PG ONLINE

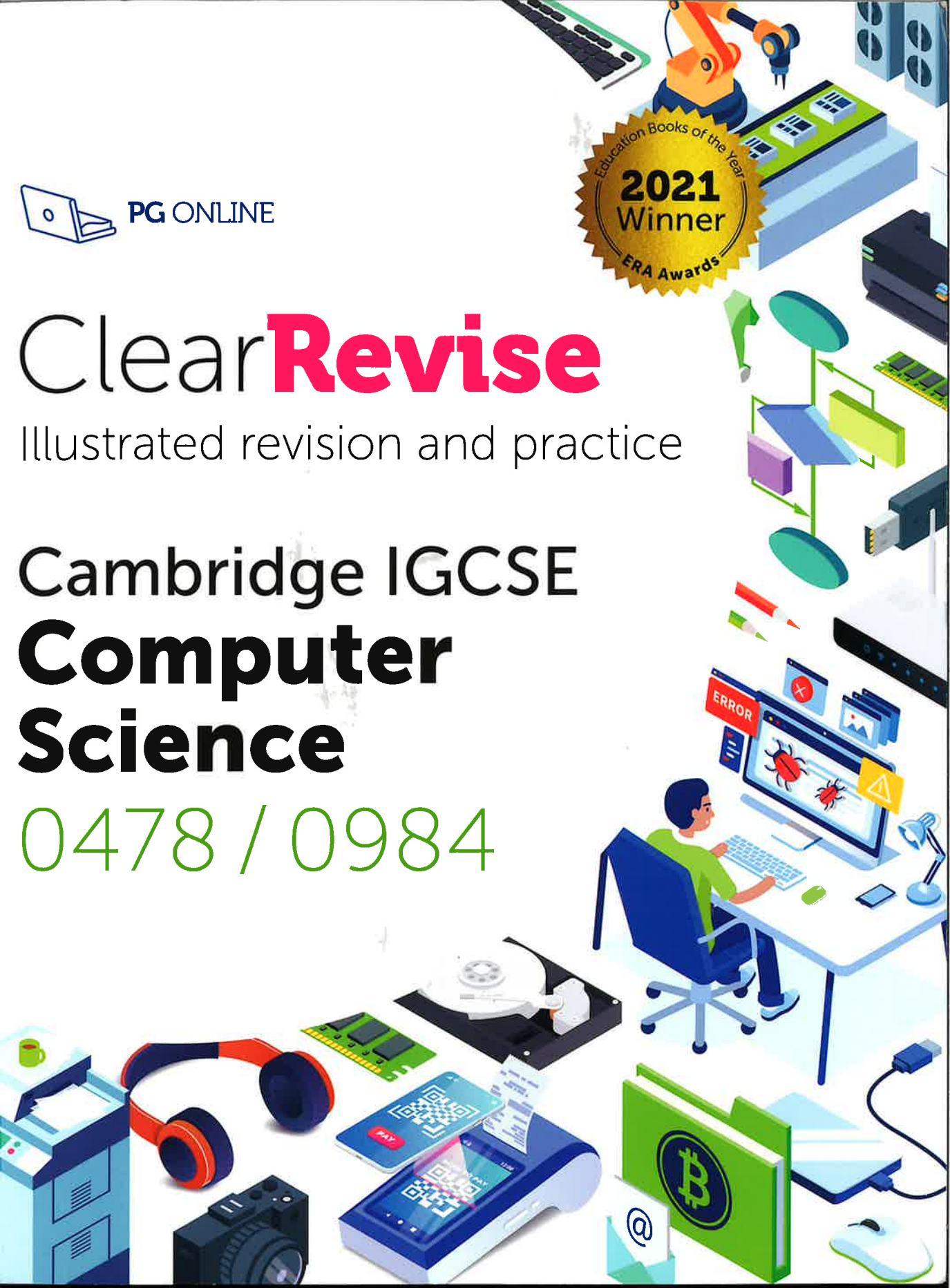
Education Books of the Year
2021 Winner
ERA Awards

ClearRevise

Illustrated revision and practice

Cambridge IGCSE Computer Science

0478 / 0984



CONTENTS

Paper 1 Computer systems

Topic 1 Data representation ☑

Specification			
1.1.1, 1.3.1	Binary representation	2	☐
1.1.2a, 1.1.2b(i)	Binary ⇄ denary conversion	3	☐
1.2.1, 1.1.2b(ii),(iii)	Hexadecimal ⇄ binary conversion	4	☐
1.1.2	Hexadecimal ⇄ denary conversion	5	☐
1.1.3	Uses of hexadecimal	6	☐
1.1.4	Adding binary integers	7	☐
1.1.5	Logical binary shifts	8	☐
1.1.6	Two's complement signed integers	9	☐
1.2.1	Representing text	10	☐
1.2.2, 1.3.2	Representing sound	11	☐
1.2.3, 1.3.2	Representing images	12	☐
1.3.3, 1.3.4	Data compression	14	☐
	Examination practice	**15**	☐

Topic 2 Data transmission ☑

2.1.1	Data packets	17	☐
2.1.2, 2.1.3	Data transmission	18	☐
2.2.1, 2.2.2, 2.2.4	Transmission error detection	20	☐
2.2.3	Check digits	22	☐
2.3	Encryption	23	☐
	Examination practice	**24**	☐

Topic 3 Hardware ☑

3.1.1, 3.1.2	Computer architecture	26	☐
3.1.3	CPU performance	28	☐
3.1.4	Instruction sets	29	☐
3.1.5	Embedded systems	29	☐
3.2.1	Scanners	30	☐
3.2.1	Digital cameras	32	☐
3.2.1	Keyboards, mice and microphones	32	☐
3.2.1	Touch screens	33	☐
3.2.2	Screen types	34	☐
3.2.2	Projectors	35	☐
3.2.2	Printers	36	☐
3.2.2	3D printers	37	☐
3.2.2	Speakers	37	☐
3.2.2, 3.2.3	Actuators and sensors	38	☐

3.3.1	Primary storage	39	☐
3.3.2	Secondary storage	40	☐
3.3.3	Device operation	41	☐
3.3.4	Virtual memory	42	☐
3.3.5, 3.3.6	Cloud storage	43	☐
3.4.1, 3.4.2	Network hardware	44	☐
3.4.3, 3.4.4	Internet Protocol (IP) addresses	45	☐
	Examination practice	**46**	☐

Topic 4 Software ☑

4.1	Application and system software	48	☐
4.2.1	Types of programming language	50	☐
4.2.2, 4.2.3, 4.2.4	Translators	51	☐
4.2.5	The Integrated Development Environment (IDE)	52	☐
	Examination practice	**52**	☐

Topic 5 The internet and its uses ☑

5.1.1-4	The Internet and the World Wide Web	54	☐
5.1.5, 5.1.6	Locating and displaying a web page	55	☐
5.2	Digital currency	56	☐
5.3	Cyber security threats	57	☐
5.3.1	Malicious code (Malware)	58	☐
5.3.2	Keeping data safe	59	☐
	Examination practice	**60**	☐

Topic 6 Automated and emerging technologies ☑

6.1	Automated systems	61	☐
6.2	Robotics	62	☐
6.3	Artificial intelligence	63	☐
	Examination practice	**64**	☐

Paper 2 Algorithms, programming and logic

Topic 7 Algorithm design and problem solving ☑

7.1	The program development life cycle	66	☐
7.1, 7.2	Structure diagrams	67	☐
7.1, 7.2c	Flowcharts	68	☐
7.1, 7.2c, 7.4	Pseudocode	69	☐
7.2b, 7.3	Explain the purpose of a given algorithm	70	☐
7.4	Linear search	71	☐
7.4	Bubble sort	72	☐

7.5	Validation and verification	74	☐
7.6	Testing	76	☐
7.4, 7.7	Trace tables	77	☐
7.8, 7.9	Types of error	78	☐
	Examination practice	**79**	☐

Topic 8 Programming ☑

8.1.1, 8.1.2	Variables, constants, assignments	82	☐
8.1.3	Input/output	83	☐
8.1.4f	Arithmetic and relational operators	84	☐
8.1.4a, 8.1.4b, 8.1.4f	Sequence and selection	85	☐
8.1.4b, 8.1.5	IF and CASE statements	86	☐
8.1.4c-d, 8.1.5	Count-controlled loops	87	☐
8.1.4c-d	Condition-controlled loops	88	☐
8.1.4e	String handling	89	☐
8.1.6	Procedures and functions	90	☐
8.1.6, 8.1.7	Local and global variables	91	☐
8.1.7	Library routines	91	☐
8.1.8	Creating a maintainable program	92	☐
8.2	Arrays	93	☐
8.2	Two-dimensional arrays	94	☐
8.3	File handling	95	☐
	Examination practice	**96**	☐

Topic 9 Databases ☑

9.1-3	Databases	100	☐
9.4	Structured Query Language (SQL)	101	☐
9.1	Validation	102	☐
	Examination practice	**103**	☐

Topic 10 Boolean logic ☑

10.1, 10.2	Boolean logic	104	☐
10.1, 10.2	NAND, NOR and XOR (EOR) gates	106	☐
10.3	Logic expressions, logic circuits and tables	107	☐
	Examination practice	**109**	☐

	Examination practice answers	110
	Index	118
	Examination tips	**121**

TOPICS FOR PAPER 1
COMPUTER SYSTEMS

Information about Paper 1

Written exam: 1 hour and 45 minutes
75 marks
50% of IGCSE

Specification coverage

Data representation; data transmission; hardware; software; the Internet and its uses; automated and emerging technologies.

The content for this assessment will be drawn from subject content sections 1 to 6 of the specification.

Questions

This paper consists of short-answer and structured questions. All questions are compulsory. Calculators are not permitted.

