimal values):

`%77 %77 %77 %2E %68 %6F %64 %64 %65 %72 %2E %63 %6F %2E %75 %62`
`w w w . h o d d e r . c o . u k`

5 Assembly code and machine code

The computer memory can be referred to directly using machine code or assembly code. This can have many advantages to program developers or when carrying out troubleshooting.

Using hexadecimal makes it much easier, faster and less error prone when writing code compared with binary. Using true machine code (which uses binary) is very cumbersome and it takes a long time to key in the values. It is also very easy to mis-type the digits in a 'sea of 1s and 0s.'

Examiner's comments

The answers given here are those you might expect from a top grade A student. To obtain a grade A for this question, it would not really be necessary to go into this amount of depth. Only three uses need to be described and between 6 and 9 marks would be allocated to the question – the higher mark indicating that each use requires three different points to be made. So which three points in each use would gain the maximum marks?

Memory dumps:

- 1st mark point: give a short example of binary and hex and indicate it is easier to work in hex.
- 2nd mark point: indicate that a memory dump shows a part of the computer's memory only.
- 3rd mark point: indicate that memory dumps are used by software developers to locate errors or by software engineers when the computer malfunctions or if the system become corrupted.

HTML:

- 1st mark point: used for writing and developing web pages.
- 2nd mark point: not a programming language, but a mark-up language.
- 3rd mark point: give some reference to the use of tags `<h1 >` or use in colour of font, e.g. `#FF00FF`

MAC address:

- 1st mark point: uniquely identifies a device on the network/internet.
- 2nd mark point: the MAC address refers to the interface card (NIC) in the device.
- 3rd mark point: show the MAC address format: $NN - NN - NN - DD - DD - DD$

Web addresses:

- 1st mark point: web addresses usually written as `www.hodder.co.uk`
 2nd mark point: it is possible to replace each character in the URL by hex: `%xx`
 3rd mark point: values used are taken from the ASCII code.

Assembly code/machine code:

- 1st mark point: machine code and assembly code are used by software developers.
 2nd mark point: hex is used to replace a string of binary values/bits.
 3rd mark point: since hex is much shorter than binary, it is both quicker and less error-prone when entering code.

The above answers indicate how it is possible to gain maximum marks by simply pointing out some of the key features of each use, rather than the need to go into long descriptions which won't gain additional marks. This is something to bear in mind when finalising your revision notes.

Examiner's tips

When converting from binary to hexadecimal, split the binary into groups of 4 starting from the right and working left; fill in with 0s if necessary in the last group.

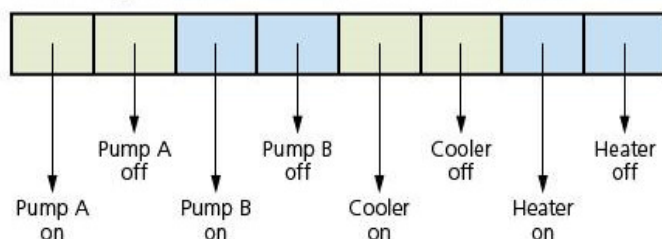
Conversely, use 4 binary bits to represent each hex digit when converting from hex to binary.

Always use the hex position values (4096, 256, 16 and 1) to avoid errors in the calculation when doing number base conversions.

When writing essay-type questions, remember to only pick out the key points in your answer; it is often a good idea to briefly write your main points in pencil in the margin or on additional paper to ensure you miss nothing out.

Exam-style questions

- 1 a Convert the following binary number into a hexadecimal number:
 1 1 0 1 1 0 0 0 1 1 1 1 1 1 0 [2 marks]
- b Convert the following hexadecimal number into a binary number:
 A 4 F [2 marks]
- c Convert the following denary number into a hexadecimal number:
 6 0 0 0 [2 marks]
- 2 a A chemical process uses two pumps (A and B), a cooler and a heater to control the conditions for the chemical reactions. The pumps can be either on or off and the cooler or heater can also be on or off. An 8-bit register is used to show the condition of all four components. A 1-value in the register indicates that the condition is true.



- (i) What conditions do the following register contents indicate?
 0 1 1 0 0 1 1 0 [2 marks]

- (ii) What would be the contents of the 8-bit register if the following conditions occurred?
- pump A is on
 - pump B is on
 - cooler is on, and the heater is off
- [2 marks]
- (iii) What binary value could be shown in the 8-bit register to show an error condition?

Explain your choice of value. [2 marks]

- b A computer is used to store photographs and videos. The average size of each photograph is 8MB and the average size of each video is 400MB,
- (i) How much memory is needed to store 800 photographs and 80 videos? Give the answer in MB. [2 marks]

- (ii) What is the value from part b(i) in gigabytes (GB)? [1 mark]

- 3 a Which computer terms are being described here?

- (i) a number which uniquely identifies a device on the internet
- (ii) when the memory contents of a computer are output to a printer or monitor
- (iii) a mark-up language used when writing and developing web pages
- (iv) items such as www.example.co.uk used to locate web pages
- [4 marks]

- b MAC addresses can be UAA or LAA. Explain the difference between the two terms. [4 marks]

- 4 a A system stores messages as 4-bit binary codes. Each of the 4 bits represents a hexadecimal digit. Convert each of the following 4-bit codes and indicate which message was stored.

1 1 0 0

1 0 1 0

1 1 1 1

1 1 1 0

1 1 0 0

0 0 0 1

1 0 1 0

0 0 0 0

[6 marks]

- b To increase the number of letters that can be used, a 12-bit register is to be used. In this new system A = 1, B = 2, ... , Y = 25 and Z = 26. The five rightmost positions are used initially to represent the letter.

The letter U (21st letter) would be stored as follows:

0	0	0	0	0	0	0	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---

- (i) Show how the letters L (12th letter) and W (23rd letter) would be stored in a 12-bit register:

L:

--	--	--	--	--	--	--	--	--	--	--	--

W:

[2 marks]

--	--	--	--	--	--	--	--	--	--	--	--

- (ii) Which letter is represented here?

[1 mark]

0	0	0	0	0	0	0	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---

- c To improve security, the 5 bits on the right are shifted 1, 2, 3 or 4 places to the left and can then occupy up to 9 bits in total:

--	--	--	--	--	--	--	--	--	--	--	--

possible expansion

original bits

To indicate how many bits the letter was shifted, the leftmost three bits can contain the value 0, 1, 2, 3 or 4. Thus if the letter U had been shifted FOUR places to the left, the register would now contain the following:

1	0	0	1	0	1	0	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

possible expansion

original bits

- (i) Shift your contents for letter W in part b(i) THREE places LEFT.

Show the contents of the register after this operation has been done:

[2 marks]

--	--	--	--	--	--	--	--	--	--	--	--

- (ii) Which letters of the alphabet are represented in the following two 12-bit registers following shift operations to the left?

[4 marks]

0	1	1	0	0	1	1	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

1	0	0	1	0	0	1	1	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

- d (i) Using the following 8-bit register, explain what happens to the stored value if the register contents are shifted THREE places to the LEFT:

[2 marks]

0	0	0	0	1	1	1	1
---	---	---	---	---	---	---	---

- (ii) Using the following 8-bit register, explain what happens to the stored value if the register contents are shifted THREE places to the RIGHT:

[2 marks]

0	1	0	1	1	0	0	0
---	---	---	---	---	---	---	---

2

Communication and internet technologies

Key objectives

Objectives of this chapter are to revise:

- simplex, half-duplex and full-duplex
- serial and parallel data transmission
- asynchronous and synchronous data transmission
- universal serial bus (USB) connections
- parity check
- automatic repeat request (ARQ)
- checksum
- echo checking
- internet technologies

● Key definitions



Term	Definition
Simplex	Data transmission in one direction only
Half-duplex	Data transmission in two directions but not at the same time
Full-duplex	Data transmission in two directions at the same time
Serial transmission	Data is sent one bit at a time over a single wire or channel
Parallel transmission	Several bits of data are sent down several wires or channels at the same time (e.g. 1 byte of data sent down 8 wires/channels)
Asynchronous transmission	Data being transmitted in an agreed bit pattern
Synchronous transmission	A continuous stream of data which is accompanied by timing signals generated by an internal clock

● Data transmission



Simplex, half-duplex and full-duplex

Simplex – data transmission in one direction only.

Half-duplex – data transmission in two directions but not at the same time.

Full-duplex – data transmission in two directions at the same time.

Serial and parallel data transmission

Serial – data is sent one bit at a time over a single wire or channel.

Parallel – several bits of data are sent down several wires or channels at the same time (e.g. 1 byte of data sent down 8 wires/channels).

Asynchronous and synchronous data transmission

Asynchronous – this refers to data being transmitted in an agreed bit pattern; data is grouped together and sent with control bits known as the *start bit* and *stop bit*.

Synchronous – this is a continuous stream of data which is accompanied by timing signals generated by an internal clock; the timing signals ensure the sender and receiver are synchronised with each other to ensure the bits are reassembled into the correct bit patterns at the receiving end.

Universal Serial Bus (USB)

USB is an asynchronous serial data transmission method which has become an industry standard. When a device is plugged into a computer using one of the USB ports:

- the device is automatically detected
- a known device is automatically recognised and a device driver is loaded
- if a new device is plugged in, the computer looks for the correct device driver or the user will be prompted to download the driver.

Common errors

- Students often confuse half-duplex and full-duplex data transmission.
- Reference to cables rather than wires or channels is a very common error when referring to data transmission.

Sample question

1 a Name the direction and method of data transmission being described below:

- data sent in one direction only, one bit at a time, along a single wire
- bytes of data being sent in two directions at the same time using eight channels.

b Indicate which of the following are features of serial data transmission or parallel data transmission (put a tick (✓) in the appropriate column):

Feature	Serial	Parallel
faster data transmission of the two methods		
uses one wire or channel to transmit data		
works well over long distances		
over long distances, bit pattern can be skewed		
USB is an example of its use		
works best over short distances		

Student answer

- 1 a (i) simplex, serial
(ii) full-duplex, parallel

b

Feature	Serial	Parallel
faster data transmission of the two methods		✓
uses one wire or channel to transmit data	✓	
works well over long distances	✓	
over long distances, bit pattern can be skewed		✓
USB is an example of its use	✓	
works best over short distances		✓

Examiner's comments

a The above answers are correct but it is necessary to analyse the sentences carefully to pick out the two parts of the data transmission:

data sent in one direction only – this is simplex

one bit at a time along a single wire – this is serial

data being sent in two directions at the same time – this is full-duplex

bytes of data ... using eight channels – this is parallel

b Questions where boxes need to be ticked should be treated with caution to avoid silly errors. Read each statement carefully and decide which category it falls into – it is unlikely any pattern of correct answers will emerge. Always remember to place only one tick for each answer unless told to do otherwise – if two ticks are given in any of the above rows, no marks would be awarded for that row.

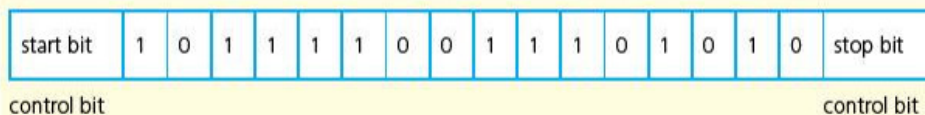


● Sample question

2 Describe the differences between *asynchronous* and *synchronous* data transmission.

Student answer

Asynchronous data transmission refers to data being transmitted in an agreed bit pattern. Data bits (1s and 0s) are grouped together and sent with control bits:



This means that the receiver of the data knows when the data starts and when it stops.

This prevents data becoming mixed up; without these control bits, it would be impossible to separate groups of data as they arrived.

Synchronous data transmission is a continuous stream of data (unlike asynchronous data which is sent in discrete groups). The data is accompanied by timing signals generated by an internal clock. This ensures that the sender and receiver are synchronised with each other.

The receiver counts how many bits (1s and 0s) were sent and then reassembles them into bytes of data. The timing must be very accurate here since no control bits are sent in this type of data transmission. However, it is a faster data transfer method than asynchronous and is therefore used where this is an important issue (for example, in network communications).

Examiner's comments

The answer given is for a top grade A student. To gain full marks doesn't really require this depth of answer. If the question carried 4 or 5 marks, then only 4 or 5 of the key points need to be mentioned:

Asynchronous:

- 1st: mention the use of control bits.
- 2nd: give a possible diagram (as shown above) or indicate where a start bit and stop bit would be situated.
- 3rd: give some description such as 'control bits prevent data being mixed up by "sandwiching" the required data'.

Synchronous:

- 1st: this is a continuous stream of data.
- 2nd: use of timing signals generated by an internal clock.
- 3rd: ensures sender and receiver are synchronised.
- 4th: receiver counts number of bits and reassembles them into bytes of data.

Your answer should contain 4 or 5 of the above points.

Examiner's tips

Always remember to analyse statements carefully to make sure you understand which key points are being asked in the question.

When writing essay-type questions, remember to pick out only the key points in your answer; it is often a good idea to briefly write your main points in pencil in the margin or on additional paper to ensure you miss nothing out.

● Error checking of data following data transmission

Parity checks

- Parity checks can be *even* (even number of 1-bits) or *odd* (odd number of 1-bits); one of the bits is known as the *parity bit*.
- Parity blocks allow an erroneous bit (following data transmission) to be identified.
- If the parity of a byte of data hasn't changed (but there is still an error following transmission), then it is impossible using this method to identify where the error has occurred.
- Alternative methods need to be employed.

Automatic Repeat Request (ARQ)

- ARQ uses an acknowledgement to indicate data received correctly.
- A timeout is used which is the time allowed to elapse before an acknowledgement is received.

Checksum

- Data is sent in blocks.
- An additional value, called the checksum, is sent at the end of the block of data.
- The checksum is based on the number of bytes in the block.
- If the checksum at the receiver's end doesn't match the checksum at the end of the data block, then an error in transmission has occurred.

Echo check

- Data is returned to the sender.
- The sender compares the data sent with the data received back.
- If different, then an error has occurred somewhere in the process.
- The problem could be that the data was actually sent correctly, but it was during the process of sending the data back to the sender that the error occurred.

● Common errors

- Students often confuse odd parity with odd numbers, e.g. 01100101 is an ODD number but actually has even parity.
- Students often confuse ARQ and echo check – carefully check the key differences above.

● Sample question

3 a What type of parity have the following binary numbers?

(i) 1 0 0 0 1 1 0 1

(ii) 1 1 1 0 0 0 1 1

b Explain why parity checking is done.

Student answer

- 3 a (i) even parity
 (ii) odd parity
 b To detect errors following data transmission or the copying of data from one medium to another.

Examiner's comments

In part a, it is important to indicate that it is the number of 1-bits that determine the parity of the byte of data.

In part b, the key part of the answer is that the check is done after the data transmission or after the copying of data operation.

● Sample question

- 4 Apart from using parity checks, describe three other methods for checking errors following data transmission.

Student answer

Automatic Repeat Request (ARQ) is another method used to check whether data has been correctly transmitted.

It uses an acknowledgement (a message sent by the receiver indicating that data has been received correctly) and timeout (this is the time allowed to elapse before an acknowledgement is received).

If an acknowledgement isn't sent back to the sender before timeout occurs, then the message is automatically re-sent.

Checksum is another way to check if data has been changed or corrupted following data transmission. Data is sent in blocks and an additional value, the checksum, is also sent at the end of the block of data.

To explain how this works, it is assumed the checksum of a block of data is 1 byte in length. This gives a maximum value of $2^8 - 1$ (i.e. 255). The value 0000 0000 is ignored in this calculation.

When a block of data is about to be transmitted, the checksum for the bytes is first of all calculated. This value is then transmitted with the block of data. At the receiving end, the checksum is re-calculated from the block of data received. This calculated value is then compared with the checksum transmitted. If they are the same value, then the data was transmitted without any errors; if the values are different, then a request is sent for the data to be re-transmitted.

With echo check, when data is sent to another device, this data is sent back again to the sender. The sender compares the two sets of data to check if any errors occurred during the transmission process.

This isn't a very reliable method. If the two sets of data are different, it isn't known whether the error occurred when sending the data in the first place, or if the error occurred when sending the data back for checking!

However, if no errors occurred then it is another way to check that the data was transmitted correctly.

Examiner's comments

The answer given is for a top grade A student. To gain full marks doesn't really require this depth of answer. If the question carried 9 marks, then only 9 key points need to be mentioned:

ARQ:

- 1st mark point: uses an acknowledgement when message received.
- 2nd mark point: use of the timeout function; the time allowed to elapse before an acknowledgement is received.
- 3rd mark point: if acknowledgement not received in the allowed time, message is automatically re-sent.

Checksum:

- 1st mark point: data is sent in blocks.
- 2nd mark point: checksum is sent at end of data block.
- 3rd mark point: checksum re-calculated at receiving end and if it doesn't match checksum sent, then an error has occurred.

Echo check:

- 1st mark point: message/data sent back to sender.
- 2nd mark point: sender compares original message/data with that sent to them from receiver.
- 3rd mark point: if the two messages/data not the same, error message is generated and the message/data is re-sent.

Examiner's tips

Always remember to analyse statements carefully to make sure you understand which key points are being asked in the question.

When writing essay-type questions, remember to only pick out the key points in your answer; it is often a good idea to briefly write your main points in pencil in the margin or on additional paper to ensure you miss nothing out.

● Internet technologies



Internet Service Provider (ISP)

An ISP is a company that provides the user with access to the internet; they usually charge a monthly fee. The ISP gives the user an account and email address.

Internet Protocol (IP) address

Each device is given an IP address as soon as it connects to the internet. The IP address will be different each time the device connects.

Media Access Control (MAC) address

A MAC address uniquely identifies a device connected to the internet. The MAC address doesn't usually change (but refer to UAA and LAA).

HTML structure and presentation

Structure: this is an essential part of HTML documents; it includes semantics and structural mark-up of the documents.

Presentation: this is the style of the document – how the document will appear on a user's computer screen.

Hypertext Transfer Protocol (http)

A set of rules that must be obeyed when transferring files across the internet.

Web browser

Software which allows a user to display a web page on the computer screen; this software interprets/translates the HTML from the website and shows the result.

● Common errors □

- IP address and ISP are often confused by students – make sure you know the difference.
- Students often confuse MAC address with IP address.

● Sample question □

5 a What is meant by a URL?

b Consider: `http://www.hachette.co.uk/igcse_books`

- (i) Which part of this address is the protocol?
- (ii) Which part of this address is the web server name?
- (iii) Which part of this address is the file name?

Student answer

5 a Uniform resource locator
 b (i) `http://`
 (ii) `www.hachette.co.uk`
 (iii) `/igcse_books`

Examiner's comments

In part a, there is no need to give a description since the meaning of the term URL is all that is required.

Part b is fairly straightforward; but students need to be careful not to include wrong parts of the address in each of the three answers, e.g. in part (ii), `http://www.hachette.co.uk` would be incorrect.

● Sample question □

6 Explain the difference between an IP address and a MAC address.

Student answer

An IP address will change each time a device connects to the internet; the address is unique.

A MAC address uniquely identifies a device connected to the internet.

Examiner's comments

It is necessary to be really careful here to distinguish between these two internet terms.

It is important to stress that the IP address changes each time a device connects to the internet. It may be worth mentioning that it is the NIC in the device that is given the MAC address in the format: NN-NN-NN-DD-DD-DD.

Examiner's tips

It is important to learn these definitions and be aware of the fact that it is easy to confuse many of the terms since they either have a similar name or their functions appear to be very similar.

Exam-style questions

- 1 a Describe **three** things that will happen when a printer is plugged into a computer via one of the USB ports. [3 marks]
- b What type of data transmission is used when connecting a device via a USB port? [2 marks]
- c Explain the difference between asynchronous and synchronous data transmission. [3 marks]
- 2 Eleven bytes of data are transmitted to another computer. Both sender and receiver use even parity checks. A twelfth byte, known as the parity byte, is also sent at the end of this string of data.

The following table shows the received bytes:

	parity bit	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8
byte 1	1	1	0	1	1	1	1	0
byte 2	0	1	1	0	0	0	0	0
byte 3	0	0	1	0	0	0	1	0
byte 4	0	1	1	0	0	1	0	1
byte 5	1	1	1	1	0	0	0	0
byte 6	1	0	1	1	1	0	1	1
byte 7	0	0	0	0	1	1	1	1
byte 8	0	0	0	0	1	1	0	0
byte 9	1	0	0	1	0	0	1	1
byte 10	1	1	0	1	0	0	1	1
byte 11	0	1	1	0	0	0	0	0
parity byte	1	0	0	0	0	0	0	1

- a One of the bits has been incorrectly transmitted. Indicate using the bit number and byte number which bit has been incorrectly transmitted. [2 marks]
- b Explain how you located the incorrect bit value. [3 marks]
- c Write down the corrected byte value. [1 mark]
- d Describe **two** examples of errors, following data transmission, that parity checks would not correctly identify. [4 marks]
- e Name **three** other methods that could be used to identify errors following data transmission. [3 marks]

3 Which **four** internet terms are being described here? [4 marks]

Description	Term
set of rules that must be obeyed when transferring files or data across the internet	
id given to a device when it connects to the internet; the id changes each time the device connects	
company that provides the user with access to the internet for a monthly subscription	
id which uniquely identifies a device connected to the internet; this id rarely changes	

4 A student made the following five statements. Explain why each is incorrect.

- a 'When data is transmitted in both directions but not at the same time, using one wire, it is known as parallel simplex data transmission.'
- b 'Asynchronous data transmission is a continuous stream of data which uses timing signals to ensure data is received correctly.'
- c 'When a device is plugged into a computer's USB port, the user is always prompted to load the appropriate device driver.'
- d 'ARQ returns data to the sender; the sender compares the sent data and the received data to see if they match.'
- e 'A binary number that ends in a 1, e.g. 11001011, is always odd parity.'

[10 marks]

3 Logic gates and logic circuits

Key objectives

Objectives of this chapter are to revise:

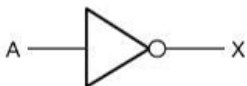
- NOT, AND, OR, NAND, NOR, XOR logic gates
- truth tables
- logic circuits
- logic statements (Boolean algebra)
- simplification of logic circuits

Key definitions

Term	Definition
Logic gates	Logic gates take in binary inputs and produce a binary output
Logic circuits	Logic circuits are made up of several logic gates and are designed to carry out a specific function
Truth tables	Truth tables check the output of a logic circuit

Logic gates

NOT gate



Description:

The output, X, is 1 if:
the input, A, is 0

Truth table:

Input	Output
A	X
0	1
1	0

How to write this:

$X = \text{NOT } A$ (logic notation)
 $X = \bar{a}$ (Boolean algebra)

AND gate



Description:

The output, X, is 1 if:
both inputs, A and B, are 1

Truth table:

Inputs		Output
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

How to write this:

$X = A \text{ AND } B$ (logic notation)
 $X = a \cdot b$ (Boolean algebra)

OR gate



Description:

The output, X, is 1 if:
either input, A or B, is 1

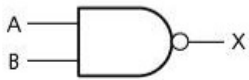
Truth table:

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

How to write this:

$X = A \text{ OR } B$ (logic notation)
 $X = a + b$ (Boolean algebra)

NAND gate



Description:
The output, X, is 1 if:
input A AND input B are
NOT both 1

Truth table:

Inputs		Output
A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

How to write this:
 $X = \overline{A \text{ NAND } B}$ (logic notation)
 $X = \overline{a \cdot b}$ (Boolean algebra)

NOR gate



Description:
The output, X, is 1 if:
neither input A nor input
B is 1

Truth table:

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

How to write this:
 $X = \overline{A \text{ NOR } B}$ (logic notation)
 $X = \overline{a + b}$ (Boolean algebra)

XOR gate



Description:
The output, X, is 1 if:
(input A is 1 AND input
B is 0) OR
(input A is 0 AND input
B is 1)

Truth table:

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

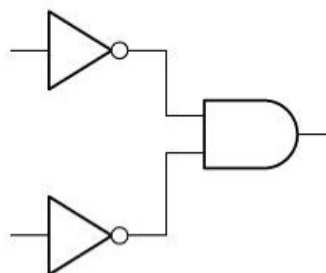
How to write this:
 $X = A \text{ XOR } B$ (logic notation)
 $X = (a \cdot \overline{b}) + (\overline{a} \cdot b)$ (Boolean algebra)
 NOTE: this is sometimes written as:
 $(a + b) \cdot \overline{(a \cdot b)}$

Common errors

- Many marks are lost by not being very clear with the symbol for each of the 6 logic gates; it is often worth writing the gate's name inside the symbol if art is not one of your strong points.
- Do not confuse the following:



with:



They do not produce the same result!