BINARY REPRESENTATION

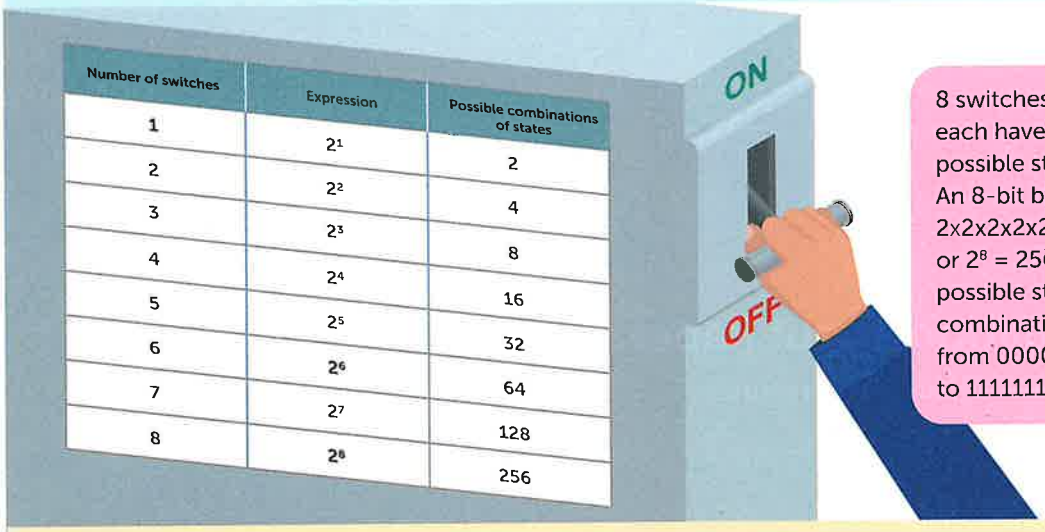
Bit (b) 0 or 1	Nibble 4 bits	Byte (B) 8 bits	Kibibyte (KiB) 1024 bytes	Mebibyte (MiB) 1024 KiB	Gibibyte (GiB) 1024 MiB	Tebibyte (TiB) 1024 GiB	Pebibyte (PiB) 1024 TiB	Exbibyte (EiB) 1024 PiB

A computer is made up of billions of **switches**, each with two states - an **off** position (represented by a 0) and an **on** position (represented by a 1). This is known as **binary**. All data therefore needs to be converted into binary before it can be processed by a computer. Data is processed using logic gates and stored in registers. See pages 27 and 104 for more details.

With two or more switches, the number of combinations of 1s and 0s is doubled with each additional switch or **bit**.

Computers use binary to represent everything including numbers, text, sound, graphics and program instructions. The number of binary digits (or switches) used determines the number of states that can be represented.

Number of switches	Expression	Possible combinations of states
1	2^1	2
2	2^2	4
3	2^3	8
4	2^4	16
5	2^5	32
6	2^6	64
7	2^7	128
8	2^8	256



8 switches or bits each have two possible states. An 8-bit byte has 2x2x2x2x2x2x2x2 or $2^8 = 256$ possible states, or combinations of bits, from 00000000 to 11111111.

Calculate the following:

(a) Calculate the number of 650 MiB CDs required to store 2 GiB of images. Show your working. [2]

(b) Calculate in TiB the total capacity of a server with 4 × 2.5 TiB hard disk drives. [1]

(c) Calculate the total storage requirement for a database of 5,000 customer records each of 1.5 kiB each. Give your answer in MiB. Show your working. [2]

(d) Calculate the maximum number of states that can be represented in a binary pattern of 10 bits. [1]

BINARY ⇄ DENARY CONVERSION

Converting binary numbers into positive denary whole numbers

Our denary or decimal system has a **base of 10** digits 0–9. Binary has a **base** of just **2** digits, 0 and 1. Instead of representing three-digit numbers with a ones, tens and hundreds column for example, binary represents them with a ones, twos and fours column, and so on.

To make a conversion from binary to denary, add the place value headers where there is a 1.

128	64	32	16	8	4	2	1
0	1	1	0	1	0	0	1
	1x64 +	1x32 +		1x8 +			1x1 = 105

Converting positive denary whole numbers to binary

To convert the denary number 87 into binary, start with the **most significant bit** (left-hand end of the table below). Does 128 go into 87? If not, add a 0 in that column. Does 64 go into 87? Yes, it does, so add a 1 to the column and calculate the remainder, 23. 32 does not go into 23 so add a 0 to the next column. 16 goes into 23 with a remainder of 7. 8 won't go into 7 so add a 0 next. 4 will go with a remainder of 3. 2 will go into 3 with a remainder of 1 and 1 goes into 1 so add a 1 to each of the last three columns.

Note that the maximum value that can be held with eight bits where all bits = 1, is 255.

A binary number with a 1 in the least significant bit (far right-hand position) will always be odd.

Most significant bit — Least significant bit

128	64	32	16	8	4	2	1
0	1	0	1	0	1	1	1
	r23		r7		r3	r1	

1. Convert the following denary numbers to binary:
   (a) 138 [1]
   (b) 739 [2]
2. Convert the following binary numbers to denary:
   (a) 0110 1101 [1]
   (b) 1001 0001 1100 1010 [2]

Counting in binary

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

HEXADECIMAL ⇄ BINARY CONVERSION

The **hexadecimal** number system uses **base 16** instead of base 2 or base 10. Given that we only have ten digits 0–9 in our system, the additional six number 10–15 in the hexadecimal system are represented by the letters A–F.

Denary	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Denary	Binary	Hex
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

One hexadecimal (or hex) number can represent one nibble of four bits. Hex is easier to remember than the binary representation.

Converting a binary number into hexadecimal

To convert the number 0100 1111 to hexadecimal, first split the eight-bit binary number into two nibbles of four bits each. Convert each nibble separately and join the results.

0100	**1111**	=	**01001111**
4	**15 (F)**		**4F**

Some further examples are:
1011 0101 = **B5** and 1100 1101 = **CD**

Converting a hexadecimal number into binary

Convert each hex character into a four-bit binary value and join them to make a byte.

7	**E (14)**	=	**7E**
0111	**1110**		**01111110**

Further examples are: B9 = 1011 1001 and DA = 1101 1010.

1. Convert the following binary values into hexadecimal: [3]
   (a) 0110 1011
   (b) 0000 1001
   (c) 1111 1111
2. Convert the following hexadecimal values into binary: [3]
   (a) 48
   (b) 6A
   (c) F9

To convert between hexadecimal and denary, you can convert via binary.

HEXADECIMAL ⇄ DENARY CONVERSION

To convert between **hexadecimal** and **denary**, you need to remember that hex has a base of 16, as opposed to our denary number system that has a base of 10. This means that instead of 1s and 10s, you have 1s and 16s.

Converting a hexadecimal number into denary

Multiply the hexadecimal digits by their column place values 16 and 1, then add the results.

To convert the hex number 5B to decimal:

16	**1**		
5	B	(B = 11)	
(5 × 16)	(11 × 1)		
80	11	80 + 11 =	91

Here are some further examples:
Hexadecimal 88 = denary **136** and hexadecimal FA = denary **250**

Converting a denary number into hexadecimal

First work out how many 16s go into the number. This is the first hex digit. Then take the remainder and use this as the second hex digit. To convert 195 to hex:

195 / 16 = 12 remainder 3
12 = **C**
3 = **3** so 195 in denary is **C3** in hexadecimal

Here are some further examples:
Denary 67 = hexadecimal **43**
Denary 219 = hexadecimal **DB**

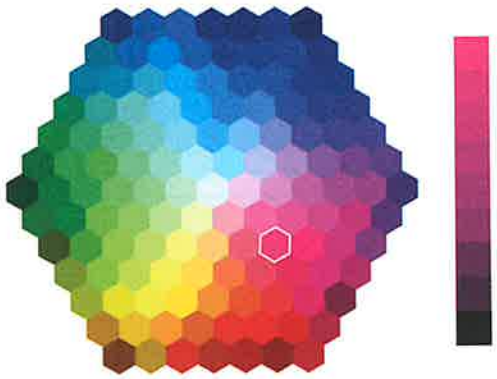
1. Convert the following hexadecimal values into denary: [3]
   (a) 9F (b) C2 (c) 63
2. Convert the following denary values into hexadecimal: [3]
   (a) 63 (b) 160 (c) 15

USES OF HEXADECIMAL

Hexadecimal numbers are easier for humans to read and remember than binary, so they are used in the following situations:

Colour values in photo editing software and web design

One hexadecimal (or hex) number can represent one nibble of four bits. This is easier to remember than the binary representation.



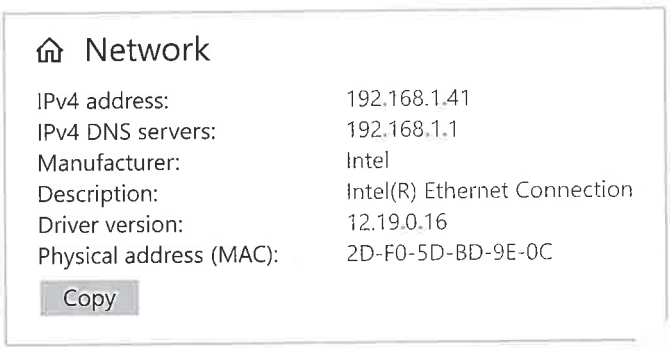
Hexadecimal colour number: ED468C

Other uses of hexadecimal include IPv6 addresses and Unicode.

State why memory address locations in assembly language are commonly expressed in hexadecimal rather their binary equivalent. [1]

MAC addresses

MAC addresses are unique numbers given to all Internet-connected devices. Each address is written in hexadecimal. See **page 44** for more detail on MAC addresses.



ADDING BINARY INTEGERS

Binary addition is done in the same way that denary numbers might be added together.

The rules are as follows:

0 + 0 = 0
0 + 1 or 1 + 0 = 1
1 + 1 = 0 carry a 1
1 + 1 + 1 = 1 carry a 1

Note that in the same way as the denary values 00028 and 28 represent the same value, the binary value 00011100 is the same as 11100. Any leading zeros (left-hand side) are ignored.

Carry	1	1	1	1		1			Check
	0	1	0	1	1	0	1	1	91
+	0	0	1	1	1	0	1	0	58
	1	0	0	1	0	1	0	1	149

Overflow

Overflow occurs when the result of adding two binary numbers is greater than the number of bits allowed, (eight in the example below). The maximum value that can be held in an 8-bit register is 255. A computer or a device has a predefined limit that it can represent or store, for example 32-bits. Once a value falls outside this limit, an overflow error is returned.

Consider the following example to illustrate this:

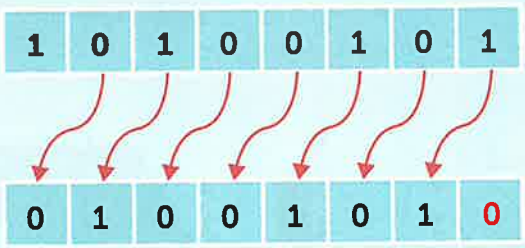
1						1	1		Check
	1	1	0	0	0	0	0	1	193
+	1	0	0	0	1	0	1	1	139
1	0	1	0	0	1	1	0	0	332

1. Add the following binary numbers, leaving the answer as binary numbers.
   (a) 0011 1011 + 1000 0110 [1]
   (b) 1001 1100 + 0111 1110 [1]
2. Explain the problem that would occur in part 1(b) if the result was to be stored as an 8-bit number. [2]

LOGICAL BINARY SHIFTS

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

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

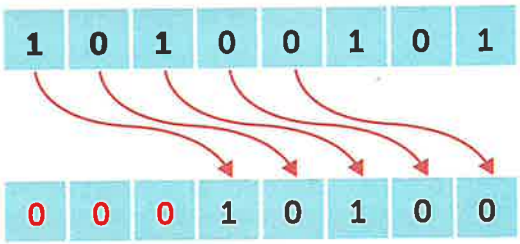


The most significant bit is lost in a left shift, and the least significant bit is lost in a right shift.