It is illegal to photocopy this page



● Sample question



1 Which logic gates are being described here:

a $a + b$

b $\overline{a \cdot b}$

c $(a \cdot \overline{b}) + (\overline{a} \cdot b)$

Student answer

- 1 a OR gate
b NAND gate
c XOR gate

Examiner's comments

You need to remember that the ' \cdot ' symbol represents AND, the '+' symbol represents OR and ' \overline{a} ' represents NOT. The only compound logic gate statement is for the XOR gate, which makes it easy to spot.

● Sample question



2 Show why $(a \cdot \overline{b}) + (\overline{a} \cdot b)$ and $(a + b) \cdot (\overline{a \cdot b})$ both represent an XOR gate.

Student answer

Use of a truth table:

Inputs							
a	b	$(a \cdot \overline{b})$	$(\overline{a} \cdot b)$	$(a \cdot \overline{b}) + (\overline{a} \cdot b)$	$(a + b)$	$\overline{(a \cdot b)}$	$(a + b) \cdot \overline{(a \cdot b)}$
0	0	0	0	0	0	1	0
0	1	0	1	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	0	1	0	0

Columns 5 and 8 are the same which proves that the two expressions are the same and the output values correspond with an XOR gate.

Examiner's comments

When trying to prove that two logic statements or logic circuits represent the same system, it's always best to use a truth table and compare the two outputs. If they are the same, then this proves both expressions are the same.

It is advisable to split up the expressions into smaller parts (as shown above) to avoid errors and make the task considerably easier.

Examiner's tips

Using Boolean algebra is much easier than trying to write the logic statement in another format such as: (NOT A AND B) OR (B OR C).

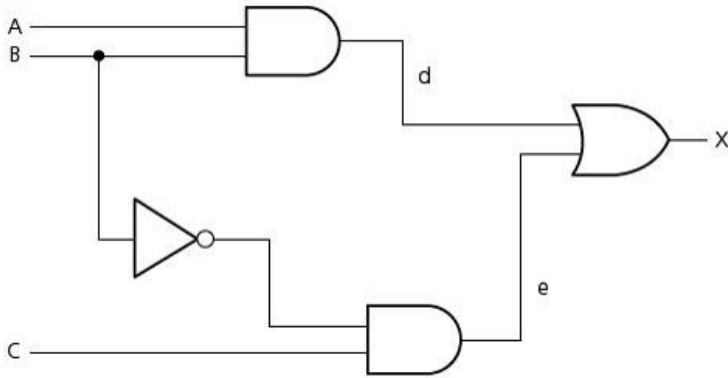
When trying to prove two logic expressions or logic circuits carry out the same task, make use of truth tables to make the task much easier; the truth table can also help if you are asked to simplify an expression or circuit.



● Logic circuits

Truth tables

Producing the truth table for:

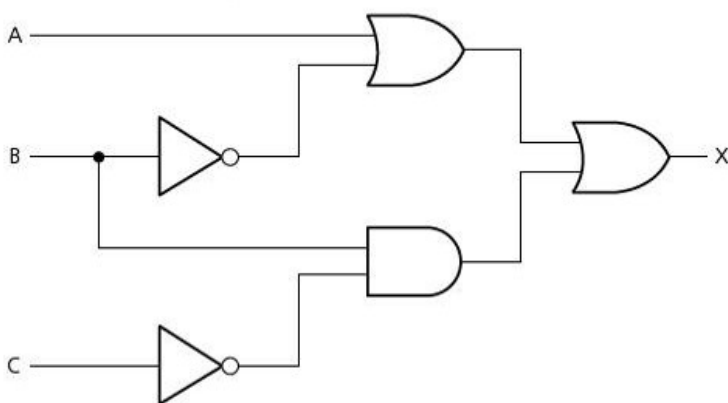


Inputs			Working		Output
A	B	C	d	e	X
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	0	1

(Values for d and e are intermediate values to help simplify the solution.)

Logic statements

Producing the logic statement for:



$X = (A \text{ OR NOT } B) \text{ OR } (B \text{ AND NOT } C)$
 or using Boolean algebra: $(a + \bar{b}) + (b \cdot \bar{c})$

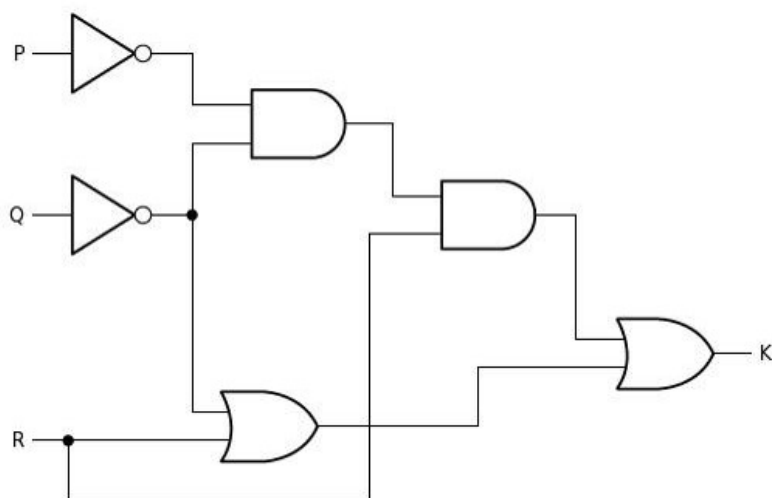
It is illegal to photocopy this page

Logic circuits

Producing the logic circuit for:

$K = ((\text{NOT } P \text{ AND NOT } Q) \text{ AND } R) \text{ OR } (\text{NOT } Q \text{ OR } R)$

Boolean expression: $((\bar{p} \cdot \bar{q}) \cdot r) + (\bar{q} + r)$



Common errors

- Many marks are lost by not being very clear with the symbol for each of the 6 logic gates; it is often worth writing the gate's name inside the symbol if art is not one of your strong points.
- Rather than writing NOT A, students often lose marks by writing $A = 0$; logic expressions are based on the value 1 and NOT A must be used rather than $A = 0$.

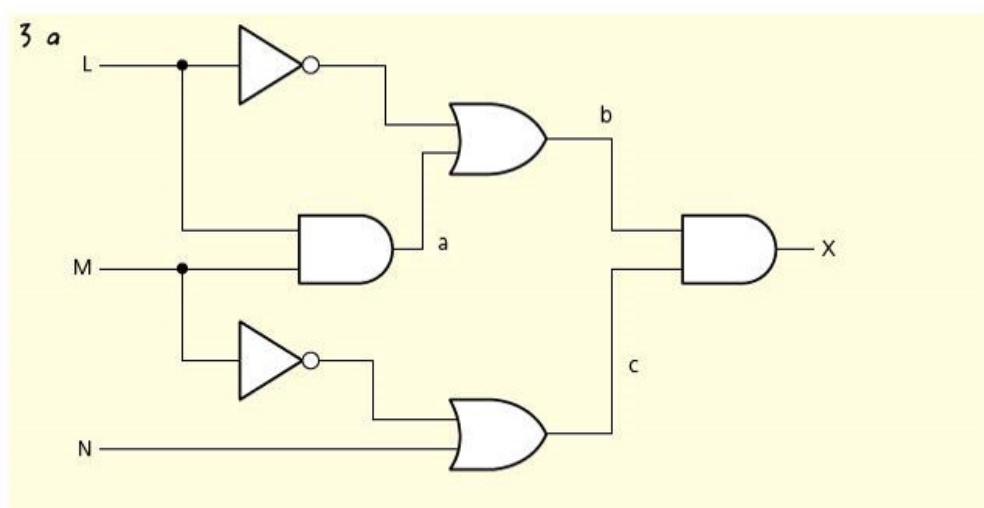
Sample question

3 a Draw the logic circuit for the following logic statement:

$$(L \cdot M + \bar{L}) \cdot (\bar{M} + N)$$

b Complete the truth table for the above logic statement.

Student answer



b

Inputs			Working			Output
L	M	N	a	b	c	X
0	0	0	0	1	1	1
0	0	1	0	1	1	1
0	1	0	0	1	0	0
0	1	1	0	1	1	1
1	0	0	0	0	1	0
1	0	1	0	0	1	0
1	1	0	1	1	0	0
1	1	1	1	1	1	1

Examiner's comments

The student has correctly labelled the intermediate stages as a, b and c and these values have been shown in the three 'working columns' in the truth table.

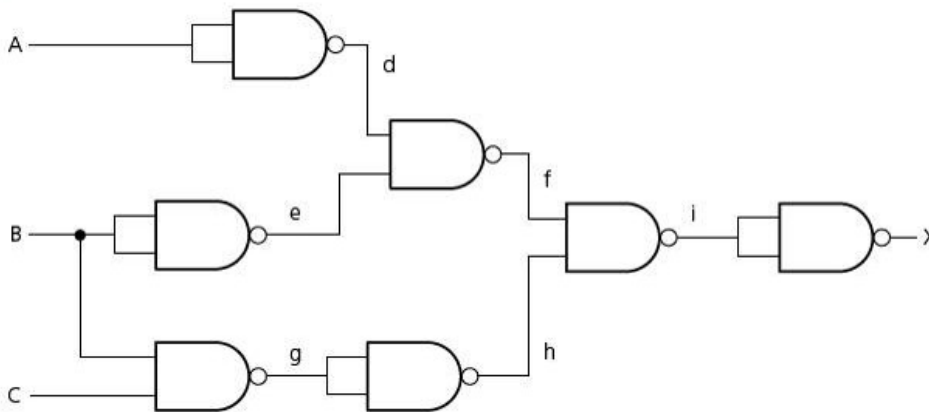
When producing the logic circuit, it is much easier to split it into four stages:

- $L \cdot M$ (join L and M with an AND gate)
- \bar{L} then joins on to the above using an OR gate
- $\bar{M} + N/M$ (join M and N with an OR gate)
- expressions 1 and 2 then link into the third expression with an AND gate.

Sample question



4 a Complete the truth table for the following logic circuit.



Inputs			Working			Output
A	B	C				X
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

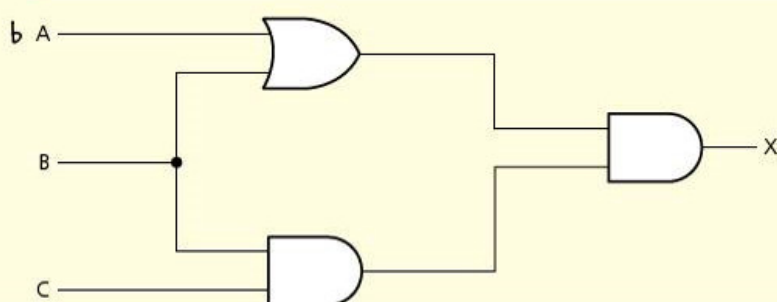
b Redraw the above logic circuit using AND and OR gates only.

It is illegal to photocopy this page

Student answer

a

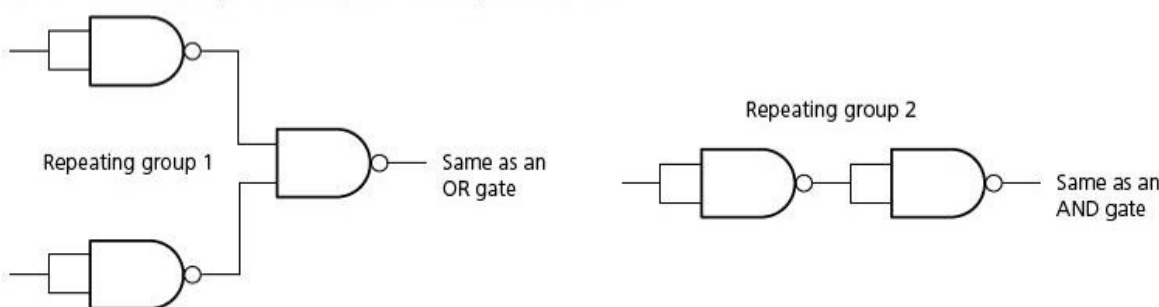
Inputs			Working						Output
A	B	C	d	e	f	g	h	i	X
0	0	0	1	1	0	1	0	1	0
0	0	1	1	1	0	1	0	1	0
0	1	0	1	0	1	1	0	1	0
0	1	1	1	0	1	0	1	0	1
1	0	0	0	1	1	1	0	1	0
1	0	1	0	1	1	1	0	1	0
1	1	0	0	0	1	1	0	1	0
1	1	1	0	0	1	0	1	0	1



Examiner's comments

In part a, the student has correctly labelled the intermediate stages as d, e, f, g, h and i. These values have been shown in the six 'working columns' in the truth table. Whilst this looks like a lot of extra work, it does reduce the possibility of errors, but it also makes it easier to backtrack in case of any mistakes made.

In part b, the student has correctly recognised which group of NAND gates make up an AND gate and which group of NAND gates make up an OR gate:



If the above NAND 'building blocks' had not been recognised, then applying a truth table to each repeating group could be used to find the appropriate single logic gate:

Inputs		Repeating group 1 output	Repeating group 2 output
A	B		
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

This shows repeating group 1 to be an OR gate and repeating group 2 to be an AND gate, which then allows you to redraw the logic circuit with AND and OR gates only.

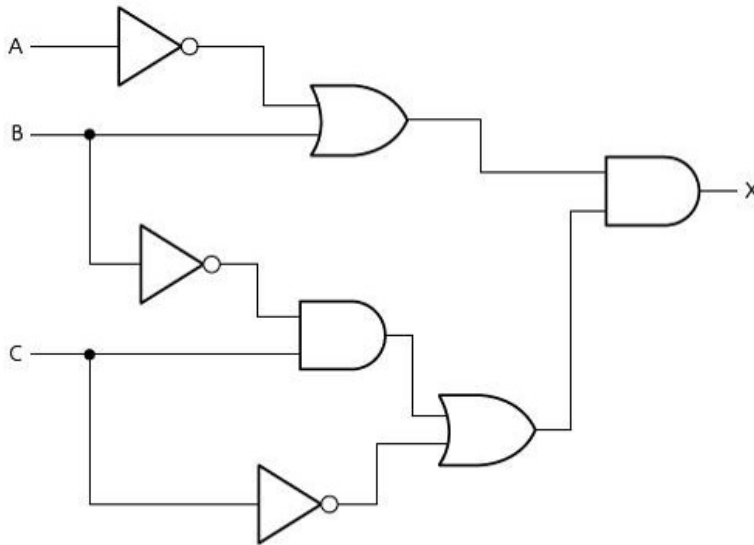
Examiner's tips

When completing truth tables from logic circuits always break up the logic circuit into smaller, more manageable steps.

It is worth remembering the 'building blocks' used by logic circuit designers since this can save a lot of time when answering questions such as that shown in part b.

Exam-style questions

- 1 a Write the logic statement which represents the following logic circuit. [3 marks]

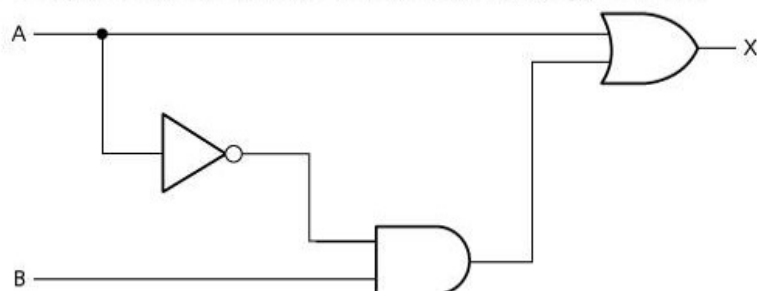


- b Complete the truth table for the above system. [4 marks]

Inputs			Working	Output
A	B	C		X
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- c By checking the truth table, or otherwise, show how the bottom half of the logic circuit in part a could be reduced to two gates only. Explain all your reasoning. [3 marks]

- 2 a Complete the truth table for the following logic circuit. [2 marks]



Inputs		Working	Output
A	B		X
0	0		
0	1		
1	0		
1	1		

- b Which single logic gate can replace the entire logic circuit shown in part a? [1 mark]
- c (i) Which logic gate is often used as the building block for more complex logic circuits? [1 mark]
- (ii) Why do companies use 'off-the-shelf' logic units to build more complex logic circuits? [2 marks]
- 3 a Draw the logic circuit which is represented by the following logic statement. $(\bar{p} \cdot \bar{q}) \cdot p + (q + r)$ [6 marks]

- b (i) Draw the logic circuit for:

$$X = (A \text{ AND } B) \text{ OR } ((B \text{ OR } C) \text{ AND } (B \text{ AND } C))$$
 [5 marks]

- (ii) Complete the truth table for this logic circuit. [4 marks]

Inputs			Working	Output
A	B	C		X
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

- (iii) Look at the logic circuit you drew in part b(i) very carefully. Also look at the truth table you completed in part b(ii).

Now produce a simplified logic circuit which does the same task as the one you drew in part b(i). [3 marks]

- (iv) Prove that the logic circuits drawn in parts b(i) and b(iii) are the same. [2 marks]

- 4 A chemical process is monitored by a logic circuit. There are three inputs to the logic circuit which represent key parameters in the chemical process. An alarm, X, will give a value of 1 depending on certain conditions.

The following table summarises these conditions.

Parameter description	Parameter	Binary value	Description of condition
temperature	T	0	temperature \geq 140°C
		1	temperature < 140°C
mixer speed	M	0	mixer speed \geq 500 rpm
		1	mixer speed < 500 rpm
acidity	A	0	acidity \geq 6 moles
		1	acidity < 6 moles

An alarm, X, will generate the value 1 if:

either: temperature \geq 140°C and mixer speed < 500 rpm

or: temperature < 140°C and acidity \geq 6 moles

or: mixer speed \geq 500 rpm and acidity < 6 moles

- Write the logic statement to represent the alarm, X, above. [3 marks]
- Draw the logic circuit to represent the above monitoring system. [7 marks]
- Complete the truth table for the above system. [4 marks]

Inputs			Working	Output X
T	M	A		
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

4

Operating systems and computer architecture

Key objectives

Objectives of this chapter are to revise:

- operating systems
- interrupts
- buffers
- computer architecture (Von Neumann concept)
- addressing
- registers
- buses
- the fetch–execute cycle

● Key definitions

Term	Definition
Operating system	The software running in the background of a computer system. It manages many of the basic functions
Interrupt	An interrupt is a signal sent from a device (or from some software) to the processor requesting its attention
Buffer	A buffer is a temporary memory area in a device
Computer architecture	How a computer system is designed
Buses	Buses move data around the computer and also send out control signals to synchronise the internal operations
Register	High-speed storage areas within the computer
Memory unit	Made up of addresses and contents
Control unit	Controls the operation of the memory, processor and input/output devices

● Operating systems, interrupts and buffers

Operating systems

The basic operations/tasks carried out by an operating system include:

- human–computer interface (HCI)
- multi-tasking
- multiprogramming
- batch processing
- error handling/reporting
- load/run applications
- management of user accounts
- file utilities (such as save, copy, delete)
- processor management
- memory management
- real-time processing
- interrupt handling
- security (e.g. log on, passwords, username, etc.)
- input/output control.

Some household devices, such as ovens or washing machines, do not require an operating system. This is because they carry out simple, unchanging tasks which are initiated by the user pressing a button or selecting from a touch screen.

Interrupts and buffers

An interrupt is a signal sent from a device (or from some software) to the processor requesting its attention. The interrupt will cause the processor to temporarily stop what it is doing to service the interrupt. Examples include: paper jam in a printer, pressing <CTRL><ALT><BREAK> keys on a keyboard or software trying to divide by zero.

A buffer is a temporary memory area in a device. It is often used to compensate for the slower operating speed of peripherals when compared to a processor. Buffers allow a processor to carry on with other tasks whilst, for example, a print job is being carried out in the background. They are also used when streaming music or videos from the internet to ensure that playback doesn't keep pausing waiting for the data transfer/download to take place.

● Common errors

- Students often claim that a message or data (rather than a signal) is sent to a processor as a definition of an interrupt; such answers always lose marks.
- Many students think that it is only hardware that can send interrupts; in fact, software interrupts are far more common (these can either be faults in the software itself or errors found during running, e.g. a missing .exe file).



● Sample question

- 1 a Name two devices that do not need an operating system.
- b Explain why an operating system is not required.



Student answer

- 1 a *Washing machine and a fridge/freezer*
 b *They carry out single tasks that do not vary.*
Input is a press of a button or an option from a touch screen.
This activates a simple hardware function which doesn't need an OS to control it.

Examiner's comments

In part a, many household devices with a microprocessor could be chosen, such as microwave oven, dishwasher, toaster, and so on.

In part b, it is probably only necessary to make two valid points to gain the full marks. It would be acceptable to explain why an operating system is used as an alternative way of answering the question.

● Sample question

- 2 Explain the role of buffers and interrupts when a document is sent to a printer to be printed out.



Student answer

- An interrupt is a signal sent, in this case, from the printer to the processor.
- The interrupt could be a request for more data or an error report (e.g. out of paper or out of ink).
- Once serviced, the processor can carry on with other tasks that were temporarily suspended.
- The (printer) buffer is a temporary memory area.
- Data is sent to the (printer) buffer waiting to be printed / waiting to be sent to the printer.
- Once the buffer is emptied of data, an interrupt is sent requesting more data to be sent.
- Buffers allow the processor to carry on with other tasks whilst printing takes place in the background ...
- ... this means the processor is not held up by the relatively slow printing process.

Examiner's comments

Questions of this type probably carry a maximum of 3 or 4 marks. This means you only need to cover the key marking points:

- role of an interrupt: used to request more data or convey an error
- role of a buffer: used to temporarily hold printing data
- how an interrupt is used: signal sent to processor requesting attention
- how a buffer is used: stores data until needed; allows microprocessor to do other tasks, etc.

Examiner's tips

When writing essay-type questions, remember to pick out the main points; it is often a good idea to note down key points (in the margin or on one of the blank sheets) to ensure you don't miss any of these points in your final answer.

● Computer architecture



Von Neumann architecture

This concept holds programs and data in memory. Data moves between the memory unit and the processor. A number of different diagrams exist to show the links and flow of data between the processor (control unit), memory unit and input/output devices.