Effects of logical shifts

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

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



The original binary value was equal to decimal 165. A right shift should divide this by 8 (or divide by 2, three times). 165 / 8 = 20.625. However, the resulting binary converted to decimal is 20.

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

TWO'S COMPLEMENT SIGNED INTEGERS

An **unsigned** representation of a binary number can only represent positive numbers. A **signed** integer can represent both positive and negative numbers.

Two's complement

Using two's complement, the leftmost bit is the **sign bit**. If it is 1, the number is negative. If it is 0, the number is positive.

Two's complement works in a similar way to numbers on an analogue counter. When it shows 0000, moving the wheel forwards by one, would create a reading of 0001. Moving it backwards from 0000 would create a reading of 9999, which is interpreted as –1.

1111 1101	=	–3
1111 1110	=	–2
1111 1111	=	–1
0000 0000	=	0
0000 0001	=	1
0000 0010	=	2
0000 0011	=	3

In an 8-bit byte, the leftmost bit represents = –128

The maximum range that can be represented with 8 bits is –128 to 127 because the leftmost bit is used as the sign bit, with a value of –128, leaving only 7 bits to represent the positive part of the number. A 1 as the leftmost bit indicates the number is negative.

Converting a negative denary number to binary

Work out the positive binary equivalent of the number, flip all of the bits and add 1.

For example:

Target: –21
Positive 21: 0001 0101
Flip the bits: 1110 1010
Add one: 1
Convert: 1110 1011

Converting a negative two's complement binary number to denary

The same method works the other way. Flip all of the bits and add 1. Then work out the denary equivalent as normal.

For example:

Target: 1110 1110
Flip the bits: 0001 0001
Add one: 1
Convert: -0001 0010 = –18

1. Convert the following denary numbers to signed binary integers:
   (a) –128 [1] (b) –57 [1] (c) –9 [1]
2. Convert the following signed binary integers to denary:
   (a) 1000 0011 [1] (b) 1110 0110 [1] (c) 1110 0101 [1]

REPRESENTING TEXT

Each **character** on the keyboard has a binary code which is transmitted to the computer each time a key is pressed. Some of the characters and their codes, known as the **character set**, for the standard keyboard are given below. The **ASCII** character set consists of 128 characters, each using 7 bits to uniquely represent them. ASCII stands for American Standard Code for Information Interchange.

ASCII	DEC	Binary	ASCII	DEC	Binary	ASCII	DEC	Binary	ASCII	DEC	Binary
NULL	000	000 0000	space	032	010 0000	@	064	100 0000	`	096	110 0000
SOH	001	000 0001	!	033	010 0001	A	065	100 0001	a	097	110 0001
STX	002	000 0010	"	034	010 0010	B	066	100 0010	b	098	110 0010
ETX	003	000 0011	#	035	010 0011	C	067	100 0011	c	099	110 0011
EOT	004	000 0100	$	036	010 0100	D	068	100 0100	d	100	110 0100
ENQ	005	000 0101	%	037	010 0101	E	069	100 0101	e	101	110 0101
ACK	006	000 0110	&	038	010 0110	F	070	100 0110	f	102	110 0110
BEL	007	000 0111	'	039	010 0111	G	071	100 0111	g	103	110 0111
BS	008	000 1000	(	040	010 1000	H	072	100 1000	h	104	110 1000

Using the ASCII table in programming

The character codes are grouped and run in sequence; i.e. if a capital 'A' is 65 then 'B' must be 66 and so on. The pattern applies to other groupings such as lowercase characters and digits. For example, '1' is 49, so '5' must be 53. Also, '3' < '4' and 'a' < 'b'.

Notice that the ASCII code value for '7' (011 0111) is different from the pure binary value for 7 (000 0111). This is why you can't calculate with numbers that have been input as strings.

Character sets

A character set consists of all the letters, numbers and special characters that can be recognised by a computer system. The ASCII character set uses 7 bits and consists of 128 characters as shown in the table above.

Extended ASCII uses 8 bits rather than 7. This allows up to 256 characters to be represented. Additional characters in the set include symbols, common foreign language characters and mathematical characters for example, ©, £, é and ¼. 7-bit codes translate into 8-bit codes directly using an additional 0 as the most significant (leftmost) bit. For example, 'a' translates from 110 0001 to 0110 0001.

Unicode uses 16 bits per character, and can represent 65,536 different characters. This is enough to represent most of the characters in international languages including those in Russian, Chinese, Arabic and emojis 😊.

Use the ASCII table for this question.

1. Show how the word CAGE is represented in ASCII. Give your answer in binary. [1]
2. State how many bytes would be used to store the phrase "BIRD CAGE" using extended 8-bit ASCII. [1]

REPRESENTING SOUND

Analogue sounds must be digitally recorded in binary. In order to record sound, the **amplitude** or height of the soundwave emitted must be measured and recorded (**sampled**) at regular intervals. How often the height is recorded (the **sample rate** or **frequency**), and the accuracy to which the height is recorded (the **bit depth** or **sample resolution**) affect the quality of the recorded sound when played back and the file size of the recording. The **duration** of the recording will also affect the file size.

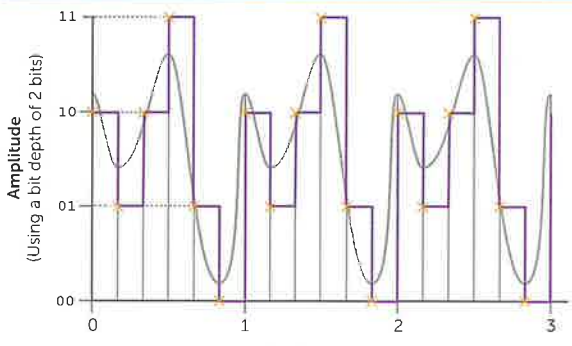
The **sample rate** is measured in **hertz**. CD quality playback is recorded at 44.1 kHz.

Examples A and B show how the digitally represented wave more accurately follows the analogue sound wave form with a greater **bit depth**.

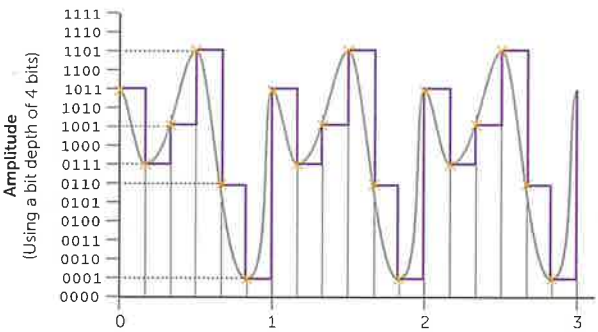
Examples B and C show how waves recorded at identical sample resolutions are much more accurately represented with a greater **sample rate**.

More samples and longer duration increases file size.

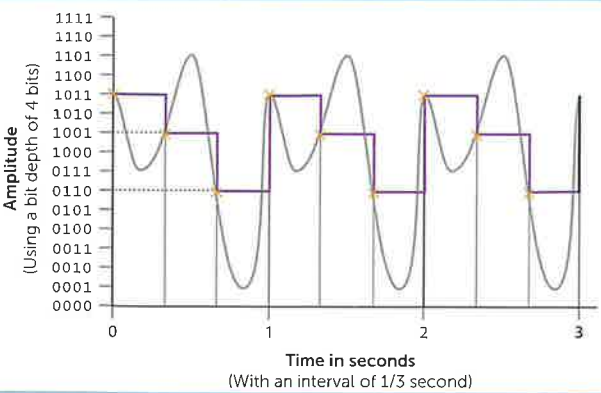
A



B



C



Look at Examples A, B and C.

(a) Give the binary representation for the first six samples taken in the first second of recording in Example A. [2]

(b) State how many different amplitudes or wave heights could be recorded if the sample resolution was 8. [1]

(c) State the sample rate in hertz of Example C. [1]

(d) Calculate the file size in bytes of a message alert tone lasting 3 seconds, using a sample rate of 8 kHz and an 8-bit sample resolution. [1]

REPRESENTING IMAGES

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

Colour depth

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

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

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

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

Image resolution

The resolution of an image is the number of pixels in the image, for example 600 x 400px = 240 000px.

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

1. Study the bitmap images above.

   (a) Give the binary representation for the top row of the 4-colour example. [2]

   (b) State the colour depth of an image if a palette of 256 colours per pixel is required. [1]

   The number of available colours in the 4 x 4 pixel image above is increased to 256.

   (c) State the effect on the file size of the image. [1]

Effect of colour depth

As the number of bits per pixel increases (the **colour depth** or **bit depth**), so does the quality of the image as you are able to more accurately represent the full range of colours visible to the naked eye. However, this increases the **file size**.

2 colours *4 colours* *8 colours* *16 colours* *256 colours* *65,536 colours* *16.7m colours*

The file size of an image in bits can be calculated as *width in pixels x height in pixels x colour depth*. Dividing by 8 will give the size in bytes.

Simply increasing the number of pixels in an image will also increase its size. An 8 x 8 pixel icon will have a file size four times larger than a 4 x 4 pixel icon with the same colour depth.

The density of pixels in the same sized area affects the sharpness of an image. More pixels per inch (**PPI**) will smooth the edges and improve the overall quality. This will increase the size of the image file, making it possible to enlarge the image without a visible loss of quality.

2. An image has 1024 x 1024 pixels and a colour depth of 24 bits.

   (a) State the file size in MiB. [2]

   (b) Write an expression to show how many colours are available for each pixel if an image has a colour depth of 24 bits. [1]

DATA COMPRESSION

Compression software uses **algorithms** to remove repeated or unnecessary data. This reduces the size of a file on a disk, and of large files sent by email where there are maximum attachment limits. It is also useful for streaming data over the Internet as it would require less bandwidth. Compressed images and video embedded within websites will also be transmitted in less time than uncompressed versions.

Type	Lossy compression	Lossless compression
Formats	JPG, MP3, WMV, MPG	TIF, PDF, GIF, PNG, ZIP
Examples		
Advantages	Smallest file sizes, least transmission time, reduces Internet traffic and collisions	Original quality is preserved / no information or data is lost
Disadvantages	Detail is permanently lost by reducing the colour depth, resolution or sample rate	Less significant reduction in file size
Example uses	Music streaming, online images and video, image libraries on devices or in the cloud	Text documents, electronic books, high resolution print documents

1. A large software program is being distributed via an online download.
   (a) Give **two** advantages of using compression software for online downloads. [2]
   (b) Explain which type of compression should be used to compress the software. [2]

Lossless compression with Run length encoding (RLE)

Data, particularly image data, can be represented using frequency/data pairs. For example, a black and white 5x5 image with a colour depth of 1 may be represented by the binary string: 00000111000001111100. This could be represented saying "five zeros, three ones" and so on or using **RLE** as 50 31 50 51 20.