Buses

Buses move data around the computer and also send out control signals to synchronise the internal operations. There are three different buses used:

- **Address bus:** this carries signals relating to addresses between the processor and memory; it is uni-directional.
- **Data bus:** this sends data between the processor, memory unit and the input/output devices; it is bi-directional.
- **Control bus:** this carries signals relating to control and coordination of all activities within the computer; it can be uni-directional or bi-directional due to internal connections.

Registers

These are high-speed storage areas within the computer; all data must be represented in a register before it can be processed. There are five different registers in this type of architecture:

- Memory Address Register (MAR)
- Memory Data Register (MDR)
- Arithmetic and Logic Unit (ALU)
- Program Counter (PC)
- Current Instruction Register (CIR).

Memory unit

This is made up of addresses and contents. Each address will uniquely identify every location within the memory map.

Control unit

This controls the operation of the memory, processor and input/output devices; the control unit reads instructions, interprets them and sends out signals along the control bus to synchronise all computer components.

● Common errors

- Students often confuse the role of the MAR and MDR in, for example, the READ and WRITE operation from/to the memory unit.
- Many students suggest that the ALU can ‘make decisions’; this is not the case – the ALU allows arithmetic and logical operations to be carried out and makes no decisions on what happens following these operations.



● Sample question

- 3 a The location in a memory unit contains the address 1 1 0 1 1 0 0 0 and the contents 0 0 0 1 1 0 0 0.

Describe a READ operation on location 1 1 0 1 1 0 0 0 indicating the role of the MAR, MDR and buses.

- b If the value 1 1 1 0 1 1 1 0 is to be stored in location 1 1 1 1 0 1 1 1, describe the WRITE operation indicating the role of the MAR, MDR and buses.



Student answer

- 3 a MAR will contain 11011000
MDR will contain 00011000
A read signal is sent to the computer memory along the control bus.
- b MAR will contain 11110111
MDR will contain 11101110
A write signal is sent to the computer memory along the control bus.

Examiner's comments

Whilst the student has answered the question correctly in both parts, it is often a good idea to show a memory map to aid in the description.

Address	Contents
1101 1000	0001 1000
↓	↓
1111 0111	1110 1110

This will help greatly in the description of the READ and WRITE operation. The location contains the address and contents; it is important to distinguish between location and address.

● Sample question

4 Describe the role of buses and addresses in the von Neumann architecture.

Student answer

Buses move data (the data bus) around the computer and also send signals (address bus and control bus) to synchronise all operations. Buses can be uni-directional or bi-directional.

An address is part of a memory location that uniquely describes where data can be found in the memory map.

Examiner's comments

Whilst the inclusion of a diagram to help explain the role of buses would be useful, it wouldn't really help in a question asking for a straightforward description of the two terms. It is very important not to confuse registers with addresses.

Examiner's tips

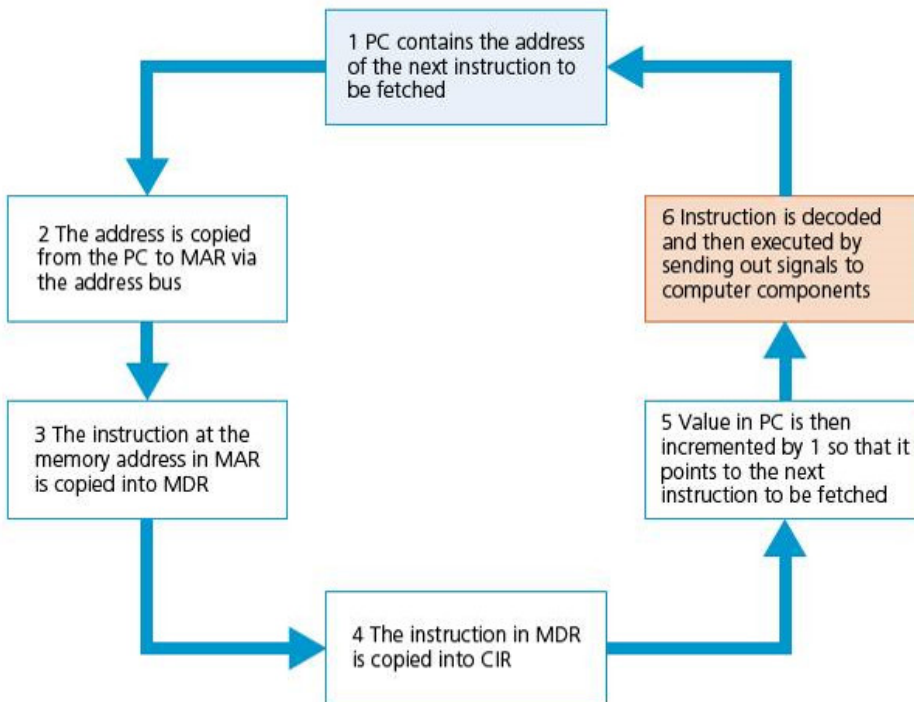
A basic understanding of how to draw a diagram linking together all the components in the von Neumann computer architecture is very useful in questions of this type. The two diagrams shown in the Student's Book are fairly simple to understand, but other diagrams exist which prove to be more useful to some students.

● Fetch–execute cycle

Fetch: the next instruction is *fetched* from the memory address currently stored in the Program Counter (PC) and is then stored in the Current Instruction Register (CIR); the PC is then incremented so that the next instruction can be processed.

Execute: the decoded instruction is then passed as a set of control signals to the appropriate components of the computer system.

In diagrammatic form:



● Sample question

- 5 Describe three of the registers used in the fetch–execute cycle including their function.

Student answer

Program counter (PC) – this is used to keep track of the address of the next instruction to be fetched.

Memory address register (MAR) – this stores the address of the instruction to be processed.

Memory data register – this stores the instruction to be processed.

Current instruction register – this temporarily stores the instruction currently being processed.

Examiner’s comments

All that is required is a brief description of the function of each register. Just naming the registers would not be enough for more than half the marks in a question of this type. Only three registers were needed.

● Sample question

- 6 Describe the function of:
- the FETCH operation
 - the EXECUTE operation
- in the fetch–execute cycle.

Student answer

- 6 a Next instruction fetched ...
 from memory address currently stored in PC
 then stored in the CIR
 PC is incremented so next instruction can be fetched.
- b Decoded instruction is passed ...
 on to appropriate components of the computer system
 in the form of control signals.

Examiner's comments

This type of question can be answered in two ways: either by giving a brief description (as shown in the Student answer above), or by giving the 6 stages in the fetch-execute cycle. Whilst the second option is more comprehensive, it isn't really necessary when answering questions of this type.

Examiner's tips

It is a good idea to learn the rudimentary diagram showing the von Neumann computer and also to try to apply the fetch-execute cycle to your diagram; this will greatly help in questions involving computer architecture and how instructions are processed.

Exam-style questions

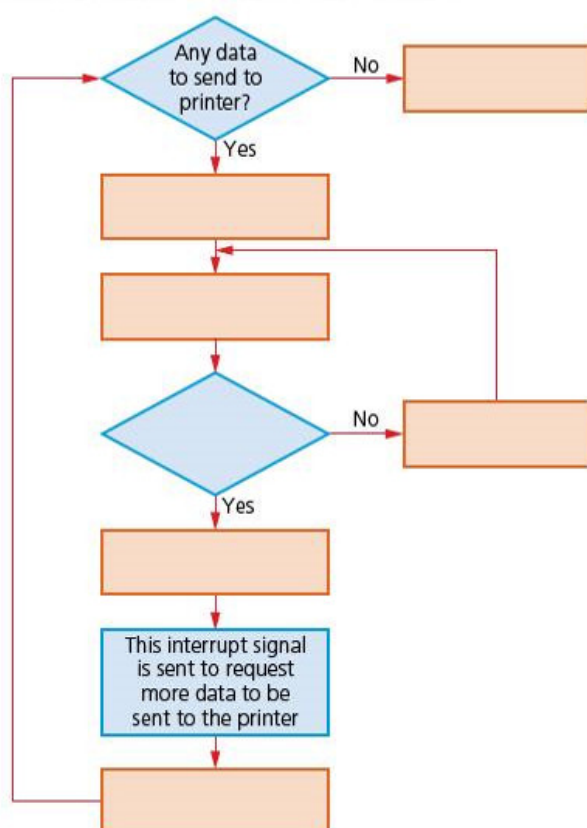
- 1 a What is meant by the two terms:

(i) buffer

(ii) interrupt?

[4 marks]

- b Look at the flowchart which shows what happens when a document is sent to a printer and is printed out.



Seven statements are missing from the flowchart.

By writing the statement numbers ONLY, complete the flowchart, using the statements from the following list. [7 marks]

Statement number	Statement
1	contents of the buffer are emptied to the printer and data from the document printed
2	current processor task is suspended whilst the interrupt is being serviced
3	data from the document to be printed is sent to the buffer from the computer memory
4	has all the data been sent to the printer?
5	processor continues with its tasks
6	processor is able to carry out other tasks while the buffer is being emptied
7	when all the data is printed, the buffer becomes empty; an interrupt signal is then sent to the processor

2 a A student made the following statements. Explain the statements.

(i) 'Windows is an example of a single-user multi-tasking operating system.'

(ii) 'Many operating systems support error handling.' [4 marks]

b Name **three** other features of an operating system. [3 marks]

3 a Name and explain the function of the three main buses used in a typical computer system. [6 marks]

b Look at this memory map.

(i) The contents at address 1 0 0 0 0 1 1 are to be read. What values would be stored in the MAR and MDR?

(ii) If the value 1 1 1 0 0 1 1 1 is to be written at memory location 1 1 1 1 1 1 0, what values would be stored in the MAR and MDR?

(iii) Complete the memory map showing the value from part (ii) after the WRITE function has been done. [5 marks]

Address	Contents
1000 0000	1000 1000
1000 0001	0111 0110
1000 0010	1100 1111
1000 0011	0101 0110
1000 0100	1111 1110
↓	↓
1111 1100	
1111 1101	
1111 1110	
1111 1111	

4 a (i) Describe the function of the ALU.

(ii) Describe the function of the Control Unit. [4 marks]

b The following table shows the stages in the fetch–execute cycle. They are not written in the correct order. Place the numbers 1 to 6 in the right-hand column to show each stage in its correct order.

The first one has been done for you. [5 marks]

Stage	Order
address is copied from PC to MAR; this is done using the address bus	
contents at memory location contained in MAR are copied temporarily into MDR	
contents of MDR are copied and placed in CIR	
instruction is finally decoded and then executed by sending out signals (via the control bus) to various components of the computer system	
program counter (PC) contains address of memory location of next instruction to be fetched	1
value in PC is incremented by 1 so that it now points to the next instruction to be fetched	

5 Input and output devices

Key objectives

Objectives of this chapter are to revise:

- the operation and applications of the following input devices:
 - scanners (including barcode readers)
 - QR code readers
 - digital cameras
 - keyboards/keypads
 - pointing devices (e.g. mouse)
 - microphones
 - touchscreens
 - sensors
 - interactive whiteboards
- the operation and applications of the following output devices:
 - printers (inkjet, laser and dot matrix)
 - 3D printers
 - 2D/3D cutters
 - actuators
 - loud speakers
 - LCD/LED monitors
 - (multimedia) projectors

● Key definitions

Term	Definition
Input device	Any device that allows a computer to receive data from the outside world
Output device	Any piece of computer hardware equipment used to communicate the results of data processing carried out by a computer

● Input devices

An input device is hardware that allows a computer to receive data from the outside world.

2D scanners

These convert hard-copy documents into an electronic form which can be stored in a computer memory. Scanners operate by using a scan head which moves across the document producing an image which is sent to a lens via a series of mirrors. The focused image falls onto a charge-coupled device (CCD) which consists of a number of integrated circuits (made up of thousands of light-sensitive elements that turn light into an electric charge). Consequently, the original document is turned into a format which can be stored in a computer's memory. One application of 2D scanners is at an airport where they can read passport pages and allow, for example, a photographic image of a person's face to be compared to an image just taken by a digital camera. This involves the use of face-recognition technology to identify passengers.

3D scanners

3D scanners scan solid objects and produce an electronic 3D image. They make use of, for example, lasers, magnetic resonance, white light or X-rays. CT (computed tomography) scanners are used to produce a 3D image of a solid object. Tomography technology splits up the object into a number of very thin 'slices'. They tend to use X-rays, radio waves or gamma imaging methods. The resultant image allows a solid object to be stored as a series of digital values representing each 'slice'.

- X-rays: CT (computed tomography) scanners.
- Radio waves: MRI (magnetic resonance imaging) scanners.
- Gamma rays: SPECT (single-photon emission computed tomography) scanners.

Barcode readers

Barcodes consist of a series of dark and light lines of varying thickness.

Each digit or character is represented by a number of lines. Whilst barcodes which use digits only are the most common in supermarkets, many can also display other characters.

Each dark and light line can be represented by the binary values of 1 (dark line) or 0 (light line); e.g. L D L D D D D, which represents the light and dark lines for the digit '6' in one system, can be represented as 0 1 0 1 1 1 1 in the computer.

Barcodes are used in supermarkets to find prices of goods and to enable automatic stock control. They are also used extensively in library systems to track the whereabouts of books. The advantages to managers include:

- easier and faster to alter prices
- give instantaneous and comprehensive sales trends
- no need to price each item (saves time and money)
- allows automatic stock control
- can check customer's buying habits more easily.

Advantages to customers include:

- faster checkouts
- less chance of errors
- get an itemised bill
- cost savings can be passed on to customer
- better tracking of 'sell by dates'.

QR (quick response) codes

QR codes are a type of barcode. However, they can hold considerably more data.

QR codes are usually read by built-in cameras in smart phones or tablets using an installed app. Once the QR code has been scanned, it will send information back to the phone or tablet, e.g. a website link or general information (e.g. a phone number or 'special offer').



Digital cameras

Digital cameras contain a microprocessor which automatically:

- adjusts shutter speed
- focuses
- operates the flash
- adjusts the aperture
- removes 'red eye'
- reduces hand shake, and so on.

Images are captured when light passes through the lens onto a light-sensitive cell, which is made up of thousands of tiny elements called pixels. The number of pixels determines the size of the file needed to store the image.

Keyboards/keypads

Keyboards are the most common input device; keys are pressed by the operator to enter data directly into the computer. When a key is pressed, it completes a circuit and a signal is sent to the microprocessor which interprets which key has been pressed. Since entering data by keyboard is a relatively slow process, most computer systems use a keyboard buffer which prevents the microprocessor waiting for keys to be pressed. Although little training is needed to use a keyboard, they do pose a health risk to operators who do a lot of continuous typing – the health risk is known as RSI (repetitive strain injury).

Pointing devices

The most common pointing devices are the mouse and the trackball. They are used to control a cursor on screen or to select options from menus. The mouse usually has two buttons (used for selection and other functions) together with a scroll up/down wheel.

Trackballs use a ball on the top or side of the device which enables control of a cursor or selecting options from a menu. They are often the preferred device in control rooms since they don't require as much desk space as a mouse and are also less likely to cause such problems as RSI (caused by repeated clicking of a mouse button).

Microphones

Microphones are used to input sound into a computer. When the microphone picks up sound, a diaphragm vibrates producing an electric signal. A sound card in the computer converts the signal into digital values which can be stored in its memory.

Voice recognition uses microphones. It is used to identify if a 'known' person is speaking in, for example, a security system. The software compares wave patterns from the person's voice with wave patterns stored in memory. If the wave patterns match, then the correct person is identified.

Speech recognition also uses microphones but is very different to voice recognition. This works by first converting speech patterns into a digital form. The digital image is broken up into phonemes which are then compared with words stored in the built-in dictionary. The spoken words can then be recognised.

Touchscreens

Touchscreens allow selections to be made by simply touching an icon or menu option on a screen. They are also used in many devices to allow input via a 'virtual keyboard'.

Mobile phones and tablets are some of the biggest users of touchscreen technology. The most common systems are capacitive, infra-red or resistive.

	Capacitive	Infra-red	Resistive
Technology	<ul style="list-style-type: none"> uses layers of glass that acts a capacitor when the top layer is touched, the electric current changes microprocessor works out the coordinates of where the screen was touched 	<ul style="list-style-type: none"> uses glass and can either detect heat or uses infra-red sensors to detect touch microprocessor works out where the screen was touched based on sensor/heat data 	<ul style="list-style-type: none"> uses upper layer of polyester and bottom layer of glass when top layer touched, it completes a circuit microprocessor works out coordinates of where the screen was touched
Benefits	<ul style="list-style-type: none"> this is a medium-cost technology good visibility in strong sunlight allows multi-touch capability very durable 	<ul style="list-style-type: none"> allows multi-touch capability can use bare fingers, gloved hand or stylus good screen durability 	<ul style="list-style-type: none"> relatively inexpensive technology can use bare fingers, gloved hand or stylus
Drawbacks	<ul style="list-style-type: none"> can only use bare fingers or a conductive stylus 	<ul style="list-style-type: none"> relatively expensive technology heat-sensitive systems only allow bare fingers to be used 	<ul style="list-style-type: none"> poor visibility in strong sunlight doesn't allow multi-touch capability screen is vulnerable to scratches

Sensors

Sensors send data from the real world to a computer. They often require an analogue to digital converter (ADC) to change the data into a format that the computer (or microprocessor) can understand.

Sensors form part of many monitoring or control systems. Monitoring simply reports the status of a system via a computer/microprocessor. Control involves altering devices (e.g. open/close valve, switch on/off a heater or turn

a pump on/off) so that effectively the output affects the next input to the computer. Many sensors exist depending on the application being monitored or controlled, for example, temperature, moisture/humidity, light, infra-red/motion, pressure, acoustic/sound, gas, pH or magnetic field.

Monitoring systems include: monitoring a patient's vital signs in a hospital, burglar/intruder alarm systems, checking key data in a car engine or monitoring pollution levels in a river or the air.

Control systems include: traffic light control, greenhouse environment, street lighting, anti-lock braking system on a vehicle or control of a chemical or nuclear process.

Interactive whiteboards

Interactive whiteboards allow computer images to be displayed using a digital projector. They allow users to write on the whiteboard and the text/images created can then be stored in a computer memory.

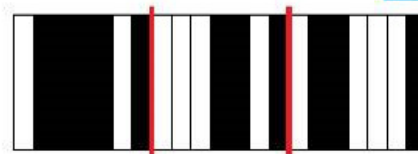
Any image projected onto the screen can be annotated or even changed and the annotated version can then be saved for later use.

Common errors

- Students often confuse voice recognition systems and speech recognition systems.
- Many students are unable to distinguish between monitoring and control; it is important to read and understand the key differences highlighted above.
- Students often claim that sensors control devices rather than just send data to a computer or microprocessor.
- Students also often claim that sensors only send data once something happens – sensors continuously send data to computers and microprocessors; these devices then determine if the received data has changed and some action is necessary.

Sample question

- 1 a What is the binary value of the section of barcode shown below?
 b (i) What is a QR code?
 (ii) Describe two applications that use QR codes.



Student answer

1 a 0111101/0001101/0110001

b (i) Quick response code:

- made up of a matrix of filled in dark squares on a light background
- read by a camera on a smart phone or tablet and uses an app to interpret the data.

(ii) Two applications include:

- a taxi company uses QR codes on business cards; when scanned the taxi company are automatically contacted and/or advertises their services
- a restaurant uses QR codes on advertising hoardings and/or magazines or flyers; when scanned, the QR code gives sample menus, links to websites, functions, and so on.

Examiner's comments

In part a, the three answers are correct. It may be necessary to make these 8-bit codes; in which case, an additional zero needs to be put in FRONT of each binary value.

In part b(i), the question asked is a little vague; giving additional information as shown in the Student answer above is good practice, especially if the question carries more than one mark.

In part (ii), the two applications can have the same purpose (e.g. link to a website) but the actual applications themselves need to be clearly different (as they are in the answers given).

● Sample question



2 An airport perimeter is being monitored for pollution levels. NO_x and CO_2 sensors are being used and the collected data is sent to a computer.

Describe how the sensors and computer are used to warn environmentalists of higher than acceptable pollution levels.

Student answer

- NO_x and CO_2 sensors continuously send data to the computer.
- The data is first sent to an ADC.
- The computer has the maximum NO_x and CO_2 levels pre-stored.
- The computer has data from previous monitoring stored in memory.
- The incoming sensor data is compared with the pre-stored values.
- If any of the stored levels are exceeded by sensor readings then the computer automatically prints out graphs and/or a report for the scientists.
- If levels are dangerously exceeded, then the computer may also generate an audible warning.
- The computer uses previous data to predict trends in pollution levels and can indicate which pollution levels (or where pollution levels) are predicted to rise and can suggest action.

Examiner's comments

These questions usually carry 4 marks or more and require the student to understand the role of the sensors and the computer. They also require the student to show an understanding of how incoming data is checked and what action (if any) is taken. Since every application is different, students need to be able to adapt their answers to suit the question.

Examiner's tips

Most questions of this type have a generic format. If a control application is given, it is necessary to remember that the computer will send signals to actuators to operate valves (for example) and it may be necessary to use a DAC so that the devices can be controlled.

● Output devices



An output device is any piece of computer hardware equipment used to communicate the results of data processing carried out by a computer; the device converts the computer's digital information into a human-readable form.

Printers

The three most common types of printer are: laser, inkjet and dot matrix. The following table compares the features of all three printers.

Type of printer	Advantages	Disadvantages
Laser printer	<ul style="list-style-type: none"> • very high quality of printing • relatively inexpensive to buy printer • large toner cartridges and large paper trays • very fast printing of multiple documents 	<ul style="list-style-type: none"> • can be expensive to maintain (e.g. fusers) • produce health hazards such as ozone or toner particles in the air
Inkjet printer	<ul style="list-style-type: none"> • high quality of printing • very suitable for printing photos • inexpensive to buy printer 	<ul style="list-style-type: none"> • ink is expensive to buy and quickly runs out on a large print run • usually have small paper trays • can be noisy in operation compared to laser printer
Dot matrix printer	<ul style="list-style-type: none"> • not adversely affected by damp or dirty/dusty atmospheres • allows use of multi-part stationery (i.e. carbon copies) • allows use of continuous/fan-folded stationery 	<ul style="list-style-type: none"> • relatively expensive to buy printer • poor print quality • very noisy and very slow at printing

The full description of how these printers work can be found in the Student's Book. However, to summarise:

- Laser printers rely on positive and negative charges on the print drum and paper; (dry powder) toner sticks to the paper in only certain areas and is permanently fixed using a fuser.
- Inkjet printers rely on a liquid ink system which sprays ink onto the paper line by line as the paper advances; the ink system uses either thermal bubble or piezoelectric technologies.
- Dot matrix printers use a matrix of pins which strike an inked ribbon (either black or up to 4 coloured) to produce characters in a matrix on the paper, e.g.

```

THE QUICK BROWN FOX
JUMPED OVER THE LAZY
DOG. THE QUICK BROWN FOX
JUMPED OVER THE LAZY DOG.
@123456789

```

3D printers

3D printers produce actual solid objects which work; they are built up in the printer in a number of very thin layers of varying materials, such as: powdered resin, powdered metal, ceramic powder or even paper. Some 3D printers use binder technology which involves the use of a binder (a type of 'glue') to form the solid layers.

Since they build up objects layer by layer, they are often referred to as additive. Some of the present applications include: making prosthetic limbs, assisting precision reconstructive surgery, manufacture of aerospace parts, fashion and art, and making parts no longer in production by conventional manufacturing.

2D/3D cutters

These are used to cut into materials to form 3D objects and are controlled by computers and software (such as CAD/CAM). Common materials include: glass, crystal, metal, polymers and wood.

Actuators

See section on sensors on page 41.

Loudspeakers

Loudspeakers convert analogue voltages into sound. If the output is from a computer, the digital signals are first converted into analogue voltages using a DAC.

The rate at which the DAC can translate digital output into analogue voltages is called the sampling rate – usually 44 100 samples/second for a typical 16-bit system.

LCD and LED monitors

Modern liquid crystal display/diode (LCD) monitors and televisions are back-lit using light-emitting diodes (LEDs) rather than cold cathode fluorescent lamps (CCFLs). LCD monitors and televisions need to be back-lit since LCD does not produce its own light. The advantages of using LEDs (rather than CCFLs) to back-light the display are:

- they reach maximum brightness much faster
- they give a whiter light making the image more vivid
- the brighter light improves colour definition
- the displays can be much thinner and lighter in weight
- LEDs last almost indefinitely
- they consume much less power and therefore also run much cooler.

Newer monitors and televisions use organic light-emitting diodes (OLEDs). These produce their own light and therefore don't need any form of back-lighting. The advantages of using OLED when compared to older LED technology include:

- they allow for very thin and very flexible screens which are very light weight
- they produce a much brighter light than LEDs
- they don't require any form of back-lighting
- they allow for a much larger field of view.

Light projectors

There are two common types: digital light projectors (DLP) and LCD projectors.

DLP relies on millions of micro mirrors which can switch on or off several thousand times a second creating various shades of grey. Colour filters allow the shades of grey to be converted into colours which are projected onto a large screen thus representing the output from the computer.

LCD projectors use chromatic-coated mirrors which reflect light at different wavelengths. The light components pass through three LCD screens which are then recombined using a prism to produce the colour image which is projected onto a large screen.

Common errors

- The statement: 'laser printers do not use ink' is incorrect since they do use ink but it is in a dry powdered form unlike inkjet printers which use liquid ink.
- Statements such as 'laser printers are faster' or 'dot matrix printers are a lower quality' are never worth any marks since they don't say what they are faster at or what is a lower quality. It is necessary to be clear and state, for example: 'laser printers print out pages faster than ...'.

Sample question

- 3 a** Give one advantage and one disadvantage of laser printers, inkjet printers and dot matrix printers when compared to each other. Different advantages and disadvantages are needed in each case.
- b** Give one application of each of these three types of printer. Give a different application in each case.



Student answer

3 a Laser:

- advantage – larger toner cartridges/paper trays
- disadvantage – maintenance can be expensive

Inkjet:

- advantage – inexpensive to buy the printer
- disadvantage – can quickly run out of ink or paper

Dot matrix:

- advantage – can work in damp or dirty/dusty atmospheres
- disadvantage – very poor printing quality

b Laser: printing out 1000s of leaflets or flyers in colour

Inkjet: one-off high quality photographs on glossy paper

Dot matrix: printing labels on the factory floor

Examiner's comments

In part a, it is important to realise that three different advantages and disadvantages must be given – many marks are lost for not doing this; the question is quite specific. The answers can be essentially generic; extra marks would not be gained by giving applications in part a since these marks have been allocated to part b.

In part b, although it doesn't say so in this question, look out for questions that require applications to match the advantages and disadvantages given in an earlier part of the question.

Sample question

- 4 a Describe three differences between LCD monitors back-lit using LEDs or with CCFLs.
- b Describe two applications which make use of the unique features of OLED technology.

Student answer

4 a LEDs reach brightness faster.
LEDs give off a whiter light.
LEDs last almost indefinitely.

b Use of smart watches that can bend easily around the wrist
Curved thin sheets used as advertising posters on curved surfaces such as underground stations

Examiner's comments

In part a, it is necessary to give a little more depth to the answers since a description was asked for. For example, 'LEDs reach brightness faster so there is no need for them to warm up first before reaching full effectiveness. LEDs give a whiter light which sharpens the image and makes the colours more vivid. LEDs last indefinitely which makes the technology more reliable and more consistent in operation.'

In part b, it is important to give two different applications. It would be worth adding the following to the answers in both applications: '... due to the flexible nature of OLEDs, it is possible to make smart watches (curved screens) ...'

Examiner's tips

Always remember to give additional information to support your answer in questions where a description is required.

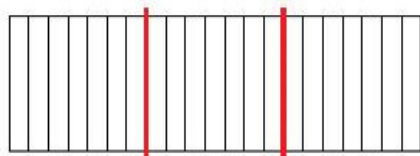
Look carefully at the number of marks allocated to determine how many 'points' need to be made when giving descriptions or explaining a given topic.

Exam-style questions

- 1 Which computer terms are being described here?
- 'splitting up of a solid object into a number of very thin slices'
 - 'machine readable code made up of an array of dark and light squares; often used to store URLs or other important advertising information'
 - 'device which converts sound into electrical signals'
 - 'technology that is used to identify if a known person is speaking'
 - 'device that sends data from the real world to a computer; it often requires the use of an ADC for the computer to process this data' [5 marks]

- 2 a Complete the following diagram to show the barcode for the three binary numbers:

0 1 0 0 0 1 1 / 0 1 0 1 1 1 1 / 0 1 1 1 0 1 1 [3 marks]



- b Name and describe one application that uses barcode technology. [3 marks]
- 3 a What is meant by the term 'sampling rate' when referring to DACs? [2 marks]
- b A music CD is produced where each piece of music is sampled 44 100 times a second. Each sample is 32-bits and the music is stored in stereo (two separate tracks) format.
- Calculate:
- how many bytes per second are sampled
 - how many KB/second are sampled. [3 marks]
- c (i) Calculate how much memory is needed to store a 4-minute music track using your answer to part b(ii). [2 marks]
- (ii) Calculate how many 4-minute music tracks could be stored on a CD with an 800MB storage capacity. [1 mark]
- 4 a Indicate with a tick (✓) which of the following statements about OLEDs are correct. [5 marks]

Statements	True (✓)
Because of plastic organic layers, OLEDs are much thicker and heavier than conventional LCD/LED formats	
OLEDs give a brighter light than LEDs leading to more vivid colours	
OLEDs produce a much larger field of view than LCD/LED formats	
OLEDs require some form of back-lighting so that the screen can be lit up	
OLEDs consume more power than LCD/LED formats and thus produce more heat	

- b Describe how OLEDs generate the screen images on monitors and televisions. [3 marks]

- 5 a A car engine management system uses temperature and carbon monoxide sensors which feed data to an on-board microprocessor.

Water is used to cool the engine and valves can be opened or closed to control the flow of water in the engine. The amount of air entering the engine is used to control the levels of carbon monoxide – again valves can be opened or closed to control the air entering the engine.

Describe in detail how the sensors and microprocessor are used to control the temperature and carbon monoxide levels in the engine. [5 marks]

- b Name a suitable sensor for each of the following applications. A different sensor needs to be given for each application. [5 marks]

Application	Suitable sensor
measure/control the dampness of the air in a paint spray booth in a factory	
automatically switch on a car's headlights when it becomes dark	
detect intruders entering a building in a burglar/intruder alarm system	
measure the acid pollution levels in a river at a number of points	
measure the speed of rotation of wheels on a car in an anti-lock braking system	

- 6 a A laser printer is used to print out a document. The table shows all the stages in printing out the document. But they are not in the correct order. By writing the numbers 1 to 11 in the right-hand column, put each stage in the correct order.

The first one has been done for you. [10 marks]

Stage	Order
data from the document is sent to a printer driver	1
data sent to the printer is stored in the printer buffer	
discharge lamp removes all electric charge from the printer drum at the end of the printing process	
negatively charged sheet of paper is rolled over printer drum	
paper goes through a fuser which fixes ink permanently to the paper	
printer driver ensures data being sent to printer is in the correct format	
printer drum is coated in positively charged toner	
printer drum is given a positive charge	
toner on printer drum now sticks to the paper to produce text and images	
to prevent paper sticking to the printer drum, the electric charge on the paper is removed after one rotation of the drum	
when printer drum rotates, laser beam scans across it removing the positive charge, leaving a negatively-charged print area	

- b Give one different application for:
- (i) laser printer
 - (ii) inkjet printer
 - (iii) dot matrix printer.
- [3 marks]

It is illegal to photocopy this page

6

Memory and data storage

Key objectives

Objectives of this chapter are to revise:

- file formats (such as MIDI, MP3/4 and JPEG)
- file compression methods
- primary, secondary and off-line storage
- magnetic, optical and solid-state media

Key definitions

Term	Definition
Musical instrument digital interface (MIDI)	This system is associated with the storage of music files. MIDI files consist of lists of commands which instruct a device how to produce a musical note/sound
MPEG-3 (MP3)	Uses audio compression to store music in an MP3 file format
MPEG-4 (MP4)	Can store multimedia rather than just music
Joint Photographic Experts Group (JPEG)	JPEG files use a LOSSY format file compression method. JPEG is used to store photographs as a reduced file size
Lossless	All the data from the original files are reconstructed when the file is again uncompressed
Lossy	Unnecessary data is eliminated forming a file that can't be reconstructed to get back to the original file
Primary memory	Includes RAM, DRAM, SRAM and ROM
Secondary storage	Includes hard disk drives, solid-state drives, and off-line storage (CDs, DVDs, DVD-RAM and blu-ray disks)

File formats

Musical instrument digital interface (MIDI)

This system is associated with the storage of music files. No sounds are stored (as in the case of MP3 and MP4); it is a form of communications protocol allowing musical instruments to interact. MIDI files consist of lists of commands which instruct a device how to produce a musical note/sound; for example: NOTE on/off or KEY PRESSURE, and so on. Also included would be a pitch byte and a velocity byte. The electronic instruments and computer both need MIDI interfaces to allow them to communicate.

Since MIDI files don't contain actual audio tracks, their size is relatively small. This makes them very suitable for storing sounds/music on devices with limited memory; for example, storing ring tones on a mobile phone.

MPEG-3 (MP3) and MPEG-4 (MP4) files

MP3 uses audio compression to store music in an MP3 file format. These can often be 90% smaller than comparable CD music files. They use a lossy format. The music quality is retained by using perceptual music shaping (i.e. removing sounds that the human ear can't clearly distinguish).

MP4 files are slightly different to MP3 files since they can store multimedia rather than just music (i.e. music, video, photos and animation).

Joint Photographic Experts Group (JPEG) files

JPEG files use a lossy format file compression method. JPEG is used to store photographs as a reduced file size. They rely on certain properties of the human eye (e.g. its inability to detect small brightness differences or colour hues).

Text and number file formats

Text is stored in ASCII format and text files are usually stored in a lossless format.

Numbers can be stored as real, integer, currency, and so on. Lossless format is used since accuracy of data is very important.

Lossless and lossy file formats

With lossless file compression, all the data from the original files is reconstructed when the file is again uncompressed.

With lossy file compression, unnecessary data is eliminated forming a file that can't be reconstructed to get back to the original file.

● Common errors

- Students often believe that MIDI files store actual music/sounds.
- Students often confuse lossy and lossless file compression formats.



● Sample question

1 a Give two differences between MIDI files and MP3 files.

b An image is 1200 pixels by 1600 pixels.

- (i) Calculate the number of megabytes used to store this image.
- (ii) The file is to be stored in a jpeg format; give one advantage and one disadvantage of doing this.



Student answer

1 a MIDI files:

- don't store actual music or sounds
- consist of lists of commands indicating how a particular note/sound can be produced.

MP3 files:

- use lossy file compression format
- store music/sounds in an actual music format
- use perceptual music shaping to retain music quality.

b (i) $1200 \times 1600 = 1\,920\,000$ bytes
 $= 1\,920\,000 / 1\,048\,576 \text{ MB}$
 $= 1.83 \text{ MB}$

- (ii) Advantage: saves memory in the computer.
 Disadvantage: quality of the image is reduced.

Examiner's comments

In part a, it is important when choosing two differences to make sure they are not just opposites of each other. For example,

'MIDI files don't store actual music files'

'MP3 files store actual music files'

These two answers would be regarded as describing only one difference since the second answer is just the converse of the first answer.

In part b(i), it is important that ALL your working is shown; if a mistake is made somewhere in the calculation, credit will still be given for showing the correct steps in the calculation. If the method is not shown and an answer, such as 14.5 MB is given, then NO marks could be awarded since the answer is clearly wrong and no supporting evidence is supplied.

In part (ii), the same comments made in part a apply.

● Sample question

2 Explain how the sentence below would be stored with a reduction of about 40% (ignoring spaces):

‘COMPARE TEXT FILES IN A COMPUTER AFTER FILE COMPRESSION’

Student answer

Using a word dictionary of 1 = COMP, 2 = FILE would give:

‘1ARE TEXT 2S IN A 1UTER AFTER 2 1RESSION’

This gives a reduction from 46 characters to 32 characters which is about 31%.

Examiner’s comments

Whilst this is a good attempt, the sentence could be further compressed by adding 3 = TER to the dictionary, giving:

‘1ARE TEXT 2S IN A 1U3 AF3 2 1RESSION’

This would give a file reduction of about 40% as required in the question. Credit would be given for a good attempt, but at least 1 mark would be lost for not achieving the 40% file reduction.

Examiner’s tips

Always show all your working when carrying out some form of calculation; the number of marks allocated to a question give a strong clue as to the amount of information you need to give.

When making comparisons, make sure you don’t just give the converse of your earlier response.

● Memory and storage

Primary memory

Primary memory refers to Random Access Memory (RAM) and Read Only Memory (ROM).

- RAM – this memory is volatile/temporary in nature.
 - It is used to store data, programs or parts of the operating system *currently* in use.
 - The memory can be written to or read from.
 - RAM can be dynamic (DRAM) or static (SRAM). DRAM needs to be constantly refreshed to retain the data; it uses millions of transistors and capacitors. SRAM doesn’t need to be constantly refreshed to retain its data; it uses flip flops.
- ROM – this memory is non-volatile/permanent in nature.
 - It is used to store the start-up procedures or BIOS (basic input/output system).
 - The memory can only be read and not written to.

Hard disk drive (HDD)

Hard disk drives use circular platters coated in magnetic material. Several read–write heads are used so that all the surfaces can be accessed; data is stored in sectors and tracks in blocks.

Access to data is slower than with RAM. Many applications require the read–write heads to move in and out several times to seek the appropriate data blocks. This is referred to as latency – the time taken for a specific block of data on a track to rotate around to the read–write head.

Solid-state drives (SSD)

Solid-state drives have no moving parts; therefore all data is retrieved at the same rate and latency is not an issue. Most common types of SSD use NAND chips to control the movement of electrons (and therefore data). Some solid-state systems use EEPROMs (electronically erasable programmable read only memory) which utilise NOR chips.

There are a number of advantages in using SSDs when compared to HDDS:

- they are more reliable and robust (no moving parts)
- they are much lighter in weight and also much thinner
- they consume less power and also run much cooler
- no need to wait for SSD to ‘get up to speed’ and also have a faster data access rate.

Off-line storage

Off-line storage devices include: CD/DVD/DVD-RAM (all optical), blu-ray disks (optical media), flash memory/memory stick/SD-XD cards (solid-state devices) and removable hard disk drives (magnetic media).

CD/DVD

CDs and DVDs use a red laser light to write and read the data. Each disk has a single spiral track that runs from the centre to the outer edge. They can be ‘-R’ (write once and can then only be read) or ‘-RW’ (can be written to and read from several times). DVDs can hold more data than CDs due to dual layering technology (i.e. the disks have two individual recording layers).

DVD-RAM

This technology is nearly obsolescent. It uses a number of concentric tracks (unlike CDs and DVDs) enabling simultaneous read–write operations to take place. They have great longevity which makes them ideal for archiving of data.

Blu-ray disks

Blu-ray disks use blue laser light to read and write the data. Since the wavelength of blue light is 405 nm (compared to red light wavelength of 650 nm), they can store considerably more data. They use a single polycarbonate disk (unlike DVDs which use a sandwich of two polycarbonate disks); because of this, blu-ray disks don’t suffer from birefringence (i.e. light refraction into two separate beams which can cause reading errors).

Flash memory/memory sticks

These use solid-state technology and usually connect to the computer through the USB port. They are very small and lightweight which makes them ideal for transferring files and photos between different computers.

As with SSD, they use NAND chips and have no physical moving parts. They need to be read every now and then to retain their data integrity and it is important to follow the correct removal procedure to avoid corrupting the storage device.

Common errors

- Students often confuse RAM and ROM (particularly with reference to volatile and non-volatile).
- Students often believe that solid-state memories are examples of magnetic media.

Sample question

3 A remote-controlled model aeroplane uses RAM and ROM. The remote control frequency is set at the manufacturing stage and shouldn't be altered. The operator can change flight manoeuvres, for example, by entering new instructions from a touch screen interface.

Explain the use of RAM and ROM in this model aeroplane. Describe, with reasons, a suitable storage device to store the operator's instructions for later use.

Student answer

RAM is used to hold the flight path/manoeuvres/instructions and data to control the aeroplane which are currently in use.

ROM will hold the data not to be changed (e.g. remote control frequency), start-up routines and the set routines (e.g. the function of the buttons on the remote control).

A solid-state memory device would be most suitable for storing user's instructions; the device is lightweight and has no moving parts (and is therefore more robust) in case the aeroplane crashes.

Examiner's comments

The description of RAM and ROM in the Student answer is very relevant. It is tempting to give a general description of RAM (it is volatile and can be written to/read from) and ROM (non-volatile and can only be read from). However, such general answers would be unlikely to gain any marks. When a question refers to a very specific application, it is essential that the Student answer refers to this application, as much as possible, to gain the marks. The answers given refer to the model aeroplane which means they will gain full credit.

Sample question

- 4 a** Describe four advantages of SSDs compared to HDDs which makes them more suitable for use in a laptop computer.
- b** Describe three differences between DVDs and blu-ray disks.

Student answer

- 4 a • No moving parts so more robust in a portable device.
 • Lightweight which is a big advantage in a portable device.
 • Very thin which helps in the construction of a laptop computer.
 • Low power consumption which means laptop battery lasts longer and also runs cooler.
- b DVDs use red laser light, use a sandwich of two polycarbonate disks and have a track pitch which is larger than blu-ray disks.
 Blu-ray disks use blue laser light, use a single polycarbonate disk and have built-in encryption routines.

Examiner's comments

In part a, it is important to refer to the application given (i.e. a laptop computer). General answers would lose many marks as would answers with no relevance to the portability of laptops (e.g. faster data access).

In part b, it is important to give three distinct differences and not simply give the converse of earlier answers.

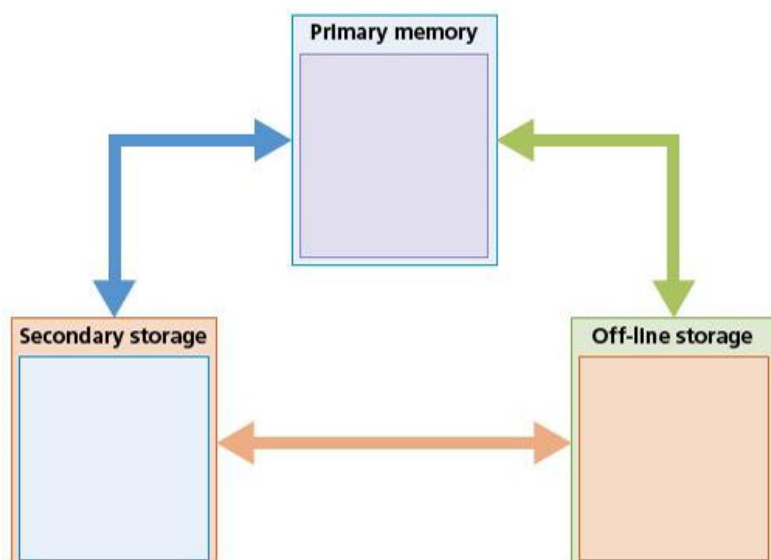
Examiner's tips

In question 3 it is clear that answers MUST refer to the application described otherwise many marks could be lost.

When giving answers requiring comparisons, it is important that the differences given are distinctly different.

Exam-style questions

- 1 a Explain the difference between lossless file compression and lossy file compression. [2 marks]
- b A CD contains 12 music tracks which are the following lengths (in minutes): 3, 7, 4, 3, 5, 6, 4, 5, 4, 7, 8, 8.
- (i) Calculate how much memory is used for the 12 tracks if each minute of music requires 12 MB of storage.
- (ii) If the tracks are to be stored in MP3 format, each music track will be reduced in size by 90%. Calculate how much storage the 12 tracks would now require.
- (iii) Explain how MP3 retains the quality of the original music tracks. [5 marks]
- 2 Complete the following diagram by writing the appropriate memory/storage device(s) from the following list. [8 marks]
- a blu-ray disk
 b DVD/CD
 c hard disk drive (HDD)
 d random access memory (RAM)
 e removable hard disk drive
 f read only memory (ROM)
 g solid-state drive (SSD)
 h USB memory stick



- 3 In the diagram below, seven descriptions are shown on the left and six devices are shown on the right. By drawing arrows, connect each description to the correct device – each device may be used once, more than once or not at all. [7 marks]

uses flip flops to hold each bit of memory; doesn't need to be refreshed to retain data

SSD

has no moving parts and is therefore a very robust and reliable technology

Blu-ray

makes use of blue laser light to save and write data on the storage media

DVD-RAM

optical media that uses concentric tracks to allow simultaneous read-write operations

DRAM

storage device where data is stored by controlling the movement of electrons within a NAND chip

SRAM

media that uses only one polycarbonate disk to remove the effects of birefringence

ROM

memory that contains capacitors and transistors and needs to be constantly refreshed to retain data

7

High- and low-level languages

Key objectives

Objectives of this chapter are to revise:

- programming languages:
 - high-level languages
 - low-level languages
 - assembly language
 - machine code
- translators:
 - compilers
 - interpreters
 - assemblers

● Key definitions

Term	Definition
Computer program	A list of instructions that enable a computer to perform a specific task
Translator	Translators convert a program into binary instructions that computers can understand

● Computer programs

A computer program is a list of instructions that enable a computer to perform a specific task. Computer programs can be written in high-level languages and low-level languages depending on the task to be performed and the computer to be used. Most programmers write programs in high-level languages.

High-level languages

High-level languages enable a programmer to focus on the problem to be solved and require no knowledge of the hardware and instruction set of the computer that will use the program. High-level programming languages are portable and can be used on different types of computer.

Low-level languages

Low-level languages relate to the specific architecture and hardware of a particular type of computer. Low-level languages can refer to machine code, the binary instructions that a computer understands, or assembly language that needs to be translated into machine code.

Assembly languages

Assembly language is a low-level programming language that needs to be translated into machine code by an assembler.

Machine code

Machine code is the binary instructions that a computer understands. No translation is required.

See the Student's Book for examples of each type of programming language.

● Common errors

- Students often say that machine code needs translation.
- Students often reverse the definitions of high- and low-level languages.

● Sample question □

- 1 a Give one advantage of writing a program in a high-level language.
- b Give one advantage of writing a program in assembly language.
- c Give one advantage of writing a program in machine code.

Examiner's tip

Read the question carefully; does it ask for advantages or benefits or disadvantages or limitations? Answer the question as set; both advantages and disadvantages require a comparison. No comparison is required for benefits and limitations.