2. Write out the bit pattern for the following RLE encoded data:
   40 31 30 21 [2]

EXAMINATION PRACTICE

1. Construct an expression to show how many bytes there are in a 4.5 KiB text document. [1]

2. The ASCII character 'F' is represented in binary as 100 0110.
   State the binary value for the character 'C'. [2]

3. A digital thermometer is used to measure the temperature inside a warehouse.
   (a) The temperature value will be displayed to the nearest whole number using one byte. The historic minimum and maximum temperature values in the area have been −14°C and 42°C.
      (i) State whether a signed or unsigned integer should be used. [1]
      (ii) Explain why eight bits provides sufficient storage space for this value. [2]
   (b) Tick (✓) **one** box to show which of the values A, B, C or D represents the temperature value −6 in two's complement binary. [1]
      A. ☐ 0000 0110
      B. ☐ 1000 0110
      C. ☐ 1111 1001
      D. ☐ 1111 1010

4. (a) (i) State the maximum positive binary number that can be held in an 8-bit two's complement number. [1]
      (ii) Convert this number to denary. [1]
   (b) Convert the two's complement binary numbers 0011 1010 and 0101 1010 to denary. [2]
   (c) Add the two's complement binary numbers 0011 1010 and 0101 1010. [1]
   (d) Explain the result. [2]
   (e) Convert the two's complement binary number 1111 0011 to denary. [1]

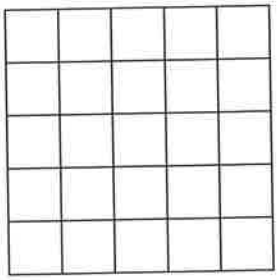
5. (a) Convert the 8-bit binary number 0000 1001 into denary. [1]
   (b) Perform a logical right shift of one place on the binary number in part (a) and convert the result to denary. Explain the result. [3]

6. (a) State the binary result after a logical right shift of four places on the bit pattern 0101 1011. [1]
   (b) State the result after a logical left shift of four places on the resulting pattern. [1]
   (c) Explain the result of these two consecutive shifts. [1]

7. Numbers larger than 255 are represented in binary using more than 8 bits.
   (a) State the value of the following 16-bit binary value in denary: 01000011 01101001 [2]
   (b) State the value of the following denary value in binary: 874 [2]

8. A bitmap image with a colour depth of 1 bit has been created.

(a) Convert the following binary data into an image with a resolution of 5 x 5 pixels, where 0 represents black and 1 represents white: 00100 10101 10001 00000 10101 [2]



(b) Describe an appropriate compression method for an image designed for use on a website. Full colour images are represented using 8 bits for each of the red, green and blue values that make up every colour. [4]

(c) Construct an expression to calculate the file size of the image above in bytes if the colour depth is increased to 24 bits. You do not have to do the calculation. [1]

(d) Using 24-bit colour, the colour purple can be represented by the hexadecimal code #902EA6.

(i) Complete the conversion table below to find the green and blue values in binary. [2]

	Red		Green		Blue	
Hex	9	0	2	E	A	6
Binary	1001 0000					

(ii) Explain why hexadecimal is commonly used to represent colour values. [2]

9. A sound recording is made of a short piece of music. The quality of the playback is poor. Two factors which may account for this are sample resolution and sample rate.

(a) Explain the difference between sample resolution and sample rate. [2]

(b) Explain how sample resolution could affect the quality of a sound recording. [2]

(c) Describe **one** factor besides sample resolution and sample rate which will affect the file size of a sound recording. [1]

DATA PACKETS

Just like some large shipments, data is broken down into smaller chunks called **packets**. Each is labelled, for example, packet 1 of 5 if it the first of five packets to be transmitted in the sequence.

Structure of a data packet

Each data packet contains:

Packet header

The **packet header** includes the **destination IP address**, the **packet number** and the **sender's address**.

Payload

The **payload** is the actual chunk of data being transmitted, e.g. 512 bytes.

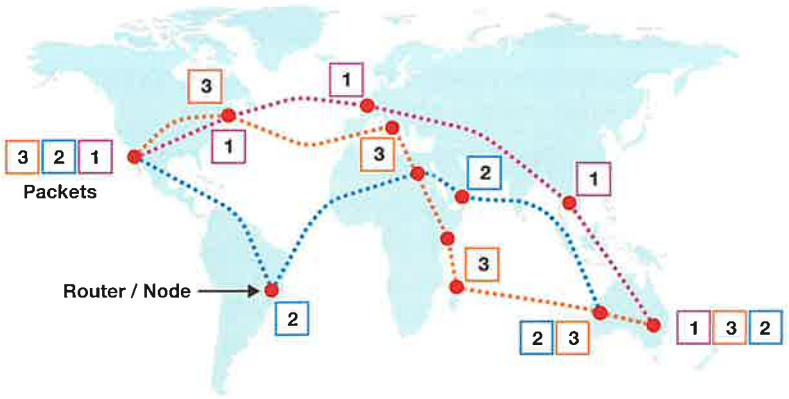
Trailer

The **trailer** marks the **end** of the packet.

Packet switching

Packet switching controls the sending and receiving of packets across a network or the Internet.

1. Data is broken down into packets.
2. Each packet is given a sequence number and sent on its way.
3. Each packet may take a different route to the next depending on the quickest route at the time.
4. Routers control the route that a packet takes.
5. Packets may arrive out of order. Once the last packet has arrived, the sequence numbers are used to put them back into order. The data can then be reconstructed correctly.

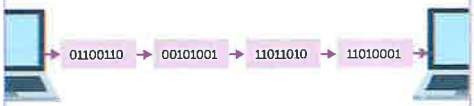


DATA TRANSMISSION

There are different methods that can be used to transmit data from one device to another.

Serial vs parallel transmission

Serial transmission



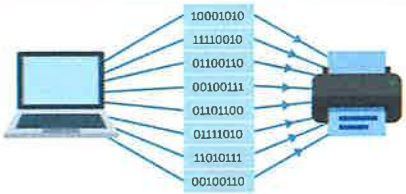
Data bits are sent one at a time down one cable.

Very high data speeds can be achieved.

Used with the USB (Universal Serial Bus) interface.

- Lower transmission speeds
- Lower interference.
- More reliable over longer distances.
- Smaller, simpler and cheaper connections to produce.

Parallel transmission



Several data bits are sent simultaneously.

Faster than serial transmission but can only be used over very short distances.

Used with printer cables or internal device connections.

- Interference issues between wires.
- Data corruption can occur over longer distances.
- Data bits travelling parallel to each other can move at different speeds causing them to arrive out of sync. This is called **skewing**.
- Very fast data transfer.
- Can send data in both directions at the same time

USB (Universal Serial Bus) interfaces

The USB interface is an industry standard way to transfer data between all sorts of devices. It is limited to about 5 metre cable length without a hub.

1. Give **three** reasons why a device manufacturer may choose to use a USB connection. [3]

Simplex, half-duplex and full-duplex transmission

Simplex, duplex and half-duplex are terms referring to the direction of data during transmission. These terms can be used in conjunction with serial and parallel transmission terminology.

Simplex transmission

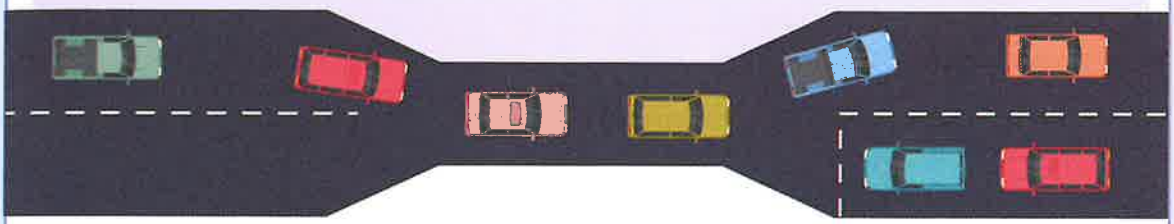
Data travels in one direction only down a single cable.

Example: Used for sending data from the computer to a monitor. It never needs to go the other way.

Half-duplex transmission

Data can travel in both directions along a single cable, but not simultaneously.

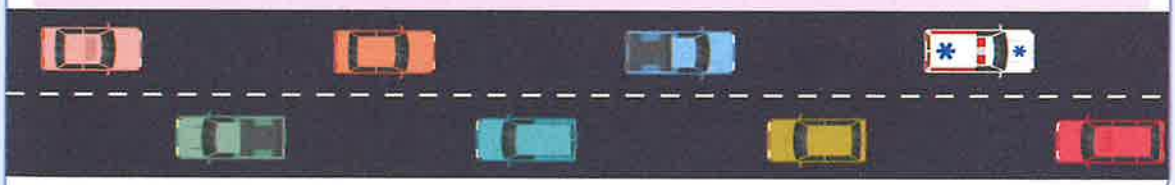
Example: Used in parallel printer cables: text is sent to a printer, and the printer can send 'Out of paper' messages back, but only when the computer has finished sending text.



Duplex transmission

Data can travel in both directions simultaneously using two communication channels.

Example: Commonly used in networking or Internet cables to transmit and receive at the same time.



2. Three applications of data transmission are given below. Draw a line to link each application to the most appropriate method.

Transmitting sound data between a microphone and the computer	Simplex
Transmitting sound data between two video phone users	Half-duplex
Transmitting sound between two walkie talkies	Duplex [3]

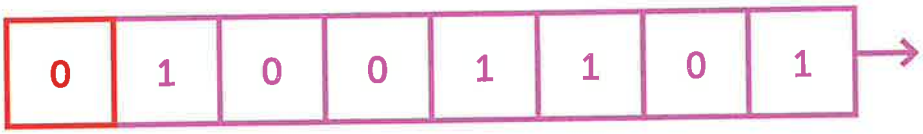
TRANSMISSION ERROR DETECTION

During data transmission, electrical interference can cause data to be lost, gained or changed along the way. This alters the pattern of 0s and 1s that are sent, affecting the integrity (or correctness) of the data.

Parity checks

Computers can use either odd or even **parity**. When sending a byte of data, one bit is used as the parity bit. This parity bit is set to either a 0 or a 1 at the start of transmission to make the total number of 1s odd or even, depending on the machine. The 1s are counted at the receiving end and if the wrong number of bits are 'on' or 1, an error is detected, and the data is resent.

This example uses even parity, so the parity bit is set to 0.