Student answer

- 1 a *Programs are easy to understand.*
- b *To make use of hardware.*
- c *Programs do not need to be translated.*

Examiner's comments

In part a, the answer is true but doesn't quite answer the question, which asked for advantages not benefits so a comparison is required. A better answer would be 'Programs written in a high-level language are easier to understand by programmers than those written in a low-level language.'

In part b, the answer is a little vague. A better answer would qualify the hardware, for example specialist hardware.

In part c, the answer given is correct; it is the only answer that does not have to be qualified by a comparison since it is an advantage over both high-level and assembly languages.

Examiner's tip

Take care with advantages and disadvantages, students sometimes read disadvantage as advantage and give the wrong answer.

● Translators □

In order to be used by a computer, programs need to be translated into the binary instructions, machine code, that the computer understands.

Compilers

A compiler is a computer program that translates a program written in a high-level language (HLL) into machine code so that it can be directly used by a computer to perform a required task.

Interpreters

An interpreter is a computer program that reads a statement from a program written in a high-level language performs the action specified and then does the same with the next statement and so on.

Assemblers

An assembler is a computer program that translates a program written in an assembly language into machine code so that it can be directly used by a computer to perform a required task.

Summary

Compiler	Interpreter	Assembler
Translates a high-level language program into machine code.	Executes a high-level language program one statement at a time.	Translates a low-level language program into machine code.
An executable file of machine code is produced.	No executable file of machine code is produced.	An executable file of machine code is produced.
One high-level language statement can be translated into several machine code instructions	One high-level language program statement may require several machine code instructions to be executed.	One low-level language statement is usually translated into one machine code instruction.
Compiled programs are used without the compiler.	Interpreted programs cannot be used without the interpreter.	Assembled programs can be used without the assembler.
A compiled program is usually distributed for general use.	An interpreter is often used when a program is being developed.	An assembled program is usually distributed for general use.

See the Student's Book for examples of program code before and after translation.

● Common errors □

- The statement: 'interpreters translate high-level program statements into a machine code program' is incorrect since the machine code statements are just executed and not stored.
- Statements such as 'low-level programs are faster' or 'high-level programs are slower' are never worth any marks since they don't say faster or slower at what; it is necessary to be clear and state, for example: 'low-level programs complete the same task faster than ...'.

● Sample question □

- 2 a Describe two differences between a compiler and an interpreter.
- b State, with a reason, a situation where you would use each one. Each situation must be different.

Student answer

- 2 a A compiler translates a whole program at one go, an interpreter translates a line at a time. A compiler produces a translated program, an interpreter does not.
- b Compiler: for distributing a program for general use, because the program will not require the compiler to run.
Interpreter: for developing a program as errors are shown on a line by line basis.

Examiner's comments

In part a, the student has given two differences by explaining what happens in the same situation in each case. For the first difference, it would have been more accurate to state that the interpreter executes the program a line at a time.

In part b, the student has identified two different situations correctly. Both reasons given identify what is happening but not why it is useful. For the compiler, the users do not have to download or purchase the compiler. For the interpreter, the developer can resume the development of the program immediately one error has been corrected without having to correct every error in the program.

Examiner's tip

Always remember to explain why as well as identifying what to support your answer in questions where a reason is required.

Ensure that each part of your answer adds value rather than repeating a point already made.

Exam-style questions

- 1 Which translators are being described here?
 - a 'translates a high-level language into machine code'
 - b 'translates a low-level language into machine code'
 - c 'executes a high-level language program line by line' [3 marks]

- 2 Identify the type of language that the following programs have been written in.
 - a LDA NoOne
ADD NoTwo
STO Answer
 - b Answer := NoOne + NoTwo
 - c 1 12
4 13
0 1A
 - d 0001 00010010
0100 00010011
0000 00011010 [4 marks]

- 3 Name **three** different types of language translator and explain what each one is used for. [6 marks]

- 4 a Give **two** disadvantages of writing a program in a high-level language. [2 marks]
 - b Give **two** disadvantages of writing a program in a low-level language. [2 marks]

8

Security and ethics

Key objectives

Objectives of this chapter are to revise:

- security and data integrity
- cookies
- loss of data and data corruption
- firewalls
- security protocols
- encryption techniques
- denial of service attacks
- computer ethics
- free software, freeware and shareware

● Key definitions □

Term	Definition
Hacking	A way of gaining illegal access to a computer system
Cracking	The editing of program source code so that it can be exploited or changed for a specific purpose
Viruses	Program code that can replicate/copy itself with the intention of deleting/corrupting data/files or causing the computer to malfunction
Phishing	Sending legitimate-looking emails to encourage users to give out personal data
Pharming	Malicious code installed to redirect users to a fake website
Spyware/key logging software	Gathers data by monitoring key presses on user's keyboards
Cookies	Pieces of data which allow detection of web pages viewed by a user and store their preferences
Accidental or malicious loss of data	This could be due to accidentally deleting a file or to malicious attack from viruses or hacking
Firewalls	Examines traffic between user's computer and a public network. They can help to prevent viruses or hackers entering a user's computer
Security Sockets Layer (SSL)	Allows data to be sent and received securely across a network
Transport Layer Security (TLS)	A protocol that is designed to ensure that no third party may eavesdrop or tamper with any message
Symmetric encryption	A secret key which makes a message unreadable unless the recipient also has the decryption key
Asymmetric encryption	A form of encryption requiring both a public and a private key
Authentication	Used to verify that data comes from a trusted source
Denial of Service attacks	an attempt at preventing users from accessing part of a network, notably internet servers
Ethics	A set of principles set out to regulate use of computers
Free software	Users have the freedom to run, copy, change or adopt free software
Freeware	Software that a user can download free of charge
Shareware	Users are allowed to try out shareware free for a trial period

● Security and data integrity □

Hacking

- Hacking is a way of gaining illegal access to a computer system.
- This can lead to identity theft and loss or corruption of data.
- The risk of hacking can be minimised by using strong passwords and firewalls.

It is illegal to photocopy this page

Cracking

- Cracking is the editing of program source code so that it can be exploited or changed for a specific purpose (mostly an illegal act).
- It is often done for malicious purposes, e.g. modifying legitimate software to do something like redirect a user to a fake website.
- It is difficult to guard against; software engineers need to make the act of breaking into the software nearly impossible (i.e. make it difficult to identify 'back doors').

Viruses

- A virus is program code that can replicate/copy itself with the intention of deleting/corrupting data/files or causing the computer to malfunction.
- They can cause the computer to run slow (due to the hard disk filling up with data, for example) or crash (e.g. due to deletion of some key .exe files); they can also cause some software to run abnormally (due to loss or corruption of files/data).
- The risk of viruses can be minimised by running anti-virus software or not opening emails or software from unknown sources.


Phishing

- Legitimate-looking emails are sent to users; on opening the email, the user could be asked to supply personal or financial details or they may be asked to click on a link which sends them to a fake/bogus website where the user could be asked to supply personal data.
- Once the user is sent to the fake/bogus website, they may be asked to give out personal or financial data.
- Many ISPs filter out phishing emails; the user should also be very cautious about opening emails from unknown sources.

There are a number of signs to look out for in phishing emails:

- messages containing poor spelling and grammar
- a message that asks for personal information or where you didn't initiate the action
- a message that makes unrealistic threats or promises, or financial rewards, for example, cash prizes, lottery winnings, investment or pensions advice.

Pharming

- Malicious code is installed on a user's computer or web server; the code redirects the user to a fake or bogus website without their knowledge.
- Once the user is sent to the fake/bogus website, they may be asked to give out personal or financial data.
- Some anti-spyware software can identify and remove pharming code on the hard drive; the user should also look out for clues that they are being redirected (websites which are safe can usually be identified by https or by the green padlock  sign in the status bar).

Wardriving

- This is the act of locating and using wireless internet connections illegally.
- Could lead to stealing of internet time and bandwidth; also user's passwords and other data may be intercepted.
- Prevented by use of WEP (wired equivalent privacy) encryption; use of complex passwords before accessing the internet and use of firewalls to prevent outsiders gaining access.

Spyware/key logging software

- Gathers data by monitoring key presses on user's keyboards and sending the data back to the person who sent the spyware.
- Sends important data, such as passwords, back to the originator of the spyware; it can also allow the originator to change settings on the user's computer.
- Prevented by use of anti-spyware; use of mouse to select characters from a drop down box to enter passwords rather than using a keyboard.

Cookies

- Pieces of data which allow detection of web pages viewed by a user and store their preferences.
- Create an anonymous user profile (e.g. user's preferences).
- Cookies can be deleted from the user's desktop, although this can remove some of the features of certain websites.

● Data integrity



Data can be lost on a computer due to:

- accidental or malicious mal-operation
- hardware malfunction
- software malfunction.

Accidental or malicious loss of data

- This could be due to accidentally deleting a file or to malicious attack from viruses or hacking.
- This can be guarded against by doing regular back-ups of data (although this won't allow recovery following virus attack); also use of passwords and user ids to protect the data (see above for virus and hacking safeguards).

Hardware faults (e.g. head crash)

These can be guarded against by regular back-ups of data, use of UPS to prevent 'power glitches/loss' and use of parallel hardware.

Software faults (e.g. two pieces of incompatible software)

These can be guarded against by regular back-ups of data and the saving of data every 30 mins, for example, in case the computer becomes unresponsive.

Firewalls

- Examines traffic between user's computer and a public network.
- Checks whether incoming/outgoing data meets certain criteria.
- If data fails criteria, the firewall blocks the traffic and issues a warning.
- Logs all incoming/outgoing traffic.
- Criteria can be set to prevent access to certain websites; this can be done by the firewall keeping a list of all undesirable IP addresses.
- Firewalls CAN help to prevent viruses or hackers entering a user's computer.
- Warns the user if some software on their system tries to access an external data source (e.g. automatic software upgrade).

Firewalls cannot prevent certain harmful traffic if an individual by-passes the firewall or if employees are careless and divulge passwords etc.

● Common errors



- Many students confuse phishing and pharming and assume phishing scams automatically link a user to a fake/bogus website.
- Many students believe backing up data guards against viruses; this is not the case, since the backed-up data may also be infected.
- Students often say 'viruses damage the computer' with no mention of the nature of the damage or how it can be caused.

● Sample question



1 Explain why the following four statements about security are incorrect.

- 'by backing up data, a user can guard against the effects of a virus'
- 'phishing automatically redirects a user to a fake/bogus website'
- 'cookies contain key data such as passwords to log onto a website'
- 'encryption of data is a powerful way to prevent hacking of data during transmission'

Student answer

- The backed-up files may also be infected; therefore when using the backed-up files to restore an infected computer, the virus could simply be reintroduced.*
- Phishing requires the user to first open the email and then click on a link in the email or attachment.*
- Cookies are anonymous user profiles – they don't contain personal data.*
- Encryption doesn't prevent hacking; it simply makes the 'hacked' files impossible to read/understand.*

Examiner's comments

The answers given address the question very well. These types of question can be answered alternatively by addressing the issue; for example, 'to guard against a virus, the only certain ways are to run anti-virus software' or 'scan data coming from unknown sources'. These are regarded as valid ways of answering the question, provided the responses are still valid.

● Sample question



2 Describe three of the functions of a firewall and, in each case, describe an example of what the function is designed to do.

Student answer

- *Examines incoming and outgoing data/traffic; this is done to check if the data meets certain criteria, e.g. undesirable websites.*
- *Logs all incoming and outgoing data/traffic; this is done to allow the user or network manager to interrogate the log at a later date.*
- *Gives warnings if software is trying to access an external link; this could happen during automatic software updates and the user/manager has the option of allowing access or not.*

Examiner's comments

This is a difficult question to answer properly since many of the examples are stated as functions of firewalls in many textbooks.

It is important to come up with three different examples to gain full marks.


Examiner's tips

Be careful with questions which ask you to explain why something is incorrect. Read the statements very carefully and pick out which part (or parts) of the statement is/are incorrect.

● Security protocols



Secure Sockets Layer (SSL)

- Type of protocol (set of rules) used by computers to communicate with each other across a network.
- Allows data to be sent and received securely across a network.
- When a user logs on, SSL encrypts the data.
- User knows if SSL is being applied when they see `https` or the green padlock  sign in the status bar.
- When a user wants to access a website, the web browser asks the web server to identify itself; the web server sends a copy of the SSL certificate which the web browser authenticates; if this is OK then SSL-encrypted two-way data transfer begins between user's computer and web server.

Transport Layer Security (TLS)

- TLS is similar to SSL but is more recent and a more effective system.
- TLS is formed of a *record protocol* (contains data being transferred over the internet) and a *handshake protocol* (which permits website and client to authenticate each other and make use of encryption algorithms).
- Only recent web browsers support TLS.
- Makes use of *session caching* which improves the overall performance (here TLS can either start a new session each time a user accesses a website, or it can attempt to resume an earlier/existing session which improves system performance).

● Encryption



Symmetric encryption

- Utilises a secret key; when the key is applied, the *plain text* (original text) goes through an *encryption algorithm*, to produce *cypher text* (encrypted message).
- The recipient needs a key to then decrypt the message back into plain text.
- The main risk is that both sender and recipient need the same key which could be hacked or intercepted allowing a third party to decrypt the sent encrypted message.

Asymmetric encryption

- Asymmetric encryption uses *public keys* and *private keys*.
- Public key is available to everybody; private key is known only to computer user; both are needed to encrypt and decrypt messages.

- Encryption keys are generated by a *hashing algorithm* – this translates the message or key into a string of characters often shown in hex notation; an example is MD4 which generates a 128-bit string (the greater the number of bits, the harder it is to ‘crack’ the encryption).

● Authentication

- Used to verify that data comes from a trusted source.
- Makes use of passwords, digital signatures and/or biometrics.

● Denial of service attacks

- Denial of service is an attempt at preventing users from accessing part of a network, notably internet servers.
- Can prevent users from accessing their emails, accessing certain websites or accessing online services.
- This is achieved by the attacker flooding the network with useless traffic; for example, sending out thousands of requests to a website or sending out thousands of spam emails to users, ‘clogging it up’.
- Can be mitigated against by:
 - using an up-to-date virus checker
 - using a firewall to restrict traffic
 - using email filters
 - looking out for signs (e.g. slow network performance, increase in spam or inability to access certain websites).

● Ethics

- Ethics is a set of principles set out to regulate use of computers; three factors are considered:
 - intellectual property rights (e.g. copying software without permission)
 - privacy issues (e.g. hacking or any illegal access to a computer)
 - effect of computers on society (e.g. job losses, social impacts, and so on).
- A code of ethics is published by the ACM and IEEE (see the Student’s Book for a list of the ten codes).

Free software

Users have the freedom to run, copy, change or adopt *free software*. There are rules, however, that need to be obeyed:

- Cannot add source code except from software which is also free software.
- Cannot produce any software which copies existing software that is subject to copyright laws.
- Cannot alter the source code so that it infringes any copyright that protects other software.
- May not produce software that is possibly offensive.

Freeware

Software that a user can download free of charge. There are no fees associated with the software (e.g. Adobe or Skype) but they are subject to copyright laws and the user is often asked to tick a box to say they understand and agree to the terms and conditions governing the software.

Shareware

Users are allowed to try out shareware free for a trial period. At the end of this period, the user will be requested to pay a fee. Very often, the trial version doesn't have all the features of the full version – all the features become available once the fee is paid.

Common errors

- Many students don't realise loss of data doesn't have to be a malicious act, it can be due to an error or not following correct procedure.
- Students often claim that firewalls ALWAYS protect against viruses and hackers – this is not the case.
- Students often confuse free software, freeware and shareware.



Sample question

- 3 Put the following statements into the correct order which describe what happens when a user wants to access a secure website and receive or send data.



Order	Statements
	web server responds by sending a copy of the SSL certificate to the user's web browser
	web browser requests that the web server identifies itself
	once message received, the web server acknowledges the web browser, and the SSL encrypted 2-way data transfer begins
	user's web browser sends a message so that it can connect with the required website which is secured by SSL
	if the web browser can authenticate this certificate, it sends a message back to the web server to allow communication to begin

Student answer

3
2
5
1
4

Examiner's comments

The student has correctly identified the five steps in the process.

Examiner's tips

In questions like this, it is usually a good idea to:

- first of all decide the first and last stage (sometimes this is given to you)
- then using rough paper, set out each of the steps in the correct order
- then check whether your steps are logical (e.g. if you put a step like 'fill up the printer paper tray' before the step 'printer sends an error message that it is out of paper' then this is clearly logically incorrect – it is usually common sense).

● Sample question



- 4 Describe denial of service attacks and describe **three** ways to possibly guard against such attacks.

Student answer

DoS:

- An attempt at preventing users from accessing part of a network; it can also affect an individual's computer.
- It can prevent the user from accessing their emails, accessing websites or online services.

Three ways:

- Use an up-to-date virus checker.
- Set up and always use a firewall to restrict traffic to and from an internet server and a user's computer.
- Apply email filters to manage and filter out unwanted traffic or spam emails.

Examiner's comments

The answer given is well structured which is essential for a 'describe' question. It is always necessary to match up your number of points with the marks available for the question. A well-explained (additional points) answer could gain more than one mark per point, however there is no way of knowing this, so following the advice about one mark per described point is a safe way to tackle the question.

Exam-style questions

- 1 Indicate which of the following statements are clear examples of when computer ethics has or has not been followed. [5 marks]

Statements	Ethics followed (✓)	Ethics NOT followed (✗)
a technician works for a software company during the day but also does freelance computer advice in the evenings		
a software engineer develops software for a security company; he uses some of the algorithms when developing computer games for a different company		
a software engineer writes software for company A but also accepts money from company B to include routines in the software which covertly advertises company B		
a company employs people in different countries to write software modules; these modules are put together to produce software packages		
a software engineer undergoes training by a different company to improve his software writing skills		

2 In the diagram, seven descriptions are shown on the left and ten computer terms are shown on the right. By drawing arrows, connect each description to the correct term – each term may be used once or not at all. [7 marks]

users are allowed to try out software free of charge for a trial period; the trial software is often missing certain features	free software
software that can be downloaded from the internet free of charge; the software is subject to copyright laws	cookies
set of principles set out to regulate the use of computers, e.g. intellectual property rights	computer ethics
a protocol used by computers on a network to communicate with each other safely	shareware
users have the freedom to run, copy, change or adapt this software; it is based on liberty and not price issues	firewall
a person takes another person's idea or work and then claims it as their own personal work	plagiarism
attempt at preventing users from accessing an internet server by sending out thousands of requests	encryption
	freeware
	denial of service attack
	secure sockets layer (SSL)

3 Name **three** security issues which can affect a user's computer. In each case, describe the possible effect of the security issue and suggest a way of minimising or removing the risk. [9 marks]

4 a Give **three** ways data can be lost on a computer. [3 marks]

b In each case, give a way to safeguard against the risk or to recover any lost data. [3 marks]

5 a Explain the function of a firewall. [2 marks]

b Describe **five** of the tasks carried out by a firewall. [5 marks]

6 Explain each of the following terms.

- a symmetric encryption
- b asymmetric encryption
- c hashing algorithm (used in encryption)
- d cypher text
- e encryption algorithm
- f key distribution problem

[6 marks]

It is illegal to photocopy this page

9 Problem-solving and design

Key objectives

Objectives of this chapter are to revise:

- computer systems and sub-systems:
 - top-down design
 - structure diagrams
 - library routines
 - sub-routines
- test data
- validation and verification
- using trace tables
- identifying and correcting errors
- the effectiveness of a solution

Key definitions

Term	Definition
Computer system	Made up of software, data, hardware, communications and people
Test data	A set of test data is all the items of data required to work through a solution
Validation	Validation is the automated checking by a program that data is reasonable before it is accepted into a computer system
Verification	Verification is checking that data has been accurately copied onto the computer or transferred from one part of a computer system to another
Trace table	A trace table can be used to record the results from each step in an algorithm; it is used to record the value of an item (variable) each time it changes

Computer systems and sub-systems

A computer system is made up of software, data, hardware, communications and people; each computer system can be divided up into a set of sub-systems. Each sub-system can be further divided into sub-systems and so on until each sub-system just performs a single action.

Top-down design

Top-down design is the breaking down of a computer system into a set of sub-systems, then breaking each sub-system down into a set of smaller sub-systems, until each sub-system just performs a single action.

Structure diagrams

In order to show top-down design in a diagrammatic form, structure diagrams can be used. The structure diagram shows the design of a computer system in a hierarchical way, with each level giving a more detailed breakdown of the system into sub-systems.

Library routines

A library routine is a set of programming instructions for a given task that is already available for use. It is pre-tested and usually performs a task that is frequently required.

Sub-routines

A sub-routine is a set of programming instructions for a given task that forms a sub-system not the whole system. Sub-routines written in high-level programming languages are called procedures or functions depending upon how they are used. See Chapter 11 of the Student's Book for further details.

● Common errors □

- Students often confuse a structure diagram with a flowchart.
- Students forget to include communications and people as part of a computer system.

● Sample question □

1 Explain what is meant by the following terms.

- computer system
- top-down design
- library routine

Examiner's tips

Read the question carefully, does it ask for an explanation, a description or a consequence?

Student answer

- Programs and a computer.*
 - Putting off coding until everything is specified.*
 - A sub-routine to perform a given task, e.g. DLL.*

Examiner's comments

In part a, the answer is true but doesn't quite answer the question, which asked for an explanation not a list of components. A better answer would be 'Computer systems include a computer with software, data, peripheral and communication devices that are necessary to make the computer system work.'

In part b, the answer given could be a result of using top-down design but it is not an explanation. A better answer would refer to stepwise refinement explaining how a whole system is broken down into sub-systems.

In part c, the answer given is accurate; a better answer would include an explanation that the sub-routine is readily available.

Examiner's tips

'Explain' answers will usually need at least two or three lines.

● Test data □

A set of test data is all the items of data required to work through a solution.

Normal data

Solutions need to be tested to prove that they do what they are supposed to do. A set of test data should be used together with the result(s) that are expected from that data. The type of test data used to do this is called normal data; this should be used to work through the solution to find the actual result(s) and see if these are the same as the expected result(s).

Erroneous/abnormal data

Solutions also need to be tested to prove that they do not do what they are not supposed to do. Test data should be used that will be rejected as the values are not suitable. This type of test data is called erroneous or abnormal test data; it should be rejected by the solution.

Extreme data

When testing algorithms with numerical values sometimes only a given range of values should be allowed. Extreme data are the largest and smallest values that normal data can take.

Boundary data

There is another type of test data called boundary data; this is used to establish where the largest and smallest values occur. At each boundary two values are required; one value is accepted and the other value is rejected.

Data types

Normal data	Abnormal/ erroneous	Extreme	Boundary
Accepted	Rejected	Accepted	Two values: one accepted, one rejected
No error message	Error message	No error message	No error message if accepted, error message if rejected
Tests that the solution works as it should.	Values not used or stored.	Tests that the solution works as it should with highest or lowest values.	Tests that the solution works as it should with highest and lowest values accepted and the lowest and highest values not used or stored.

See the Student's Book for examples of test data.

Common errors

- Students often say: 'extreme data is large values that are rejected'; this is incorrect since all extreme data should be accepted.
- Statements such as 'all test data was accepted' or 'all test data worked' are never worth any marks since they don't say what type of test data was used; it is necessary to quantify the type of test data and state whether it should be accepted or rejected.

Sample question

2 A program has been written to store the number of students in each class. There are 6 classes and each class can have between 2 and 30 students. Here are 3 examples of test data. For each example identify the type of test data and the expected outcome.