Parity bit

Parity block check

A **parity block check** is used when sending multiple bytes in one transmission block. Each byte will have its own parity bit. A final parity byte is used which contains a parity bit for each column.

An error in the transmission can then not only be detected but the exact position can be identified and corrected.

1. A computer uses odd parity.

   State the value of the parity bit that should be added to the following byte of data. [1]

?	1	1	0	1	0	1	0

2. One of the following four bytes of data has been transmitted incorrectly.

Byte 1	Byte 2	Byte 3	Byte 4
10110110	11101100	10111011	11011010

   (a) Identify which byte has had an error. [1]

   (b) Explain how you identified which byte contained an error. [3]

   (c) Give **one** example of how a parity check could allow an erroneous byte to pass undetected. [2]

Checksums

A **checksum** is calculated using an algorithm to sum all of the data to be sent in a transmission. The checksum is sent with the transmission and recalculated at the end using the same method. If the start and end sums do not match, the data has changed during transmission.

A very simple example

There are many different algorithms for calculating a checksum. The following is an example of a very simple algorithm.

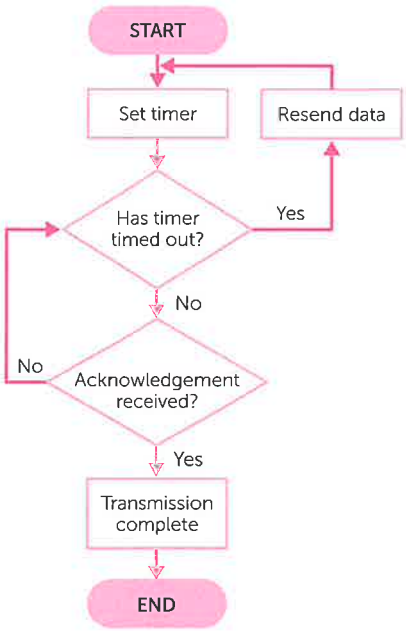
The number of bytes in a transmission is counted as 4 and added to the total number of 1s (19). The number 23 (converted to binary) is then added to the transmission. The calculation is repeated at the receiving end and if the two checksums do not match, an error has been detected. The data can then be resent.

Echo check

An **echo check** is performed by asking the recipient computer to send back an exact copy of the data it has just received to the originator. The data is compared with what was originally sent. If both copies match, the data is deemed to have arrived without error.

Automatic Repeat Request (ARQ)

Computers using **ARQs** will automatically send back a positive **acknowledgement** if a **data packet** is deemed correct or they will ask for the data to be resent if an error has been detected. If no acknowledgement is returned within a certain time period, a **timeout** is registered and the sending computer resends the packet. This process carries on until the packet received is correct or until the ARQ limit is reached.



CHECK DIGITS

A check digit check is a form of validating data entry rather than detecting errors in transmission.

The check digit is an additional digital at the end of a string of other numbers. It is commonly used with barcodes and ISBN numbers. A check digit is recalculated on data entry. If this matches the one at the end of the number, the numbers are deemed to be **correct** and in the **right order**.

Calculating a check digit

Using the **Modulus 11** system, weights of 2, 3, 4,...etc are assigned to each digit starting from the right. To assign a check digit to the number 45305 to form a valid product code:

Product code	4	5	3	0	5	
Weight	6	5	4	3	2	
Multiplication	24	25	12	0	10	
Addition	*Add all the numbers*					71
Remainder	*Remainder when sum divided by 11*					5
Subtraction	*Subtract the remainder from 11*					**6**

The check digit is 6. The full product code, including the check digit, is therefore 453056

If the result of subtracting the remainder from 11 is 10, the check digit is X. (X = Roman 10.)

Checking the check digit

The check digit on a product code may be checked as follows:

- Each digit is multiplied by a 'weight' of 1, 2, 3... starting with 1 at the right-hand end.
- Each result is added together to form a total.
- The total is divided by 11. If the remainder is 0, the number is valid.

(a) State whether the barcode number 6 7 1 0 2 4 has been read correctly or incorrectly. Show your working. [3]

(b) Calculate the check digit for the following product number. 7 4 2 0 3 _ [2]

(c) Suggest **one** way in which an error could occur in reading a product barcode. [1]

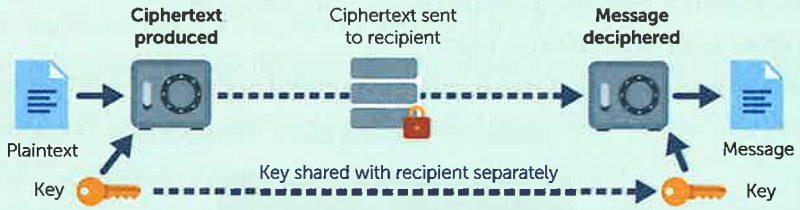
ENCRYPTION

Encryption is the process of encoding data so that it cannot be easily understood if it is discovered, stolen or intercepted.

An original message or data file is known as **plaintext**. This is converted into **ciphertext** by encoding it using a mathematical **encryption algorithm** and **key**. Both a key and the algorithm are required to encode or decode data.

Symmetric encryption

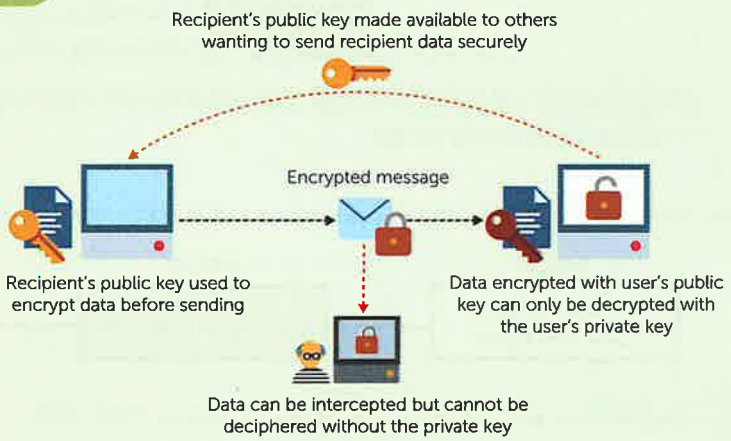
Symmetric encryption (**private key** encryption) uses the same pre-shared key to encode and decode data. This means that the key must also be transferred (separately) to the recipient so that they can decrypt it.



Asymmetric encryption

Asymmetric encryption (**public key** encryption) uses two separate but related keys. A public key is shared with others and used by them to encrypt a message so it can only be deciphered by the intended recipient with their private key.

A private key must never be shared.



1. Explain why the key is sent separately from the encoded data when using symmetric encryption. [3]
2. Jonny says that he has encrypted his work so that no one else can access it.
   Explain why this isn't strictly true. [2]

EXAMINATION PRACTICE

1. When data is sent from one computer to another across the Internet, the data is split up into packets.

   (a) Tick (✓) **one** box to show which statement is true of packet switching. [1]

   A. ☐ Data is lost when it is broken into packets.

   B. ☐ Packets are sent by different routes depending on which is fastest at the time.

   C. ☐ Packets are sent in numerical order so they can be reassembled more quickly.

   D. ☐ Packets require one specific route to travel through from sender to receiver.

   (b) Give **four** items that would be included in each packet in addition to the data. [4]

2. Data may be transmitted using either serial or parallel transmission.

   (a) Explain the difference between these two forms of transmission. [2]

   (b) Give **two** advantages of serial transmission. [2]

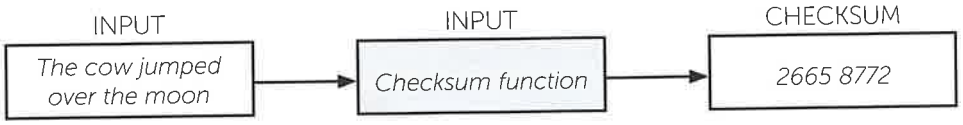
   (c) Rashid has a desktop computer with a printer connected to it using a short parallel cable.

   (i) Use a tick (✓) to identify the most suitable data transmission type for this application. [1]

Method	Tick (✓)
Simplex	
Half-duplex	
Duplex	

   (ii) Explain why your answer to part (c)(i) is the most suitable method of data transmission for sending data to the printer. [2]

3. The sentence "*The cow jumped over the moon*" is transmitted along with a checksum.



   The checksum is recalculated at the receiving end as 3518 5003.

   Explain what action should be taken. [2]

4. A company selling sportswear online uses a unique 6-digit product number for each item.
The rightmost digit is a **check digit**.
For example:

6 5 4 3 2 1 (digit position)
4 8 3 1 6 8 (product number)

A check is carried out to ensure the product code is valid. The calculation is as follows:

- multiply each digit by the digit position, starting with 1 at the righthand end
- add the results of the multiplications
- divide the answer by 11
- if the remainder is 0 the product number and the check digit are valid.

(a) State whether the following product number is valid or invalid. Show your working. [3]
Product number: 3 1 5 4 6 9

(b) Calculate the check digit for the product number 2 6 1 8 5 _ [3]

(c) Describe, with examples, **two** different types of data entry error that a check digit would detect. [4]

5. Encryption is used by organisations, email systems and banks.

(a) Explain why sensitive data is encrypted before being transmitted. [2]

(b) Using a simple Caesar cipher, each letter is replaced by one further along the alphabet.
Using a shift key of 1, the word FUN may be encrypted as GVO, where F + 1 = G and so on.
Using a shift key of 3, write the word FLEA in ciphertext. [1]

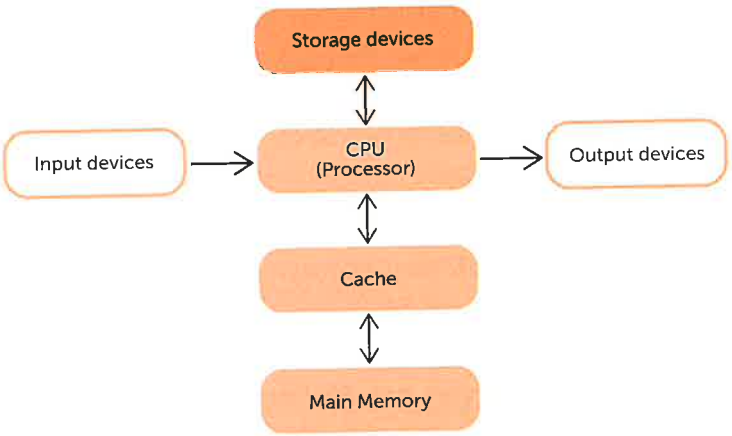
(c) Give the meaning of the terms **plaintext** and **ciphertext**. [2]

(d) Explain the difference between **symmetric** and **asymmetric** encryption.
Include in your explanation the reason why asymmetric encryption is more secure than symmetric encryption. [4]

COMPUTER ARCHITECTURE

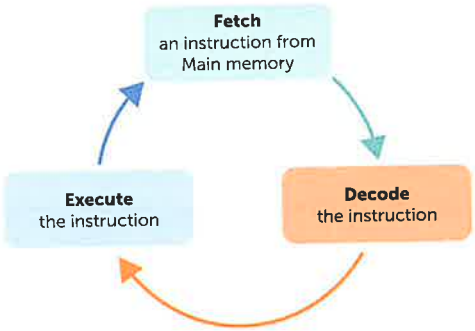
The purpose of the CPU

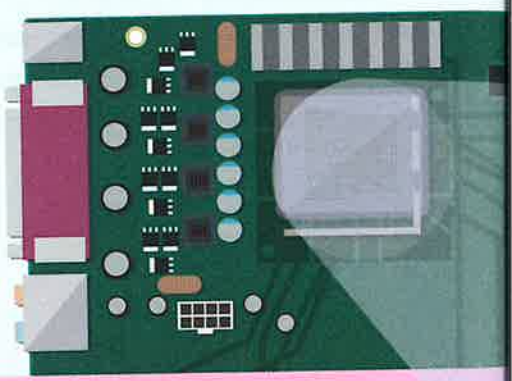
The purpose of the **Central Processing Unit** (**CPU**) is to continuously process instructions and data that are input by repeatedly carrying out the **fetch-execute cycle** in order to output a result. The CPU contains the **Arithmetic Logic Unit** and the **Control Unit**, in addition to several general-purpose and special-purpose **registers**.



The fetch-execute cycle

Every CPU instruction is fetched from memory. Once fetched, it is decoded by the Control Unit to find out what to do with it. Then the instruction is executed. Every operation carried out within the fetch-execute cycle is regulated by a 'tick' or cycle of the CPU clock.





A single core 4.5 GHz processor has 4,500,000,000 clock cycles or 'ticks' a second. This is known as the clock speed.

Von Neumann architecture

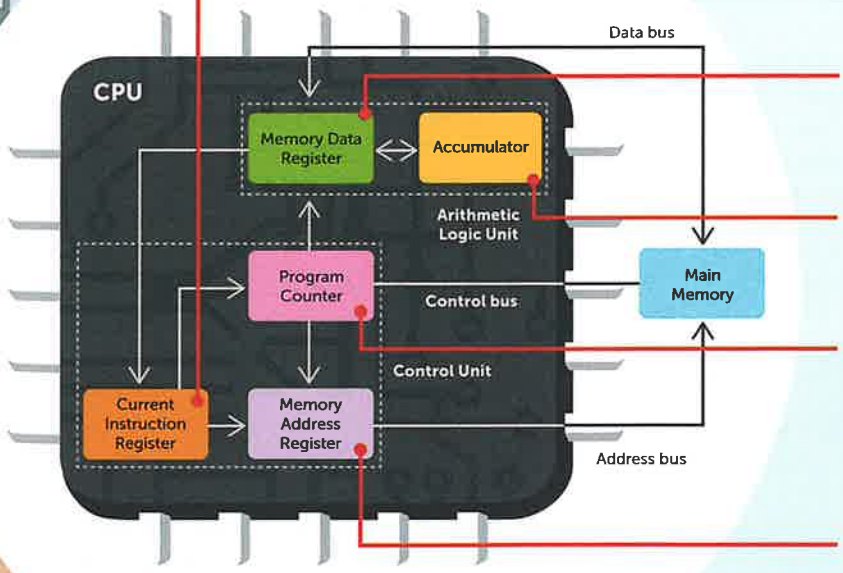
John von Neumann developed the **stored program computer**. In a von Neumann computer, the programs and the data they use are stored in the same memory.

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

A microprocessor is a type of integrated circuit on a single chip that processes instructions.

Identify **two** events that happen during the fetch-decode-execute cycle. [2]

The **current instruction register** (**CIR**) holds the instruction that is currently being executed.



MDR holds data or a program instruction when it is fetched from memory or data that is waiting to be written to memory.

The **accumulator** (**ACC**) is a register in which results of operations carried out in the **ALU** are stored.

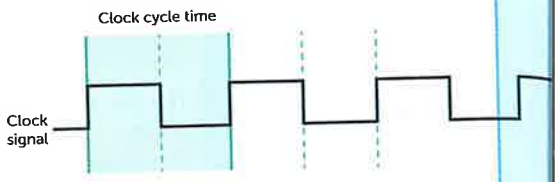
PC is a register which holds the **memory address** of the next instruction to be processed.

MAR holds the address (location in memory) of the current instruction or piece of data to be fetched or stored.

CPU PERFORMANCE

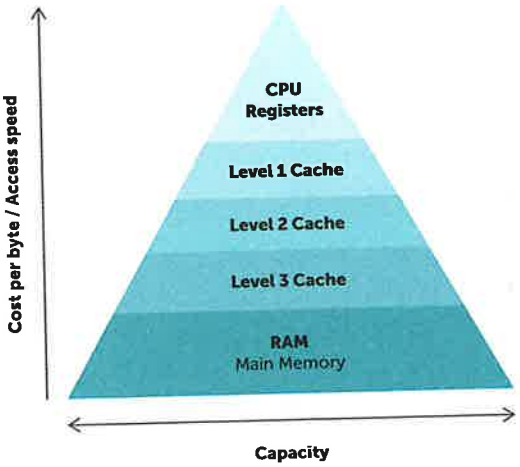
Clock speed

The **clock speed** determines the number of **fetch-execute cycles** per second.
Every action taking place in the CPU takes place on a tick of the clock, or clock cycle. Each cycle is one **hertz** so a 3.7 GHz processor will cycle at 3.7 billion times per second.



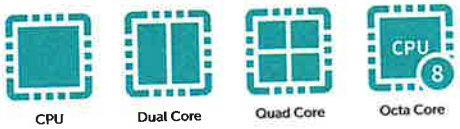
Cache size

Since **cache memory** operates much faster than main memory, data is transferred in and out of cache memory more quickly, which makes the CPU more efficient as less time is spent waiting for data to be transferred. There are two or three levels of cache. The fastest cache with the smallest capacity is Level 1 cache. The CPU will optimise its use of the fastest cache before using the next level, or using **Random Access Memory** (**RAM**), in order to improve performance speed.



Number of cores

A processor may contain more than one **core**. Each core can process one operation per clock cycle. A dual- or quad-core processor will be able to perform two or four operations simultaneously (for example, run two programs simultaneously), but only if the software it is running is designed for multi-core processors.



Amy's computer has a 4.5 GHz, dual core processor.

(a) How many operations is a 4.5 GHz, dual core processor theoretically able to perform each second? [1]

(b) Explain why a computer with a dual core processor may not be twice as fast as a single core processor with the same clock speed. [2]

3.1.4

INSTRUCTION SETS

Processors execute machine code, the binary instructions which the computer can respond to directly to perform a task.

Each type of processor has its own specific list of commands written in machine code that can be understood and executed by the CPU. This is called an **instruction set**. Machine code produced for one type of computer will not work on another.

A simple instruction set may look like this:

0000	LDA	Load into the accumulator the value stored in a specified memory location
0001	STORE	Store the value in the accumulator in a specified memory location
0010	ADD	Add a specified value to the value in the accumulator
0011	HLT	Stop

Assembly language instructions can be represented using **mnemonics** which make them easier for humans to understand

Using the sample instruction set opposite, the following instructions below would load into the accumulator the value 5, stored in memory location 6. The value 2 is added to the accumulator to get 7. It then stores the total 7 in location 3 and halts:

0000 0110
0010 0010
0001 0011
0011 0000

3.1.5

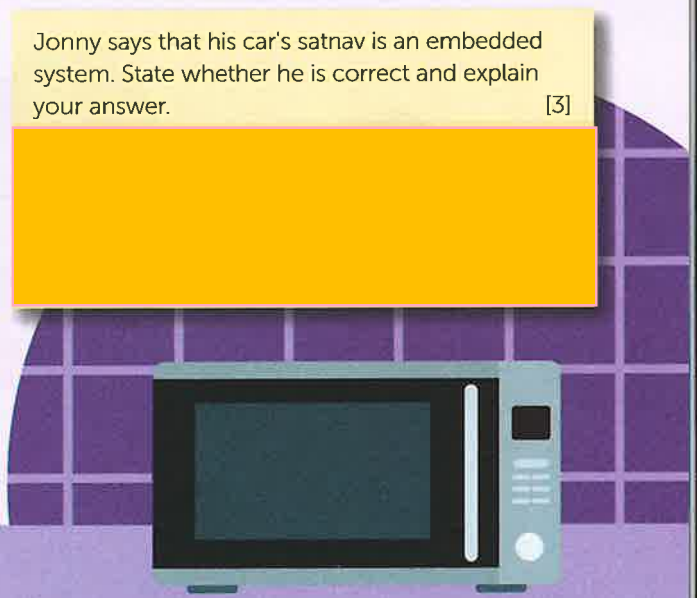
EMBEDDED SYSTEMS

An **embedded system** is used to control the function of electronic devices such as those commonly found in the home. They often don't need a full operating system since they perform limited and very specific tasks with their input frequently controlled by a button press or switch.

Embedded systems must be reliable since they cannot be modified once manufactured. The program that controls them is held in **Read Only Memory** (**ROM**).

Examples include air conditioning or heating systems, security systems, vending machines, cars, microwave ovens and lighting systems.

Jonny says that his car's satnav is an embedded system. State whether he is correct and explain your answer. [3]



SCANNERS

Scanners are **input devices** that all use reflected light to recognise a pattern or shape which a computer can convert into meaningful data.

Barcode scanners

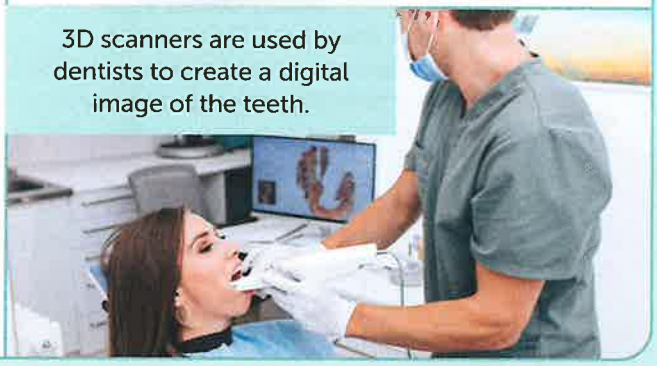
Red laser **light** is shone at a barcode to **illuminate** it. The light from white lines is **reflected** back again. Black lines absorb the light so less is reflected. Sensors within the scanner detect the **pattern** of strong and weak reflections from the lines. The reflection strength from each line is interpreted by a **microprocessor** and converted into a **binary value**.