- a 24 b 67 c 30

Student answer

- 2 a Normal data – this is accepted.
b Abnormal data – this is rejected.
c Boundary data – this is accepted.

Examiner's comments

In part a, the student has given the correct data and stated that it is accepted, this is true but the information in the question about the value being stored was not used. A better answer would read 'This is normal data; the value is accepted and stored for future use.'

In part b, the student has answered correctly.

In part c, the answer is insufficient as only one side of the boundary is being tested; it would be better to write 'Extreme data that is the largest value to be accepted and stored.'

Examiner's tips

Always remember to use the information given to you in the question.

When there are two possible options, consider each one carefully and choose the one that best matches the information given in the question.

● Validation and verification



For data entry, validation is performed automatically by the computer system to ensure that only data that is reasonable is accepted and verification is used to check that the data does not change as it is being entered.

Validation

Validation is the automated checking by a program that data is reasonable before it is accepted into a computer system. Different types of check may be used on the same piece of data.

Verification

Verification is checking that data has been accurately copied onto the computer or transferred from one part of a computer system to another.

Summary

Validation	Verification
range check length check type check character check format check presence check check digits	double entry (data entry) screen/visual check (data entry) parity check (data transmission) checksum (data transmission)

See the Student's Book for examples and descriptions of each type of check.

● Common errors



- Students often give proofreading as an example of verification; this is incorrect since proofreading is checking for errors such as spelling, grammar and incorrect facts whereas verification on data entry is checking that the data entered into a computer is the same as the data written down.
- Students often confuse checksums and checkdigits. Checksums are used for data transfer and check digits are used for data entry.

● Sample question



- 3 Explain, using examples, the differences between verification and validation when data is entered into a computer.

Student answer

Validation is checking that the data entered is correct, for example an age of 160 should not be accepted. Verification is checking that no errors are made during data entry, for example typing in a password twice.

Examiner's tips

When the word 'difference' is used, ensure that differences are identified.

Give examples that reinforce the answer given, like using the same scenario to highlight what the different types of check can do.

Examiner's comments

The question has asked for an explanation of the differences, but the answer given does not consider the difference between validation and verification. The answer is an attempt to give the meaning of each term separately.

Two differences should be mentioned in the answer, for example 'Verification can be an automatic check completed by a computer, for example typing in a password twice, or done by a human being, for example checking that the name they typed is correct. Validation is always an automatic check completed by a computer, for example checking for an unacceptable age. Validation checks that data is reasonable, for example that an age cannot be over 125, whereas verification checks that the data has been copied correctly, for example an age of 25 has not been entered when the age on the input document was 52.'

Other points to note: validation checks that data is reasonable not correct, verification checks that the data is the same as that provided.

● Tracing algorithms

A thorough structured approach is required to find out the purpose of an algorithm; this involves recording and studying the results from each step in the algorithm. This will require the use of test data and trace tables. A trace table can be used to record the results from each step in an algorithm; it is used to record the value of an item (variable) each time it changes. This manual exercise is called a dry run. A trace table is set up with a column for each variable and a column for any output.

Trace tables

A trace table can be used to record the results from each step in an algorithm; it is used to record the value of an item (variable) each time it changes.

Dry run

This manual exercise is called a dry run. A trace table is set up with a column for each variable and a column for any output.

A	B	C	X	Output
0	0	100		

Effectiveness of a solution

An effective solution is the simplest solution that works for all sets of data and contains no unnecessary processes or repeated actions.

See the Student's Book for an example of a completed trace table.

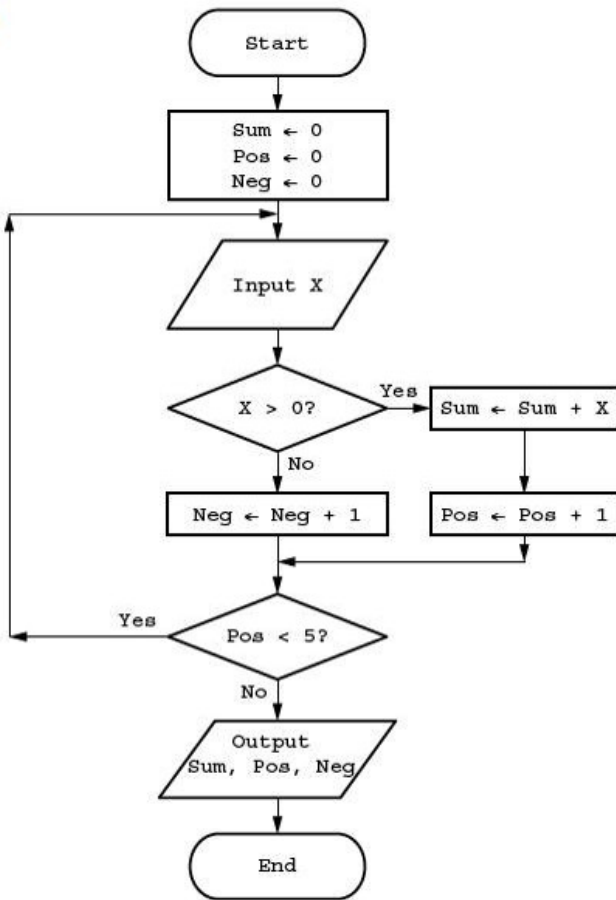
● Common errors

- Forgetting to initialise values such as zero.
- Including working out not stored in a variable.
- Repeating output values.



● Sample question

4



Test data 5, -1, 4, 7, -3, 2, 1

Use this test data and a trace table to dry run this algorithm.

Student answer

Sum	Pos	Neg	X	Output
5	1	1	5	
9	2	2	-1	
15	3		4	
17	4		7	
18	5		-3	18 5 2
			2	
			1	

It is illegal to photocopy this page

Examiner's comments

The initialisation has not been completed, there is an error in the addition, the trace table is not easy to check as each iteration is not recorded on a separate row.

Sum	Pos	Neg	X	Output
0	0	0		
5	1		5	
		1	-1	
9	2		4	
16	3		7	
		2	-3	
18	4		2	
19	5		1	
				19 5 2

Each iteration is set out on a separate row for ease of checking.

Examiner's tips

Set out each iteration of a loop on a separate row.

Always double check arithmetic and the format of output.

Sample question

5 Identify the error, and suggest a correction, in the algorithm for question 1.

Student answer

Zero is counted as a negative number; the algorithm should ignore zero.

Examiner's comments

The answer identifies the problem; the solution given needs to explain how the algorithm should be corrected. For example 'The algorithm should test for zero and in this case not update, Neg, Pos or Sum.'

Exam-style questions

- 1 List the **five** components of a computer system. [5 marks]
- 2 Explain what is meant by a sub-system; use a structure diagram to illustrate your answer. [4 marks]
- 3 Describe **three** different types of test data and use examples to explain what each type is used for. [9 marks]
- 4 **a** Give **two** examples of validation. [2 marks]
- b** Give **two** examples of verification. [2 marks]

- 5 The following items of data are to be input using a computer-based form.

Name

Address

Date of birth

Shoe size

Password

For each item state, with a reason, whether validation or verification should be used. Give an example of a method of validation or verification that would be suitable.

[10 marks]

- 6 The following algorithm should add up six negative numbers.

```
Counter ← 1
```

```
FOR Counter ← 1 TO 6
```

```
  REPEAT
```

```
    PRINT "Enter a negative whole number"
```

```
    INPUT Number
```

```
  UNTIL Number >= 0
```

```
  Total ← Total + Counter
```

```
  Counter ← Counter + 1
```

```
OUTPUT Total
```

```
NEXT
```

- a Identify all the errors in the algorithm. [5 marks]

- b Re-write the algorithm without the errors. [5 marks]

- c Set up a trace table and test your algorithm with the following test data:

-4, -3, 1, -2, 0, -5, -5, 27, -1 [3 marks]

- d Explain why your algorithm was more effective than the original. [2 marks]
-

10 Pseudocode and flowcharts

Key objectives

Objectives of this chapter are to revise:

- pseudocode:
 - assignment
 - loop structures
 - input and output
 - standard actions
- flowcharts:
 - symbols

Key definitions

Term	Definition
Pseudocode	Describes what an algorithm does by using English key words that are very similar to those used in a high-level programming language
Flowchart	Describes what an algorithm does by using a standard set of symbols joined by lines showing the direction of flow

Pseudocode

Using pseudocode is a clear and concise way to represent an algorithm. It describes what the algorithm does by using English key words that are very similar to those used in a high-level programming language. Data items to be processed by the algorithm are given meaningful names in the same way that variables and constants are in a high-level programming language. Pseudocode is not bound by the strict syntax rules of a programming language. It does what its name says, pretends to be programming code!

Assignment

Values are assigned to an item/variable using the \leftarrow operator. The variable on the left of the \leftarrow is assigned the value of the expression on the right.

Conditional statements

When different actions are performed by an algorithm according to the values of the variables, conditional statements can be used to decide which action should be taken. There are two types of conditional statement:

- a condition that can be true or false:
`IF ... THEN ... ELSE ... ENDIF`
- a choice between several different values:
`CASE ... OF ... OTHERWISE ... ENDCASE`

Loop structures

When some actions performed as part of an algorithm need repeating this is called iteration. Loop structures are used to perform the iteration.

There are three different types of loop structure:

- A set number of repetitions: `FOR ... TO ... NEXT`
- A repetition, where the number of repeats is not known, that is completed at least once: `REPEAT ... UNTIL`
- A repetition, where the number of repeats is not known, that may never be completed: `WHILE ... DO ... ENDWHILE`

Input and output statements

INPUT and OUTPUT are used for the entry of data and display of information. Sometimes READ can be used instead of INPUT; this is usually used for reading from files, which is not covered in this book. Frequently PRINT is used instead of OUTPUT.

Standard actions

The ability to repeat actions is very important in the design of algorithms. When an algorithm is turned into a program the same set of actions may be repeated many thousands of times, for example to keep a running total.

```
RunningTotal ← RunningTotal + Value
```

Keeping a count of the number of times an action is performed is another standard action.

```
Count ← Count + 1
```

Counting is also used to count down until a certain value is reached.

```
NumberInStock ← NumberInStock - 1
```

See the Student's Book for examples of each type of pseudocode statement.

● Common errors

- Students often use = instead of ←. = is a comparison that stands for 'is the value of the expression on the left equal to the value of the expression on the right?'. ← stands for 'assign the value of the expression on the right to the variable on the left'.
- → is not used in pseudocode.
- An IF statement is not a loop.



● Sample question

- 1 Write an algorithm in pseudocode to input 100 integers and store the total in a variable Sum.



Student answer

```
FOR Counter ← 1 TO 100
  INPUT X
  SUM ← Sum + X
```

Examiner's comments

There are a number of problems with this pseudocode:

- SUM and Sum are not the same variable.
- Sum should be initialised to zero before the loop.
- There is no NEXT; some programming languages, e.g. Python do not require NEXT, only indentation, but pseudocode does require a NEXT.

● Sample question



2 Rewrite the algorithm below using a REPEAT ... UNTIL loop.

```
FOR count ← 1 TO 100
  INPUT X
  A[count] ← X
NEXT
```

Student answer

```
REPEAT
  INPUT X
  A[count] ← X
  count ← count + 1
UNTIL count = 100
```

Examiner's tips

There are often many different correct pseudocode answers; if you are unsure do a quick dry run with fewer values, e.g. 5 instead of 100.

Examiner's comments

There are a number of problems with this pseudocode:

- count should be initialised to 1 before the loop, as the first value is stored in A [1] not A [0]
- Only 99 values will be stored as the count is incremented after the value is stored; the UNTIL statement should be UNTIL count > 100.

Examiner's tips

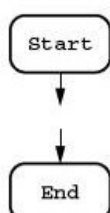
Take care with loop counters. If you use a FOR ... TO ... NEXT loop, the loop counter is automatically initialised and incremented, if you use REPEAT ... UNTIL or WHILE ... DO ... ENDWHILE you have to manage the initialisation, incrementation and condition testing.

● Flowchart symbols

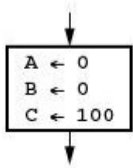


Flowcharts are drawn using standard symbols.

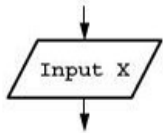
Start/End



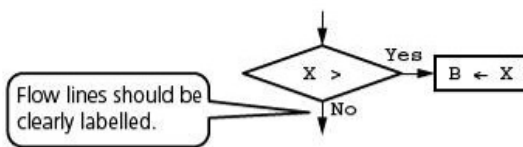
Process



Input/Output



Decision



See the Student's Book for an example of a flowchart.

● Common errors



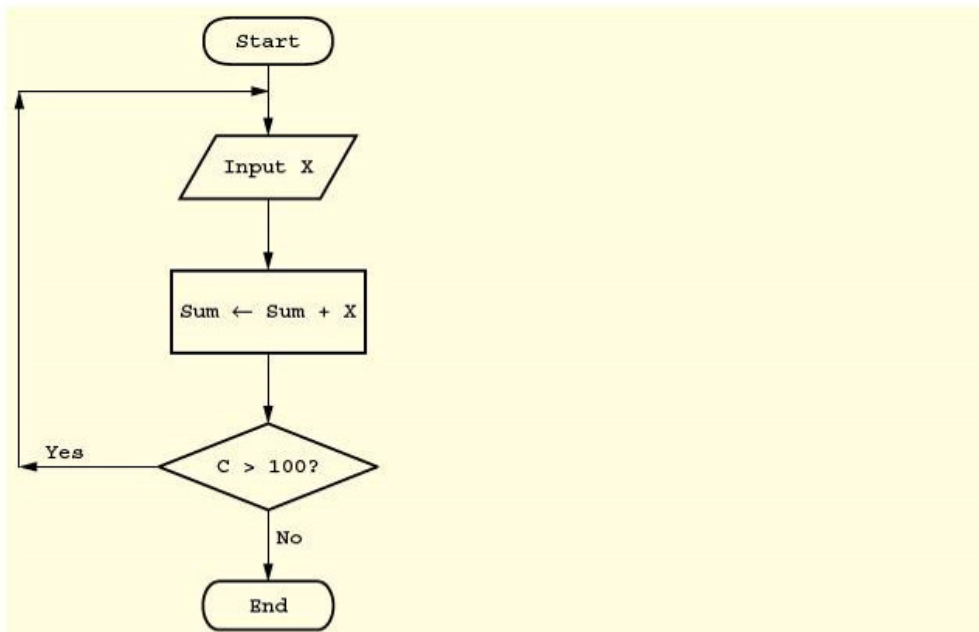
- Students often draw flow lines without arrows.
- Students often draw decision boxes where flow lines are not clearly labelled.
- Program flowcharts should have only one begin and one end.

● Sample question



3 Write an algorithm using a flowchart to input 100 integers and store the total in a variable Sum.

Student answer



It is illegal to photocopy this page

Examiner's comments

There are a number of problems with this flowchart:

- C should be initialised to 1 before the loop
- Sum should be initialised to zero before the loop
- C needs to be incremented as well as tested
- the flow lines from the decision box are incorrectly labelled.

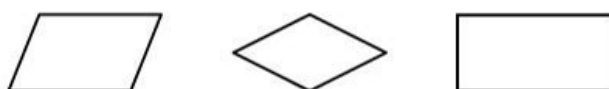
Examiner's tips

Always remember to manage any initialisation and counting when using a flow chart and check labelling of flow lines.

Use a flowchart template and a ruler in an examination. This will make your flowchart quicker to draw, neater and easier to understand for the examiner.

Exam-style questions

- 1 Describe **three** types of loop statement that can be used in pseudocode. [6 marks]
- 2 Describe, using examples, **two** types of selection statement that can be used in pseudocode. [6 marks]
- 3 Using pseudocode, write an algorithm to input a number and reject any values less than 1 or more than 10. [3 marks]
- 4 Explain, using pseudocode examples, the difference between selection and iteration. [4 marks]
- 5 State the correct use for each of the following flowchart symbols. [3 marks]



- 6 Using a flowchart, write an algorithm to input 10 whole numbers and output the largest number. [6 marks]

11

Programming concepts

Key objectives

Objectives of this chapter are to revise:

- programming
- declaration and use of:
 - variables
 - constants
- basic data types:
 - integer
 - real
 - char
 - string
 - Boolean
- how to use
 - sequence
 - selection
 - repetition
 - totalling
 - counting

Key definitions



Term	Definition
Variable	A named data store that contains a value that may change during the execution of a program
Constant	A named data store that contains a value that does not change during the execution of a program
Sequence	Statements are followed in sequence so the order of the statements in a program is important

Basic data types



In order for a computer system to process and store data effectively, different kinds of data are formally given different types.

Integer

An integer is a positive or negative whole number that can be used with mathematical operators.

Real

A real number is a positive or negative number with a fractional part. Real numbers can be used with mathematical operators.

Char

A variable or constant of type char is a single character.

String

A variable or constant of type string is several characters in length. Strings vary in length and may even have no characters, an empty string; the characters can be letters and/or digits and/or any other printable symbol.

Boolean

A Boolean variable can have only two values, TRUE or FALSE. See the Student's Book for examples of each data type.

Common errors



- Variable names and constant names must not contain spaces or punctuation marks.
- Programs written with variable and constant names of just one or two letters are not easily understandable.

● Sample question



- 1 You have been asked to write a program to input the weights of 10 people and find out the average weight and how many people weighed over 80 kilograms.
- Name two variables that you will need to use. For each one, give the data type and explain what it will be used for.
 - Name one constant that you could use. State the data type and explain what it will be used for.

Student answer

- 1 a First variable W, type integer, weight of a person. Second variable Average, type integer, average weight of all 10 people.
b Constant eighty, integer, value to check against weights input.

Examiner's comments

This is a good attempt that addresses every point in the question. Improvements could be made as follows:

- The first variable name is not meaningful, Weight would be a better answer.
- For the second variable, as the result may not be a whole number a data type of real is needed.
- The answer to part b is good; for consistency all the names should begin with either a small letter or a capital letter.

● Sample question



- 2 Name three data types and provide an example of each one.

Student answer

Real 3.4
Integer 3
Boolean True
Char M

Examiner's comments

An excellent answer in content but poor examination technique; always give the same number of answers as requested in the question because examiners will not pick out your best answers from a list.

Examiner's tips

Always use meaningful names for variables and constants.

Never include spaces or punctuation in names for variables and constants.

When asked for a set number of answers in a question, do not include extra answers.

● Programming techniques □

Sequence

Statements are followed in sequence so the order of the statements in a program is important.

Selection

Selection is a very useful technique, allowing data items to be picked out according to given criteria, for example selecting the largest value or the smallest value, selecting items over a certain price, selecting everyone who is male.

Repetition

Most programming languages support three types of repetition:

- a fixed number of repetitions
- an unknown number of repetitions with at least one repetition, as the condition is tested at the end of the loop
- an unknown number of repetitions which may not be completed at all, as the condition is tested at the beginning of the loop.

Totalling

Totalling is used with repetition with the total updated every time the loop is repeated. Keeping a running total is one of the most frequently used programming techniques in many computer systems.

Counting

Counting is used with repetition with the counter increased by 1 every time the loop is repeated. Counting items or events is another of the most frequently used programming techniques.

See the Student's Book for examples of all these programming techniques.

● Common errors □

- Students often use variables for counting and totalling that haven't been initialised to zero before the start of the loop.
- Students often set the loop counter inside a FOR loop to another value within the loop. This means that the number of repetitions will be wrong.

● Sample question □

- 3 Identify the errors in the following program code to print the average of 10 numbers input, and then rewrite the program correctly.

```

10 Total = 1
20 Count = 0
30 WHILE Count <= 10
40   INPUT Number
50   Total = Total - number
60   Count = Count + 1
70   PRINT Total/Count
80 ENDWHILE

```

Student answer

```

10 Total = 0
20 Count = 0
30 WHILE Count < 10
40   INPUT Number
50   Total = Total + number
60   Count = Count + 1
70 ENDWHILE
80 PRINT Total/10

```

Examiner's comments

The student has rewritten the program correctly but hasn't explicitly identified the errors. This would need to be done for full marks. For example, in line 10 the total is incorrectly initialised, in line 30 the test should be for less than 10 to stop the loop repeating 11 times, in line 50 the total is decreased rather than increased and in line 70 an attempt at an average will be printed every time the loop is repeated.

Examiner's tips

Always make sure all tasks are completed.

The programming statements on an examination paper will not be from any particular programming language.

The programming statements usually have line numbers to make it easier to identify errors.

Exam-style questions

- 1 Explain, using examples, the difference between a variable and a constant when a program is being executed. [2 marks]
- 2 Write a program that checks the number of items left on the shelf against a reorder level. You can input a reorder level at the start of the program and a value of -1 for the number of items should finish the program. [6 marks]
- 3 Identify the errors in the following program code to print the highest value of 15 numbers input, and then rewrite the program correctly.

```

10 Highest = 1000000
20 Count = 1
30 WHILE Count < 15
40   INPUT Number
50   IF Highest > Number THEN Highest = Number
60   Count = Count - 1
70   PRINT Number
80 ENDWHILE

```

[8 marks]

- 4 Explain, using program code examples, the difference between counting and totalling. [4 marks]
 - 5 Give **two** different sets of test data that you could use to test the program that you wrote for question 2. Explain why you chose each set of data. [6 marks]
-

12

Data structures: arrays and using pre-release material

Key objectives

Objectives of this chapter are to revise:

- declaration of one-dimensional arrays
- the use of one-dimensional arrays
- the use of a variable as an index in an array
- reading and writing values in an array using a FOR ... TO ... NEXT loop
- using pre-release material

● Key definitions

Term	Definition
One-dimensional array	A list where each item can be identified by an index number
Element	An item in a list
Index	The number of each element within a one-dimensional array

● Arrays

One-dimensional array

Lists in programming are identified by the use of a single name and each item in the list can be found by an index number showing the place in the list. This type of list is called a one-dimensional array.

Element

The items in the list are the elements of the array.

Index

In order to repeat the same task many times for all the items in a list, each item needs to be referred to in the same way using the index number for each element of the one-dimensional array. If a variable is used for this index number, then the same code can be re-used for each item.

Read or write values in an array using a FOR ... TO ... NEXT loop

A FOR ... TO ... NEXT loop uses a fixed number of repeats so it is ideal to use with an array, when the number of elements is known, as the loop counter can be used as the array index.

● Common errors

- Not knowing whether an array index can start at 0 or 1, this depends on the programming language used.
- Not giving arrays and their indexes meaningful names.

● Sample question

- 1 a An array `Pupil [1:30]` stores the number of times each pupil has been late. Write a program to find the index number of the pupil who has been late the most.
- b Identify a problem that could occur with finding the pupil.

Student answer

1 a

```
import array
ConstNoPupils = int(30)
MostLates = 0
PupilIndex = 0
Pupil = array.array("i", range (ConstNoPupils + 1))
Counter = int(0)
for Counter in range (1, ConstNoPupils + 1):
    if MostLates < Pupil[Counter]:
        MostLates = Pupil[Counter]
        PupilIndex = Counter
print ("Pupil ", PupilIndex, " has the most number of lates ", MostLates)
```

b No problem – this program works

Examiner's comments

The answer to part a does work in Python.

The answer to part b needed more consideration as this method will only pick out the first pupil if more than one pupil has been late the most times.

Examiner's tips

Always use the array index range given in the question, for example 1 to 30 rather than 0 to 29.

If a question asks for the identification of a problem, there will be one.

When providing programming answers in Python, take great care with the indentation as this shows the beginning and end of a repetition in a program.

● Common errors



- Not answering the question as set on the examination paper.
- Only providing program code when a description or an explanation is asked for.
- Providing an answer for the whole of the task when the question asks about a specific part of that task.

● Sample mini pre-release



You have been asked to write a program to record the height in metres and the shoe size of each member of your class. These values are to be stored in arrays then each array is to be searched to find the pupil(s) with the smallest shoe size, the tallest pupil(s) and the number of pupils that are over 1.5 metres tall.

- 1 Name the arrays you will need, the type of data to be stored in each array and what each array will be used for.

Student answer

Height is the name of one array and the data type is real, ShoeSize is the name of the other array and the data type is integer.

Examiner's comments

The pupil has given meaningful names to the two arrays and identified appropriate data types. There is no indication of what each array will be used for apart from the meaningful names. A better answer would have continued 'The array Height [] will store the heights of all the pupils in the class and the array ShoeSize [] will store the shoe sizes of all the pupils in the class.'

- 2 Describe validation checks that you performed when a pupil's height was input.

Student answer

```
if Height > 2.0:
    print "too tall"
```

Examiner's comments

The pupil has written program code, not the description asked for and there is only one validation check. A better answer would be 'the program performs a range check on the pupil's height using a low value, e.g. 0.5, for the lower bound and a high value, e.g. 2.0, for the upper bound'.

Examiner's tip

Always make sure you have answered the question that was set, giving a description if requested and giving the number of answers required.

Never include items that are not required in the answer, e.g. mentioning a validation check on another input.

Read your answer through when you have written it to spot any errors.

Exam-style questions

- 1 Identify **one** constant you could use in your program, state its data type and what you used it for. [3 marks]
- 2 Identify **two** variables that you used in your program. For each one state its data type and what you used it for. [6 marks]
- 3 Explain how you identified the number of pupils over 1.5 metres tall. You may use programming statements as part of your explanation. [4 marks]
- 4 Explain how you could extend your program to find the largest shoe size and the average shoe size. [4 marks]

13 Databases

Key objectives

Objectives of this chapter are to revise:

- single-table database structure
- using query-by-example

● Key definitions



Term	Definition
Database	A structured of collection of data
Table	Contains data about one type of item or person or event, using rows and columns
Record	A row within a table that contains data about a single item, person or event
Field	A column which contains one specific piece of information
Primary key	A unique reference for each record

● Database



A database is a structured collection of data that allows people to extract information in a way that meets their needs.

Table

Tables contain data about one type of item or person or event, using rows and columns. There are a set number of columns and a variable number of rows.

Record

Each record, or row, within a table contains data about a single item, person or event.

Field

Each field, or column, contains one specific piece of information about a single item, person or event. Fields are given a data type, for example, number, text, yes/no, currency and date/time.

Primary key

In order to be sure that each record can be found easily and to prevent more than one copy of the same record being kept, each record includes a primary key field. Each primary key field in a table is unique.

Summary

Record 1	Primary key	Field 1	Field 2	Field 3	Field 4
Record 2	Primary key	Field 1	Field 2	Field 3	Field 4
Record 3	Primary key	Field 1	Field 2	Field 3	Field 4
Record 4	Primary key	Field 1	Field 2	Field 3	Field 4
Record 5	Primary key	Field 1	Field 2	Field 3	Field 4
Record 6	Primary key	Field 1	Field 2	Field 3	Field 4

See the Student's Book for examples of the use of databases.

● Common errors

- Not knowing the difference between records (rows) and fields (columns).
- Choosing a field for the primary key that may contain duplicates; 'name' is a common incorrect answer.



● Sample question

1 A single-table database, STUDENT, is set up to store for each student the name, registration number, date of birth, gender and address.



- Identify the fields needed; for each field give an appropriate data type.
- Select, giving a reason, the field you would choose as the primary key.

Student answer

1 a Name – text
 Registration number – number
 Date of birth – date
 Gender – char
 Address – text
 b Registration number, as it uniquely identifies each student.

Examiner's comments

The answer to part a will work apart from gender; this should have a data type of text since char isn't a separate data type for databases unlike programming languages. Also many Registration numbers are of a fixed length, for example 6 characters, so it would be better to use a text field.

The answer to part b is correct.

Examiner's tips

Always use the data type of text for any numeric fields that will not be used for calculations.

Use database data types of text and number, not programming types of string, char, integer and real.

● Query by example grid

Field:				
Table:				
Sort:				
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				



● Common errors

- Not completing the Table row for every column, it will always be the same for a single-table database.
- Ticking the Show box when the field is not going to be displayed.
- Not checking that each field's name matches that in the database exactly.





● Sample question

2 A single-table database, STUDENT, is set up to store for each student the name, registration number, date of birth, gender and address.

Display the names in alphabetical order and the registration number of the male students.

Student answer

Field:	Name	Registration number	Gender	
Table:	STUDENT	STUDENT	STUDENT	
Sort:	Ascending			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:			= 'M'	
or:				

Examiner's comments

The student has correctly identified the fields to use and the order to sort the results. An extra field, Gender, has been incorrectly displayed.

● Sample question



3 A single-table database, BABY, is set up to store for each baby the name, hospital number, date of birth, gender and weight in kilograms. Part of the database is shown here.

Name	Hosp no	DOB	Gender	Weight
Alice	HN5677	05/09/2015	F	2.8
Banan	HN5621	06/10/2015	F	2.9
Tariq	HN5701	08/10/2015	M	3.5
Karen	HN5689	12/09/2015	F	3.1
Sonia	HN5691	09/09/2015	F	3.2
Samuel	HN5710	12/09/2015	M	4.1
Hassan	HN5702	11/10/2015	M	3.9

Show the results for the following query-by-example grid.

Field:	Name	Gender	Weight	
Table:	BABY	BABY	BABY	
Sort:	Ascending			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:			> 3.5	
or:				

Student answer

Tariq	M
Samuel	M
Hassan	M

Examiner's comments

The student has correctly identified the fields to show. The names have not been sorted in alphabetical order and Tariq's record should not have been included.

Examiner's tips

Always make sure you have only written the contents of the fields marked as 'show'.

Never include extra information that will not be shown.

Check any criteria very carefully.

Always show a list in the order selected; the list may not be ordered on the first field.

Exam-style questions

Answer these questions about the BABY database.

- 1 How many fields and records are there in the part of the database shown? [2 marks]
 - 2 State, giving a reason, which field should be used for the primary key. [2 marks]
 - 3 Give a data type and describe a validation check that would be appropriate for each field. Each of your validation checks must be different. [10 marks]
 - 4 Use a query-by-example grid to select all the female babies and show their names in ascending order of date of birth. [3 marks]
-

Chapter 1

1 a 6 C 7 E

b 1 0 1 0 0 1 0 0 1 1 1 1

c 1 7 7 0

2 a (i) pump A is off
pump B is on
cooler is off
heater is on

(ii) 1 0 1 0 1 0 0 1

(iii) 1 1 1 1 1 1 1 1 – this would be an impossible combination so could be used to indicate a fault/error condition

b (i) $800 \times 8 = 6400$ MB and $80 \times 400 = 32000$ MB

Total size = 38400 MB

(ii) $38400/1024 = 37.5$ GB

3 a (i) MAC address

(ii) memory dump

(iii) HTML

(iv) web address

b The UAA is by far the most common type of MAC address and this is the one set by the manufacturer at the factory. It is rare for a user to want to change this MAC address.

However, there are some occasions when a user or organisation wishes to change their MAC address. This is a relatively easy task to carry out but it will cause big problems if the changed address isn't unique.

There are a few reasons why the MAC address needs to be changed using LAA:

- Certain software used on mainframe systems needs all the MAC addresses of devices to fall into a strict format; because of this, it may be necessary to change the MAC address of some devices to ensure they follow the correct format.
- It may be necessary to bypass a MAC address filter on a router or a firewall; only MAC addresses with a certain format are allowed through, otherwise the devices will be blocked if their MAC address doesn't adhere to the correct format.
- To get past certain types of network restrictions it may be necessary to emulate unrestricted MAC addresses; hence it may require the MAC address to be changed on certain devices connected to the network.

4 a C A F E C I A O

b (i) L = (0 0 0 0 0 0 0) 0 1 1 0 0

W = (0 0 0 0 0 0 0) 1 0 1 1 1

(ii) X

c (i)

0	1	1	0	1	0	1	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---

(ii) letters O (15th letter) and S (19th letter)

d (i) $0 0 0 0 1 1 1 1 = 15$

after shifting three places to the left gives:

$0 1 1 1 1 0 0 0 = 120$

this is equivalent to multiplying by 8 (i.e. 2^3)

(ii) $0 1 0 1 1 0 0 0 = 88$

after shifting three places to the right gives: $0 0 0 0 1 0 1 1 = 11$

this is equivalent to dividing by 8 (i.e. 2^3)

Chapter 2

1 a • the device is automatically detected

• a known device is automatically recognised and the device driver loaded

• if a new device is plugged in, the computer looks for the correct device driver or the user will be prompted to download the driver

b asynchronous serial data transmission

c asynchronous – this refers to data being transmitted in an agreed bit pattern; data is grouped together and sent with control bits known as the *start bit* and *stop bit*

synchronous – this is a continuous stream of data which is accompanied by timing signals generated by an internal clock; the timing signals ensure the sender and receiver are synchronised with each other to ensure the bits are reassembled into the correct bit patterns at the receiving end

2 a 4, 10

b • column 4 had odd parity (five 1-bits)

• row 10 had odd parity (five 1-bits)

• intersection of column 4 and row 10 gives incorrect bit

c 1 1 0 0 0 1 1

d • swapped bits, such as 1 1 1 1 0 0 0 0 → 1 1 1 0 1 0 0 0

• several bits changed, such as 1 0 1 1 1 1 0 1 → 1 1 0 1 1 0 1 1

e • automatic repeat request (ARQ)

• checksum

• echo check

3

Description	Term
set of rules that must be obeyed when transferring files or data across the internet	hypertext transfer protocol (http)
id given to a device when it connects to the internet; the id changes each time the device connects	Internet protocol (IP) address
company that provides the user with access to the internet for a monthly subscription	Internet service provider (ISP)
id which uniquely identifies a device connected to the Internet; this id rarely changes	media access control (MAC) address

- 4 a half duplex, simplex transmission has been described
 b synchronous transmission has been described
 c device automatically recognised and device driver automatically loaded; only if device not known is user prompted
 d echo check has been described; ARQ only uses acknowledgement and time out
 e ends in a 1 means number is ODD, but not necessarily odd parity; this requires an odd number of 1-bits

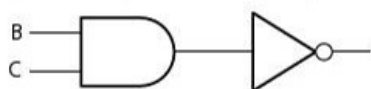
Chapter 3

- 1 a $(A = \text{NOT } 1 \text{ OR } B = 1) \text{ AND } ((B = \text{NOT } 1 \text{ AND } C = 1) \text{ OR } C = \text{NOT } 1)$
 $(\bar{a} + b) \cdot (\bar{b} \cdot c + \bar{c})$

b

Inputs			Working	Output
A	B	C		X
0	0	0		1
0	0	1		1
0	1	0		1
0	1	1		0
1	0	0		0
1	0	1		0
1	1	0		1
1	1	1		0

- c $(\bar{b} \cdot c + \bar{c})$ becomes $(\bar{b} \cdot c)$



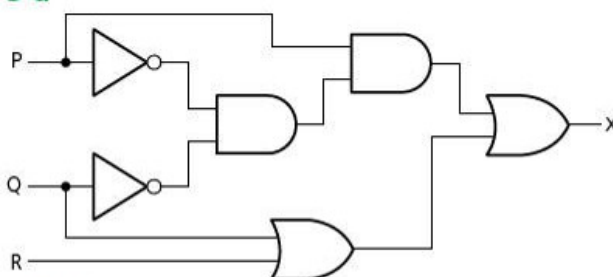
2 a

Inputs		Working	Output
A	B		X
0	0		0
0	1		1
1	0		1
1	1		1

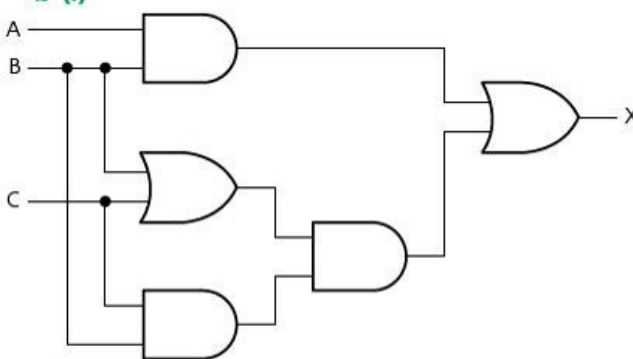
- b OR gate
 c (i) NAND gate

- (ii) • reduced costs
 • more reliable units
 • often easier to locate faults
 • reduced development costs

3 a



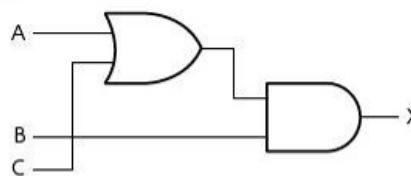
b (i)



(ii)

Inputs			Working	Output
A	B	C		X
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		1
1	0	0		0
1	0	1		0
1	1	0		1
1	1	1		1

(iii)



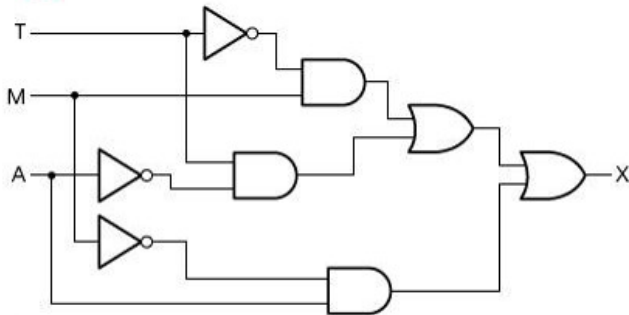
(iv)

Inputs			Working	Output
A	B	C		X
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		1
1	0	0		0
1	0	1		0
1	1	0		1
1	1	1		1

Truth tables in (ii) and (iv) are the same therefore both logic circuits are the same.

4 a $((\bar{T} \cdot M) + (T \cdot \bar{A})) + (\bar{M} \cdot A)$

b



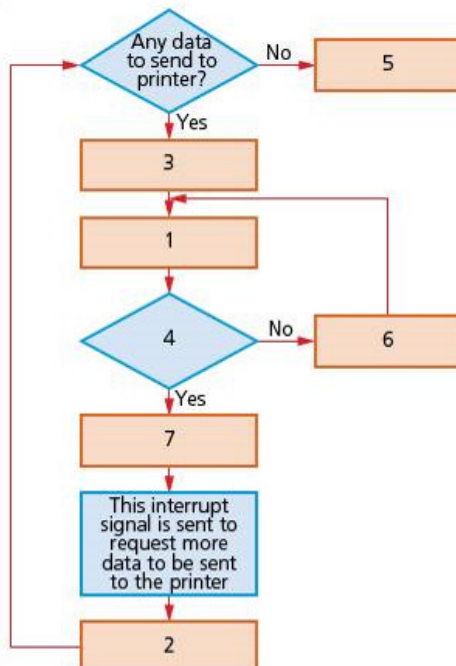
c

Inputs			Working	Output
T	M	A		X
0	0	0		0
0	0	1		1
0	1	0		1
0	1	1		1
1	0	0		1
1	0	1		1
1	1	0		1
1	1	1		0

Chapter 4

- a (i) • temporary area of memory
 • holds data waiting to be printed/sent to the printer
- (ii) • signal sent to processor from hardware or software
 • requesting it to be serviced
 • processor temporarily stops its current tasks

b



- a (i) • one user can use the computer at a time
 • but it is possible to have many applications open simultaneously
- (ii) • when an error occurs the user is warned
 • this prevents the computer from 'hanging up' every time an error is encountered

- b • HCI
 • multiprogramming
 • batch processing
 • load/run software
 • manage user accounts
 • file utilities (copy, save, etc.)
 • processor management
 • memory management
 • real-time processing
 • interrupt handling
 • security
 • I/O control

3 a address bus

- carries signals relating to addresses between processor and memory
- it is uni-directional

data bus

- sends data between processor, memory unit and I/O devices
- it is bi-directional

control bus

- carries signals relating to control and coordination of all activities within the computer
- can be uni-directional or bi-directional

b (i) MAR: 1 0 0 0 0 1 1

MDR: 0 1 0 1 0 1 1 0

(ii) MAR: 1 1 1 1 1 1 0

MDR: 1 1 1 0 0 1 1 1

(iii)

Address	Contents
1111 1100	
1111 1101	
1111 1110	1110 0111
1111 1111	

- 4 a (i) • carries out arithmetic operations
 • carries out logic operations
- (ii) • controls operation of memory, processor and I/O devices
 • sends signals to coordinate all computer components

b

Stage	Order
address is copied from PC to MAR; this is done using the address bus	2
contents at memory location contained in MAR are copied temporarily into MDR	3
contents of MDR are copied and placed in CIR	4
instruction is finally decoded and then executed by sending out signals (via the control bus) to various components of the computer system	6
program counter (PC) contains address of memory location of next instruction to be fetched	1
value in PC is incremented by 1 so that it now points to the next instruction to be fetched	5

Chapter 5

1 a tomography

b QR code

c microphone

d voice recognition

e sensor

2 a



b Full description of automatic stock control in supermarkets (marks for how barcodes are read, how data is stored, how new stock is automatically ordered)

Full description of library system (marks for how barcodes are read, what data is stored and how books coming in and out are tracked)
Accept any other application if fully described

3 a • rate at which DAC can translate digital output ...
• ... into analogue voltages

b (i) $44\,100 \times 2 \times 32 = 2\,822\,400$ bits/sec
= 352 800 bytes/sec

(ii) $352\,800/1024 = 344$ KB/sec

c (i) $344 \times 60 \times 4 = 82\,560$ KB
= $82\,560/1024 = 80$ MB

(ii) $800/80 = 10$

4 a

Statements	True (✓)
Because of plastic organic layers, OLEDs are much thicker and heavier than conventional LCD/LED formats	✗
OLEDs give a brighter light than LEDs leading to more vivid colours	✓
OLEDs produce a much larger field of view than LCD/LED formats	✓
OLEDs require some form of back-lighting so that the screen can be lit up	✗
OLEDs consume more power than LCD/LED formats and thus produce more heat	✗

b Organic layers are 'sandwiched' between two charged electrodes (a *cathode* which is *negatively* charged and an *anode* which is *positively* charged). Applying an electric charge results in the emission of light.

A simple OLED has six layers which are made up as follows:

- a top layer which is made from either glass or plastic/polymer
- a negatively-charged cathode
- an *emissive* layer
- a *conductive* layer
- a positively-charged anode
- a bottom layer made from either glass or plastic/polymer.

The emissive and conductive layers are made from organic materials; the emissive layer produces the light as follows:

When an electric field is applied, electrons flow from the negative cathode towards the positive anode. The emissive layer becomes negative and the conductive layer becomes positive thus producing positive 'holes' – these 'holes' can jump across the boundary from the conductive to the emissive layer. When the 'holes' encounter the electrons they cancel each other out and give out energy in the form of light.

- 5 a • temperature and carbon monoxide sensors constantly send signals/data to the microprocessor
- signals/data first converted into digital using an ADC
 - microprocessor compares sensor data with stored values
 - if temperature > stored value, signal sent to ...
 - ... actuator to open valve to allow more cool water to enter
 - if temperature ≤ stored value, no action is taken
 - if CO levels > stored value, signal sent to ...
 - ... actuator to open valves to admit more air into the engine
 - if CO levels ≤ stored value, no action is taken
 - monitoring continues until engine switched off

b

Application	Suitable sensor
measure/control the dampness of the air in a paint spray booth in a factory	humidity/moisture
automatically switch on a car's head lights when it becomes dark	light
detect intruders entering a building in a burglar/ intruder alarm system	Infra-red/pressure/acoustic
measure the acid pollution levels in a river at a number of points	pH
measure the speed of rotation of wheels on a car in an anti-lock braking system	magnetic field

6 a

Stage	Order
data from the document is sent to a printer driver	1
data sent to the printer is stored in the printer buffer	3
discharge lamp removes all electric charge from the printer drum at the end of the printing process	11
negatively charged sheet of paper rolled over printer drum	7
paper goes through a fuser which fixes ink permanently to the paper	10
printer driver ensures data being sent to printer is in the correct format	2
printer drum is coated in positively charged toner	6
printer drum is given a positive charge	4
toner on printer drum now sticks to the paper to produce text and images	8
to prevent paper sticking to the printer drum, the electric charge on the paper is removed after one rotation of the drum	9
when printer drum rotates, laser beam scans across it removing the positive charge; leaving a negatively-charged print area	5

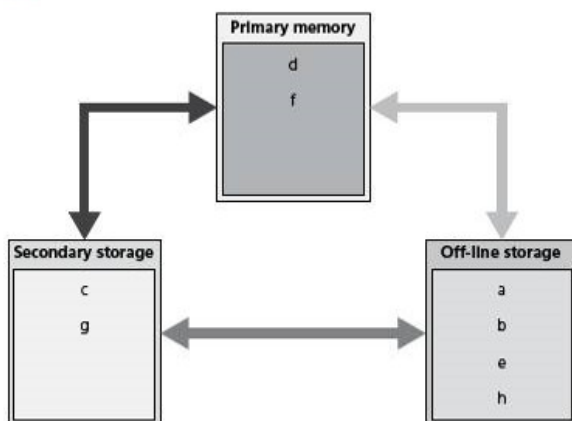
- b (i) laser – printing out 1000s of colour fliers (any task needing high output)
- (ii) inkjet – printing out a high-quality photograph (any one-off task needing high quality)
- (iii) dot matrix – printing labels on the factory floor using continuous stationery (any task where specific features of a dot matrix printer are utilised)

Chapter 6

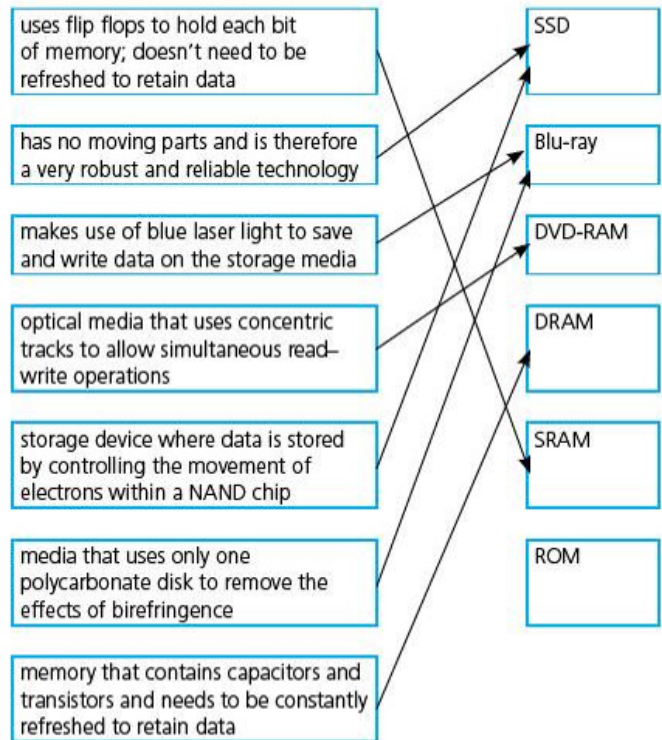
- 1 a lossless – all data from original file can be reconstructed when file is uncompressed
lossy – file compression eliminates unnecessary data and it is not possible to get the original file back
- b (i) $3 + 7 + 4 + 3 + 5 + 6 + 4 + 5 + 4 + 7 + 8 + 8 = 64$ minutes
 $64 \times 12 = 768$ MB
- (ii) 10% of 768 = 76.8 MB
- (iii) perceptual music shaping; removes sounds human ear can't distinguish