1. A barcode system is used by a parcel delivery company. Explain how barcodes could be used to track and update the delivery status of a parcel. [5]

2D and 3D scanners

A two-dimensional (or flatbed) scanner shines a strip of light onto a document or photograph. The strip moves over the whole document and as it does, the reflected light is captured using mirrors and lenses and the light intensity is measured for each pixel. This information is converted into a digital file to produce a digital image of the original document.

A 3D laser scanner shines a light over the surface of an object. It records the **geometry** and **dimensions** of the object to produce a digital file. From this file, a digital model of the original object can be created using specialist software. Parts could then be printed using a 3D printer.



3D scanners are used by dentists to create a digital image of the teeth.

QR code scanners

A QR code is read by using a **camera**, commonly on a **mobile device**. Three large squares in the corners of the QR code are used for **alignment**. The smaller black and white squares then reflect or absorb light in the same way as a normal barcode. Software on the mobile device then converts each pixel into a binary value to process the image. A QR code commonly contains a hyperlink to an online file, some text or a hyperlink to a webpage containing further information.

2. A restaurant is seeking to provide digital menus for customers to read on their smartphones when they scan a code at the table.

   Justify which type of code would be most suitable to use. [3]

3.2.1

DIGITAL CAMERAS

A pixel is the smallest identifiable area of an image.

Function and use

A digital camera contains a **CMOS** or **CCD** sensor. As light enters the **lens**, it falls onto the sensor which is divided up into millions of individual squares or **pixels**. Each square measures the **light intensity** that falls on it and this value is converted into digital data as a pixel with a single **binary colour value**. An image can then be stored as a file on a flash memory card or reassembled on a screen by a **microprocessor** within the camera using the information from each pixel.

Smartphones commonly have an integrated digital camera. Give **one** advantage and **one** disadvantage of using this compared to a dedicated digital camera. [2]

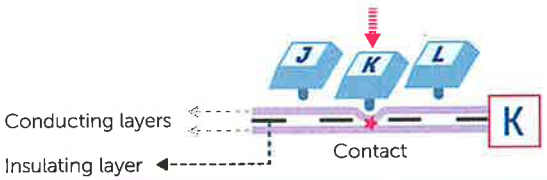
3.2.1

KEYBOARDS, MICE AND MICROPHONES

Keyboards, mice and microphones are input devices.

Keyboards

When a key is pressed on a standard keyboard, a peg under the key is pressed into a springy layer to make contact with a conducting membrane.



Optical mice

An optical mouse uses a red LED and a sensor to determine the movement of the mouse relative to a surface. The speed and distance of movement is then analysed by a digital signal processor to create the same movement of the mouse cursor.

Microphones

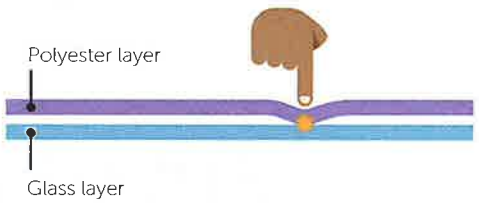
A microphone uses a diaphragm to vibrate in response to sound. The vibrations cause a coil to move around a magnet which creates changes in the electromagnetic field. Changes are detected and recorded by a processor using an analogue to digital convertor (ADC).

TOUCH SCREENS

A touch screen is both an **input device** and an **output device**.

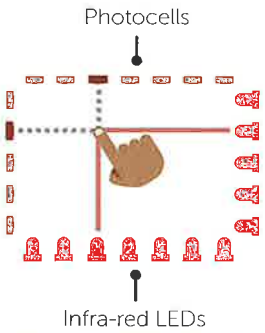
Resistive screens

Resistive touchscreens are made up of a layer of conductive polyester and a layer of conductive glass. The two layers are separated by an insulating membrane. When the screen is gently pressed by a finger, the two conducting layers make contact and complete a circuit. The position where the screen is touched is calculated by a microprocessor. Resistive screens are often used with some ATM screens and medical equipment as users commonly wear gloves.



Infra-red screens

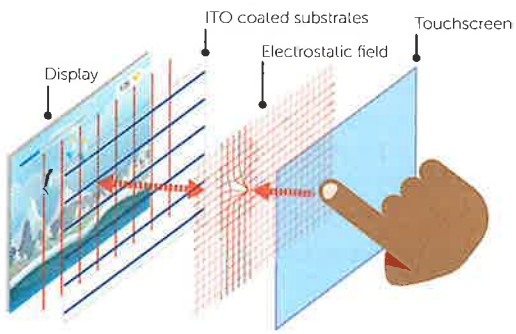
LEDs shine infra-red light across the screen from the top and sides. The infra-red rays form a grid across the screen. When the screen is touched by a finger, the infra-red beam is interrupted, and a microprocessor calculates the coordinates of where the screen was touched.



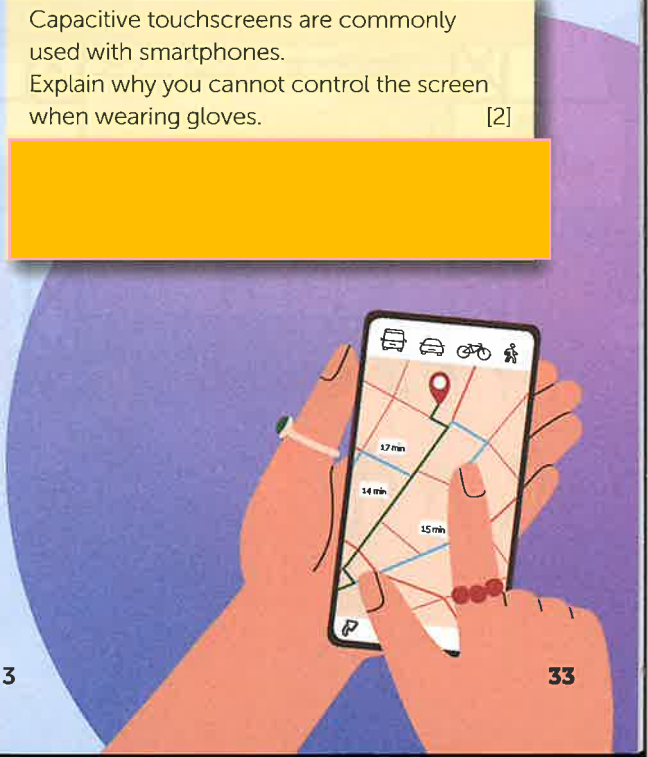
Infra-red screens are commonly used at staff tills in fast food outlets.

Capacitive screens

Capacitive touchscreens are made up of glass layers. The bottom surface of the upper layer and top surface of the bottom layer are coated in perpendicular lines of transparent conductive material forming a grid. The screen behaves like a capacitor, storing electrical energy. When the screen is touched by a finger, there is a change in the electric field at that position. The position where the field changes is calculated by a microprocessor Unlike resistive screens, capacitive screens can be touched in more than one place at a time.



Capacitive touchscreens are commonly used with smartphones.
Explain why you cannot control the screen when wearing gloves. [2]



SCREEN TYPES

Screens with no touch function are solely **output devices**.

Liquid crystal display (LCD) screens

Liquid Crystal Display (LCD) screens are made from millions of pixels. Each pixel is effectively a separate red, blue and green light that can be switched on or off electronically using liquid crystals to rotate polarised light. The pixels are switched on or off very rapidly to make a moving colour picture. A special fluorescent lamp is required to provide the necessary backlight.

Light emitting diode (LED) screens

An LED screen uses the light from LEDs to produce a brighter backlight almost instantly, replacing the need for the additional fluorescent lamp required by LCD technology.

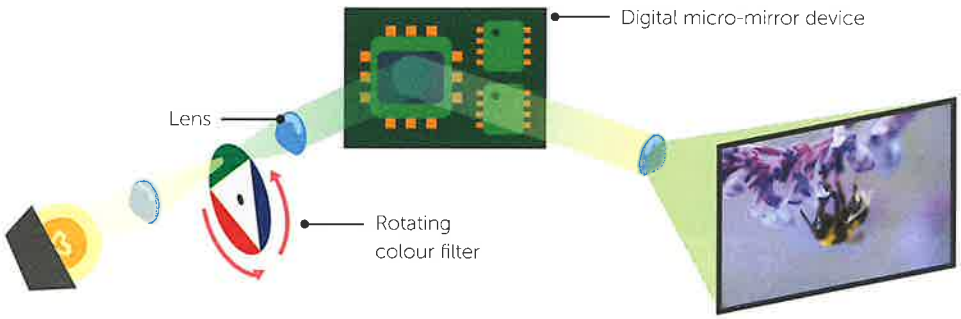
LED screens can be made to be very big for sports stadiums or concerts. These are made of many smaller screens in a cluster.

Kester is considering the purchase of a new advertising screen for a shop window. He has been told about LCD and LED screens. Discuss the advantages and disadvantages of each. [4]

PROJECTORS

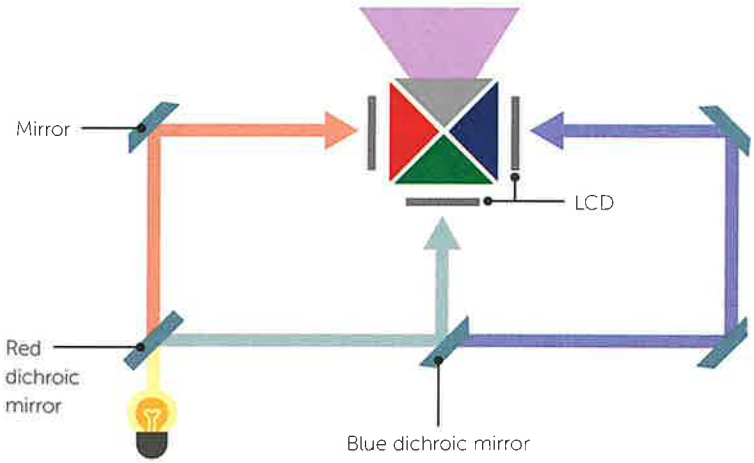
Digital light processing (DLP) projectors

DLP systems utilise millions of micro-mirrors arranged in a grid on a small microchip within the projector. Shining white light through colour filters alters the colour whilst the angle of the mirror alters the intensity of colour.



Liquid crystal display (LCD) projectors

LCD projectors use three mirror filters to separate an image into red, green and blue wavelengths. The three images are then combined using a prism to produce the full colour image which then passes through a lens and is projected on to a screen.