2

It is illegal to photocopy this page



3



Chapter 7

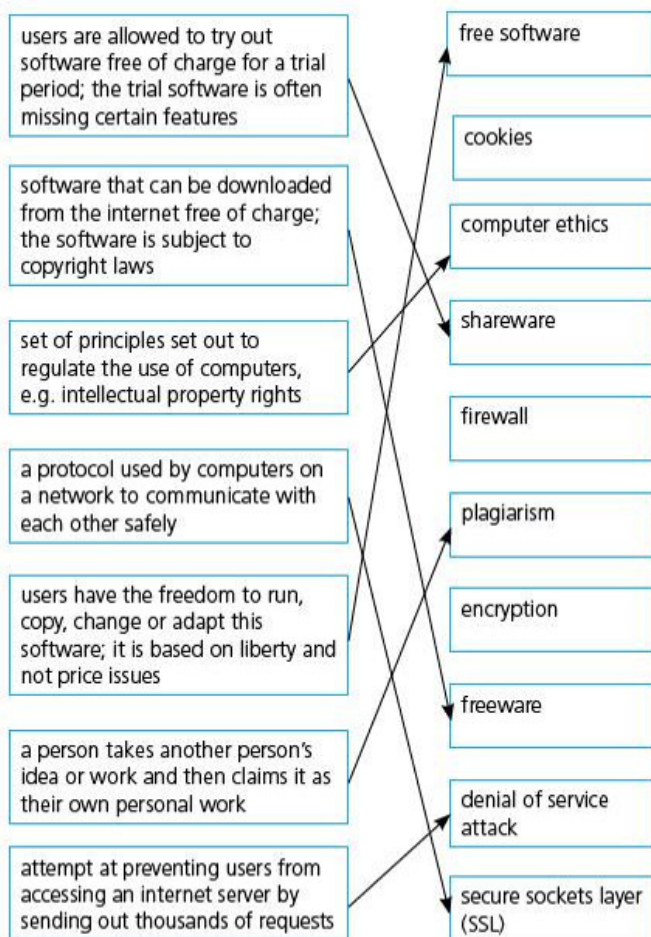
- 1 a Compiler
b Assembler
c Interpreter
- 2 a Assembly language
b High-level language
c Machine code (hexadecimal)
d Machine code (binary)
- 3 An assembler [1 mark] translates a program written in assembly or low-level language into machine code. [1 mark]
A compiler [1 mark] translates a program written in a high-level language into machine code. [1 mark]
An interpreter [1 mark] executes a program written in a high-level language one statement at a time. [1 mark]
- 4 a A program written in a high-level language takes a longer time to complete the same task than a program written in a low-level language. [1 mark] A program written in a high-level language takes up more space in memory than a program written in a low-level language to perform the same task. [1 mark]
- b Programs written in a low-level language are more difficult for programmers to understand than programs written in a high-level language. [1 mark] Programs written in a low-level language take longer for programmers to debug than programs written in a high-level language. [1 mark]

Chapter 8

1

Statements	Ethics followed (✓)	Ethics NOT followed (✗)
a technician works for a software company during the day but also does freelance computer advice in the evenings	✓	
a software engineer develops software for a security company; he uses some of the algorithms when developing computer games for a different company		✗
a software engineer writes software for company A but also accepts money from company B to include routines in the software which covertly advertises company B		✗
a company employs people in different countries to write software modules; these modules are put together to produce software packages	✓	
a software engineer undergoes training by a different company to improve his software writing skills	✓	

2



3 Hacking

- this is a way of gaining illegal access to a computer system
- this can lead to identity theft and loss or corruption of data
- the risk of hacking can be minimised by using strong passwords and firewalls

Cracking

- this is the editing of program source code so that it can be exploited or changed for a specific purpose (mostly an illegal act)
- it is often done for malicious purposes, e.g. modifying legitimate software to do something like redirect a user to a fake website
- it is difficult to guard against; software engineers need to make the act of breaking into the software nearly impossible (i.e. make it difficult to identify 'back doors')

Viruses


- a virus is program code that can replicate/copy itself with the intention of deleting/corrupting data/files or cause the computer to malfunction
- they can cause the computer to run slow (due to the hard disk filling up with data, for example) or crash (e.g. due to deletion of some key .exe files); they can also cause some software to run abnormally (due to loss or corruption of files/data)
- the risk of viruses can be minimised by running anti-virus software or not opening emails or software from unknown sources

Phishing

- legitimate-looking emails are sent to users; on opening the email, the user could be asked to supply personal or financial details or they may be asked to click on a link which sends them to a fake/bogus website where the user could be asked to supply personal data
- once the user is sent to the fake/bogus website, they may be asked to give out personal or financial data
- many ISPs filter out phishing emails; the user should also be very cautious about opening emails from unknown sources

Pharming

- malicious code is installed on a user's computer or web server; the code redirects the user to a fake or bogus website without their knowledge
- once the user is sent to the fake/bogus website, they may be asked to give out personal or financial data

- some anti-spyware software can identify and remove pharming code on the hard drive; the user should also look out for clues that they are being redirected (websites which are safe can usually be identified by https or by the green padlock  sign in the status bar)

Wardriving

- this is the act of locating and using wireless internet connections illegally
- stealing of internet time and bandwidth; could also lead to user's passwords and other data being intercepted
- use of WEP (wired equivalent privacy) encryption; use of complex passwords before accessing the internet and use of firewalls to prevent outsiders gaining access

Spyware/key logging software

- gathers data by monitoring key presses on user's keyboards and sending the data back to the person who sent the spyware
- sends important data, such as passwords, back to the originator of the spyware; it can also allow the originator to change settings on the user's computer
- use of anti-spyware; use of mouse to select characters from a drop down box to enter passwords rather than use a keyboard

- 4 a
- accidental mal-operation
 - hardware malfunction
 - software malfunction
 - malicious act (e.g. virus, hacking, etc.)
- b
- this can be guarded against by doing regular back-ups of data (although this won't allow recovery following virus attack); use of passwords and user ids to protect the data (all examples)
 - this can be guarded against by regular back-ups of data, use of UPS to prevent 'power glitches/loss' and use of parallel hardware (hardware fault)
 - this can be guarded against by regular back-ups of data and the saving of data every 30 mins, for example, in case the computer becomes unresponsive (all examples)

- 5 a
- hardware or software that sits between computer and external network
 - it filters out information coming to and going from the computer
- b
- monitors traffic in and out
 - checks if traffic meets given set of criteria
 - if data fails criteria, firewall blocks the traffic and informs user
 - logs all incoming and outgoing traffic
 - criteria can be set to prevent access to certain websites

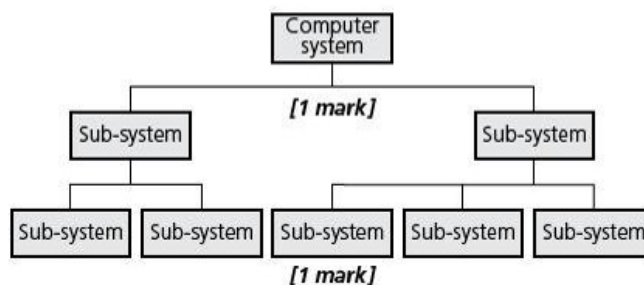
- helps to prevent viruses or hackers entering user's computer
- warns user if installed software tries to communicate with an external source

- 6 a
- b
- c
- d
- e
- f

Chapter 9

- 1 Software [1 mark] data [1 mark] hardware [1 mark] communications [1 mark] people [1 mark]

- 2 Each computer system can be divided up into a set of sub-systems. [1 mark] Each sub-system can be further divided into sub-systems and so on until each sub-system just performs a single action. [1 mark]



- 3 Normal data [1 mark] to test that a program processes the data as it should, [1 mark] for example inputting three exam marks and checking that the average output is correct. [1 mark] Using erroneous data [1 mark] to test that a program rejects this type of data and outputs a suitable error message, [1 mark] for example inputting an exam mark of -1 which should be rejected. [1 mark]

Using extreme data [1 mark] to test that a program accepts and processes this type of data correctly, [1 mark] for example inputting an exam mark of 0 which is the lowest mark that should be accepted and processed. [1 mark]
Note there are many acceptable examples.

- 4 a A range check [1 mark] and a presence check [1 mark]
 b Double entry [1 mark] and a visual check by user [1 mark]
- 5 Name – verification to ensure that no errors are made when typing the name, visual check by user [2 marks] OR validation as this field must be present, presence check [2 marks]
 Address – verification to ensure that no errors are made when typing the address, visual check by user [2 marks]
 Date of birth – validation to ensure that the date is not impossible for example in the future, range check [2 marks]
 Shoe size – validation as there is a number of values that are acceptable, range check [2 marks]
 Password – verification to ensure that the password has been typed correctly, double entry [2 marks]

- 6 a Counter \leftarrow 1 should be Total \leftarrow 0 [1 mark]

UNTIL Number \geq 0 test incorrect [1 mark]

Total \leftarrow Total + Counter incorrect running total [1 mark]

Counter \leftarrow Counter - 1 a counter should not be decremented in a FOR ... NEXT loop [1 mark]

OUTPUT Total and NEXT are in the wrong order [1 mark]

- b Total \leftarrow 0 [1 mark]

FOR Counter \leftarrow 1 TO 6

REPEAT

PRINT "Enter a negative whole number"

INPUT Number

UNTIL Number < 0 [1 mark]

Total \leftarrow Total + Number [1 mark]

NEXT

OUTPUT Total [1 mark and 1 for keeping the correct statements]

c

Counter	Number	Total	Output
		0	
1	-4	-4	
2	-3	-7	
3	1		
	-2	-9	
4	0		
	-5	-14	
5	-5	-19	
6	27		
	-1	-20	-20

[1 mark for each of the Counter and Number columns and 1 for the Total and Output columns together]

- d Counter \leftarrow 1 was a redundant statement and has been removed [1 mark]. Total is output only once, instead of 6 times [1 mark]. Also the loop would never have ended as the counter was always updated by the FOR ... NEXT loop and then decremented [1 mark].

● Chapter 10

1 FOR ... TO ... NEXT [1 mark] for a fixed number of repetitions [1 mark]
 REPEAT ... UNTIL [1 mark] for a loop where the code is run at least once [1 mark]
 WHILE ... DO ... ENDWHILE [1 mark] for a loop where the code may not be executed if the conditions are not met [1 mark]

2 IF ... THEN ... ELSE ... ENDIF [1 mark] a condition that can be true or false [1 mark]

```
IF Age < 18
```

```
    THEN PRINT "Child"
```

```
    ELSE PRINT "Adult"
```

```
ENDIF [1 mark]
```

CASE ... OF ... OTHERWISE ... ENDCASE [1 mark] a choice between several different values [1 mark]

```
CASE Grade OF
```

```
    "A" : PRINT "Excellent"
```

```
    "B" : PRINT "Good"
```

```
    "C" : PRINT "Average"
```

```
    OTHERWISE PRINT "Improvement is needed"
```

```
ENDCASE [1 mark]
```

3 INPUT Number [1 mark]

```
WHILE Number < 1 OR Number > 10 DO [1 mark]
```

```
    OUTPUT "Please Enter a value between 1 and 10" [1 mark]
```

```
    INPUT Number
```

```
ENDWHILE
```

4 Selection provides alternative routes through the program code [1 mark] for example

```
IF Age < 18
```

```
    THEN PRINT "Child"
```

```
    ELSE PRINT "Adult"
```

```
ENDIF [1 mark]
```

Iteration means that the program code is repeated a number of times [1 mark] for example

```
FOR Counter ← 1 TO 10
```

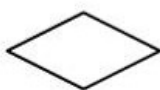
```
    PRINT "*"
```

```
NEXT [1 mark]
```

5



Input/output
[1 mark]

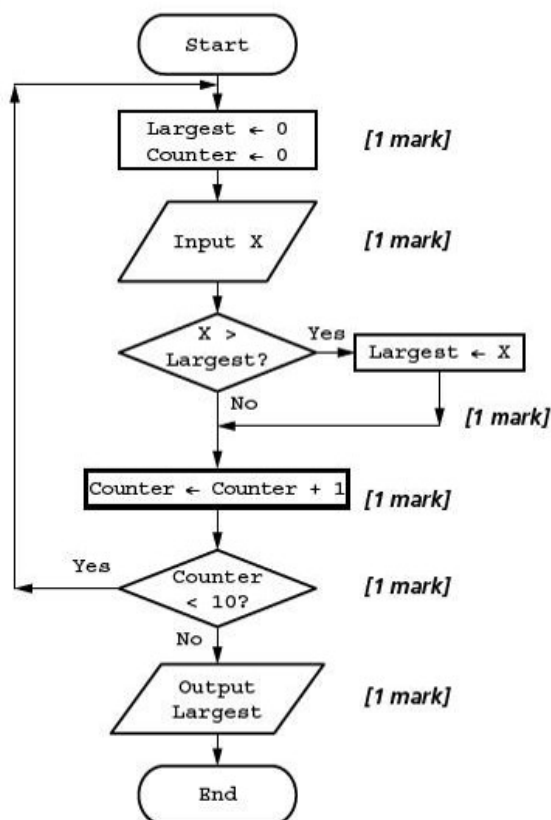


Decision
[1 mark]



Process
[1 mark]

6



Chapter 11

- 1 A constant does not change its value for example VAT could be set to 20%. [1 mark]
 A variable can have its value changed for example NumberSold could be used for input. [1 mark]

2 REPEAT

INPUT ReorderLevel

UNTIL ReorderLevel > 0 [1 mark]

INPUT NumberOnShelf

WHILE NumberOnShelf <> -1 DO

IF NumberOnShelf < ReorderLevel AND NumberOnShelf >= 0 [1 mark]

THEN PRINT "You need to reorder" [1 mark]

INPUT NumberOnShelf [1 mark both inputs of NumberOnShelf]

ENDWHILE [1 mark]

3 Line 10, Highest = 1000000 [1 mark]

Line 30, WHILE Count < 15 [1 mark]

Line 50, IF Highest > Number [1 mark]

Line 60, Count = Count - 1 [1 mark]

Line 70, PRINT Number and line should move to after ENDWHILE [1 mark]


```

10 Highest = 0
20 Count = 1
30 WHILE Count < 16 [1 mark no errors in 10, 20 and 30]
40     INPUT Number
50     IF Number > Highest THEN Highest = Number [1 mark no errors in 40 and 50]
60     Count = Count + 1
70 ENDWHILE
80 PRINT Highest [1 mark no errors in 60, 70 and 80]
    
```

- 4 Counting is adding one every time an action is taken [1 mark] for example `Count = Count + 1`. [1 mark] Totalling is adding a value to a running total [1 mark] for example `Total = Total + Number`. [1 mark]
- 5 Erroneous test data [1 mark] -20 [1 mark] to check that negative reorder levels are rejected. [1 mark] Normal test data [1 mark] 10, 20, 8 [1 mark] and -1 to check that the program's tests work as expected. [1 mark]

Chapter 12

- 1 Constant `HeightCheck = 1.5`, [1 mark] data type `real` [1 mark] used to check for pupils over 1.5 metres in height. [1 mark]
- 2 Variable `NoPupils` [1 mark] data type `integer` [1 mark] used to store the number of pupils in the class. [1 mark] Variable `TallestPupil` [1 mark] data type `real` [1 mark] used to store the height of the tallest pupil in the class. [1 mark]
- 3 I used one counter as an index [1 mark] to search through the array of pupil heights. [1 mark] I used another counter and added one to this counter every time a pupil height was greater than 1.5, [1 mark] this counter was set to zero at the start of the program. [1 mark]
- 4 I set up a variable to store the largest shoe size, which was set to zero at the start of the program. [1 mark] I used a counter as an index to search through the array of pupil shoe sizes and check if a value was greater than the one stored, [1 mark] if so the largest shoe size was replaced by this value. [1 mark] I used a running total in a loop to add all shoe sizes together then I divided this answer by the number of pupils in the class [1 mark] and turned this into a whole number to display. [1 mark]

Note: this answer is worth more than the 4 marks on offer but it is often the case that there are more marking points than marks.

Chapter 13

- 1 There are 5 fields [1 mark] and 7 records. [1 mark]
- 2 The field `Hosp No` should be used as the primary key [1 mark] as it uniquely identifies each baby. [1 mark]
- 3 Name data type `text`, [1 mark] validation check type check [1 mark]
- Hosp No data type `text`, [1 mark] validation check format check [1 mark]
- DOB data type `date`, [1 mark] validation check range check [1 mark]
- Gender data type `text`, [1 mark] validation check length check [1 mark]
- Address data type `text`, [1 mark] validation check presence check [1 mark]

4

Field:	Name	Gender	DOB	
Table:	BABY	BABY	BABY	
Sort:			Ascending	
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:		= 'F'		
or:				
	[1 mark]	[1 mark]	[1 mark]	

Index

2D scanners 39
2D/3D cutters 44
3D printers 44
3D scanners 39
8-bit register 2, 3–4

A

abnormal test data 70, 71
accidental loss of data 60, 62, 66
address bus 33, 35
AND gate 21, 22, 26–7
arithmetic and logic unit (ALU) 34
arrays 87–9
assemblers 57, 58
assembly code 5, 8, 9
assembly languages 56, 57
asymmetric encryption 60, 64–5
asynchronous transmission 12, 14
authentication 60, 65
automatic repeat request (ARQ) 15, 16, 17

B

barcode readers 40
binary registers 2, 3–4
binary system 1–4
blu-ray disks 52, 53–4
Boolean data type 82, 83
boundary data 71
buffers 31, 32–3
buses 31, 33, 35

C

capacitive touchscreens 41
CDs 52
char data type 82, 83
checksums 15, 16, 17, 72
compilers 57, 58
computer architecture 31, 33–5
computer programs 56–7
computer systems 69–70
conditional statements 77
constants 82–3
control bus 33, 35
control systems 41–2, 43
control unit 31, 34
cookies 60, 62, 63
counting 78–9, 84–5
cracking 60, 61
current instruction register (CIR) 34, 35, 36, 37
cutters 44

D

databases 90–3
data bus 33, 35
data integrity 62–4, 66
 security and 60–2
data structures 87–9
data transmission 12–14
 error checking of data
 following 15–17
data types 82–3, 91
denary system 1
denial of service attacks 60, 65, 67
digital cameras 40

digital light projectors (DLPs) 45
dot matrix printers 43–4, 45–6
dry run 73
DVD-RAM 52
DVDs 52, 53–4

E

echo check 15, 16, 17
effective solution 73
elements of an array 87
encryption 60, 63, 64–5
erroneous test data 70, 71
error checking of data 15–17
ethics 60, 65–6
extreme data 71

F

fetch–execute cycle 35–7
fields (columns) 90, 91, 92, 93
file formats 49–51
firewalls 60, 62, 63–4, 66, 67
flash memory 53
flowcharts 70, 77, 79–81
free software 60, 65
freeware 60, 65
full-duplex 12, 13

H

hacking 60, 63
half-duplex 12, 13
hard disk drives (HDDs) 52, 53–4
hardware faults 62
hexadecimal system 1, 4–9
 uses 5, 7–9
high-level languages 56, 57
HTML (hypertext mark-up language) 5, 7, 8
 presentation 17
 structure 17
hypertext transfer protocol (http) 17

I

index number 87–8
infra-red touchscreens 41
inkjet printers 43–4, 45–6
input devices 39–43
integers 82, 83
interactive whiteboards 42
internet protocol (IP) address 17, 18
internet service provider (ISP) 17, 18
internet technologies 17–19
interpreters 57, 58
interrupts 31, 32–3

J

joint photographic experts group (JPEG) files 49, 50

K

keyboards/keypads 40
key logging software 60, 62

L

laser printers 43–4, 45–6
LCD monitors 45, 46
LCD projectors 45

LED monitors 45, 46
library routines 69, 70
light projectors 45
logic circuits 21, 24–7
logic gates 21–3, 25
logic statements 24
loop structures 77, 78, 79, 84–5, 87
lossless file compression 49, 50
lossy file compression 49, 50
loudspeakers 44–5
low-level languages 56

M

machine code 5, 8, 9, 56, 57, 58
malicious loss of data 60, 62, 66
media access control (MAC) address 5, 7–8, 17, 18
memory 51–4
memory address register (MAR) 34, 36
memory data register (MDR) 34, 36
memory dumps 5, 7, 8
memory size measurement 2
memory sticks 53
memory unit 31, 34–5
microphones 41
monitoring systems 41–2, 43
monitors 45, 46
mouse 41
MPEG-3 (MP3) files 49, 50–1
MPEG-4 (MP4) files 49
musical instrument digital interface (MIDI) 49, 50–1

N

NAND gate 22, 23, 26–7
NOR gate 22
normal data 70, 71
NOT gate 21, 22
number files 50

O

off-line storage 52–3
one-dimensional array 87
operating systems 31, 32
organic light-emitting diodes (OLEDs) 45, 46
OR gate 21, 23, 26–7
output devices 39, 43–6

P

parallel transmission 12, 13
parity checks 15–16
pharming 60, 61, 63
phishing 60, 61, 63
pointing devices 41
pre-release material 88–9
primary key 90, 91
primary memory 49, 51, 53
printers 32–3, 43–4, 45–6
program counter (PC) 34, 35, 36, 37
programming 82–6
 techniques 84–5
programming languages 56–7
projectors 45
pseudocode 77–9

Q

QR (quick response) codes 40, 42–3
 query by example grid 91–3

R

Random Access Memory (RAM) 51, 53
 Read Only Memory (ROM) 51, 53
 real numbers 82, 83
 records (rows) 90, 91
 registers 2, 3–4, 31, 34, 36
 repetition 84
 resistive touchscreens 41

S

scanners 39
 secondary storage 49, 52–3
 security 60–2, 63
 security protocols 60, 64, 66
 security sockets layer (SSL) 60, 64, 66
 selection 84
 sensors 41–2, 43, 44
 sequence 82, 84

serial transmission 12, 13
 shareware 60, 66
 simplex 12, 13
 software faults 62
 solid-state drives (SSDs) 52, 53–4
 speech recognition 41, 42
 spyware 60, 62
 standard actions 78
 storage 51–4
 string data type 82
 structure diagrams 69, 70
 sub-routines 69
 sub-systems 69–70
 symmetric encryption 60, 64
 synchronous transmission 12, 14

T

tables 90
 test data 69, 70–1, 73, 74
 text files 50, 51
 top-down design 69, 70
 totalling 84–5
 touchscreens 41
 trace tables 69, 73, 74–5

tracing algorithms 73–5
 trackerballs 41
 translators 56, 57–8
 transport layer security (TLS) 60, 64
 truth tables 21–2, 23, 24, 26–7

U

universal serial bus (USB) 12

V

validation 69, 72–3, 89
 variables 82–3
 verification 69, 72–3
 viruses 60, 61, 63
 voice recognition 41, 42
 von Neumann architecture 33, 35, 37

W

wardriving 61
 web addresses 5, 8, 9
 web browser 18

X

XOR gate 22, 23