DLP Projectors

- Higher contrast and definition
- Smoother video
- Smaller

LCD Projectors

- Cheaper
- Quieter
- Less power and heat output

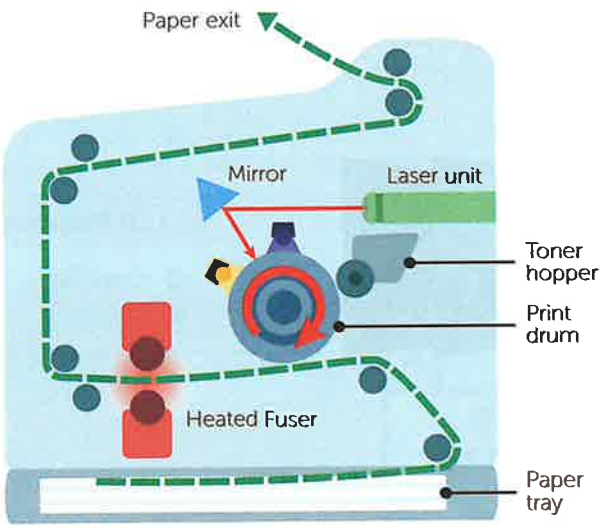
PRINTERS

Inkjet printers

Inkjet printers use **liquid ink** reservoirs, forcing droplets through a fine nozzle onto a sheet of paper. They are very good for small print runs and high-resolution colour photographs, especially when specialist coated photo paper is used.

Laser printers

Laser printers offer high-quality and inexpensive printing. They are suited to larger print jobs as they are usually faster than inkjet printers. Laser printers use **dry ink** in powder form called **toner** that is electrostatically charged. The printer generates a mirror image of the printed page and 'draws' the image onto a drum using a laser, creating negatively charged areas. The drum rotates past the toner hopper to attract positively charged ink particles. These are then transferred onto the paper and bonded using a heat roller.



Inkjet printers

- ⊕ Compact and good all round.
- ⊕ Excellent for colour photos.
- ⊖ Paper can be damp or wrinkle after printing.

Laser printers

- ⊕ Very fast and accurate for black and white documents.
- ⊕ Lower running costs per page.
- ⊖ Colour printers and toner cartridges are expensive.

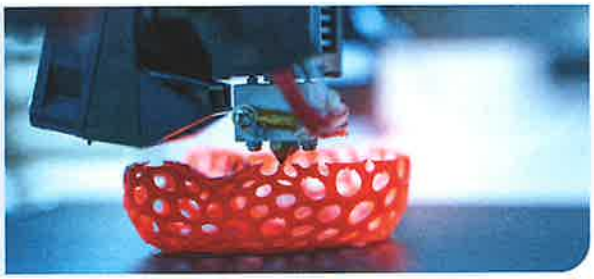
Romy is setting up a small office for a business and requires a printer for office documents and reports. Justify which type of printer she should purchase. [4]

3.2.2

3D PRINTERS

3D printers work by depositing layers of material, usually melted plastic, upon each other to gradually build up a 3D form. Other materials can include powdered metal, ceramic, paper, some foods or even cement to 'print' houses.

A **Computer Aided Design** (**CAD**) model is developed on a computer and converted to a print file as a set of instructions for the printer to follow. Unlike other production techniques for 3D forms, 3D printing enables very complicated shapes to be created with intricate internal spaces and even moving parts.

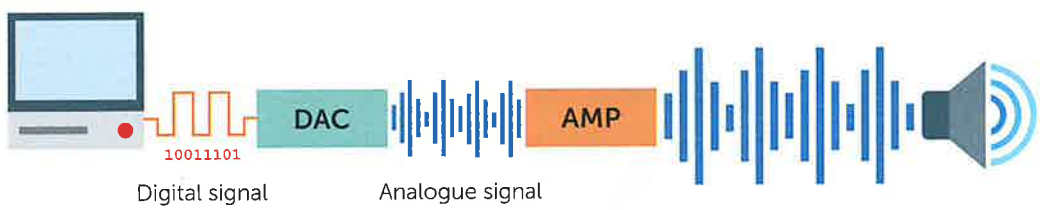


There is growing environmental pressure on manufacturers to design products that are more easily repairable. Explain how 3D printing technology could help increase repairs on home appliances. [2]

3.2.2

SPEAKERS

Speakers work in the opposite way to **microphones**. A digital signal from a computer is sent through a **digital to analogue convertor** (**DAC**) which can then be boosted by an **amplifier** and passed on to a speaker to create sound.



Explain why speakers are used at self-service supermarket tills. [2]

ACTUATORS AND SENSORS

Actuators

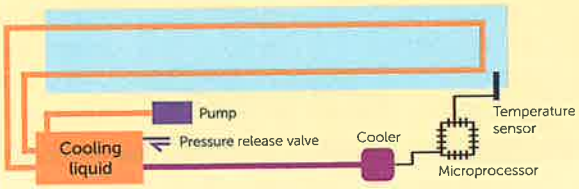
Actuators are devices that allow physical movements to be carried out, often with a motor. They are commonly used in conjunction with sensors to control a mechanism. Examples include opening a valve or door, starting a pump, turning a wheel or fan, or moving an aircraft wing flap.

Sensors

Sensors measure the physical properties of their environment. There are many types of sensor designed to detect the levels of a variety of conditions. These sensors include:

- **Acoustic** (sound levels)
- **Accelerometer** (acceleration rate, tilt, vibration)
- **Flow** (rate of gas, liquid or powder flow)
- **Gas** (presence of a gas e.g. carbon monoxide)
- **Humidity** (levels of water vapour)
- **Infra-red** (detecting motion or a heat source)
- **Level** (liquid levels)
- **Light** (light levels)
- **Magnetic field** (presence and strength)
- **Moisture** (presence and levels of moisture)
- **pH** (acidity or alkalinity)
- **Pressure** (gas, liquid or physical pressure)
- **Proximity** (distance e.g. reversing sensors)
- **Temperature** (thermistor)

The conditions of an ice rink floor are maintained using sensors and a microprocessor. To keep the floor at the perfect temperature, liquid is constantly fed through pipes within the ice. The ice temperature must be between –3°C and –5°C and the cooling liquid supply needs to be pressurised to 2.5 bar. The supply contains a pump and a pressure release valve.



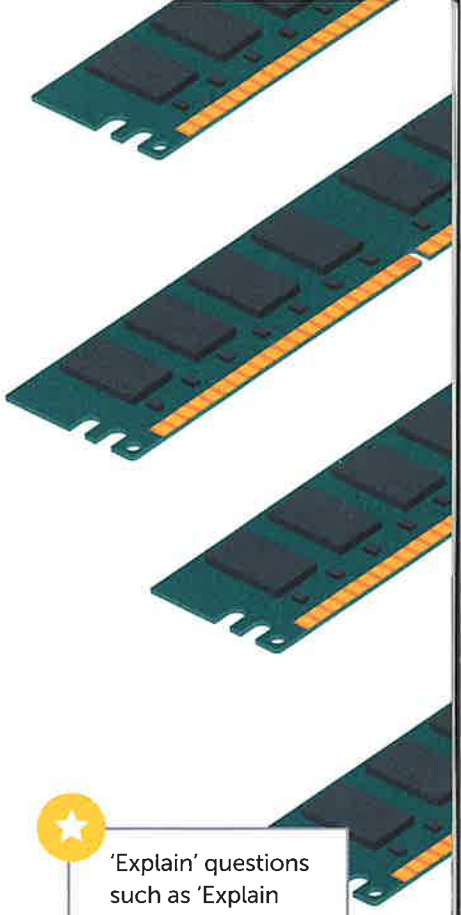
(a) Name the **two** sensors used in this context. [2]

(b) Describe how the sensors and microprocessor are used to maintain the conditions of the ice. [6]

(c) Suggest **one** safeguard that could be put in place to ensure that the temperature never exceeds –3°C. [1]

PRIMARY STORAGE

Main memory (RAM or ROM) is directly accessible by the CPU. It is also known as primary storage.



The need for primary storage

RAM (**Random Access Memory**) is required to **temporarily store** the **instructions** and **data** the computer needs whilst it is in operation. These are copied from the hard disk into main memory when they are required, because it would be too slow to access everything directly from the hard disk. For even faster access, the most frequently used program instructions and data are held in **cache**.

RAM and ROM

RAM is the computer's temporary working memory. It is **volatile** which means it gets wiped as soon as the power is switched off. **ROM** (**Read Only Memory**) stores instructions and data that never need to be changed, such as the computer's start-up instructions so that it knows what to do when you push the 'on' button. ROM is **non-volatile**. As it is read-only, you cannot overwrite its contents once they have been set by the manufacturer.

RAM	ROM
Volatile – All data is lost when the power is turned off	Non-volatile – Data is permanently retained without power
Used as the computer's working memory for instructions, programs and data	Used for the computer's start-up instructions and in embedded systems
Can be written to, and read from	Read only, so cannot be written to

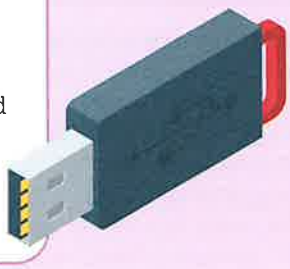
'Explain' questions such as 'Explain why this is the most appropriate...' are not just a list of benefits. You should identify the benefits but then expand on each one, whilst also applying it to the scenario or context in the question.

1. Abeel had a power cut whilst working on a spreadsheet document. He said that he lost the entire document but could still open the spreadsheet program when the power came back on.
   Explain why this was the case for Abeel. [2]
2. An increase in RAM capacity can improve the performance of your computer.
   Explain why an increase in ROM size would not have the same effect. [2]

SECONDARY STORAGE

Secondary storage is not directly accessed by the CPU. It is used for more permanent data storage.

The need for secondary storage



Secondary storage includes **hard drives** (internal and external), **USB flash drives**, **CDs** and other portable storage devices. Secondary storage is needed for longer term storage of files and data because it is non-volatile, which means data will not disappear when the power is turned off. External devices are portable and may have very large capacities.

Applications of storage media

Solid state drives (**SSDs**) require very little power and create little heat owing to the lack of moving parts. This makes them suitable for laptop and tablet devices commonly used on the go. The lack of moving parts also means they are very small and reliable – perfect for small portable devices with built-in storage such as cameras and smartphones. SSDs are also used in desktop and larger computers. They are also replacing hard disks in mass storage facilities as they can be 100 times faster than hard disks and do not require expensive cooling equipment.

Hard disk drives (**HDDs**) are cheaper and commonly found in desktop computers. SSDs may be used as a sole alternative or in conjunction with HDDs to store the operating system and other software that needs to execute as fast as possible. **CDs** and **DVDs** are useful for music and video data in the short to medium term with a life expectancy of 10–25 years. **USB flash drives** may be more effective for more regular backup of small files as they are more durable.

1. Explain why secondary storage is necessary in most smartphones. [2]
2. Explain why a solid state drive is commonly chosen for smartphone storage. [4]

DEVICE OPERATION

Solid-state (flash) memory

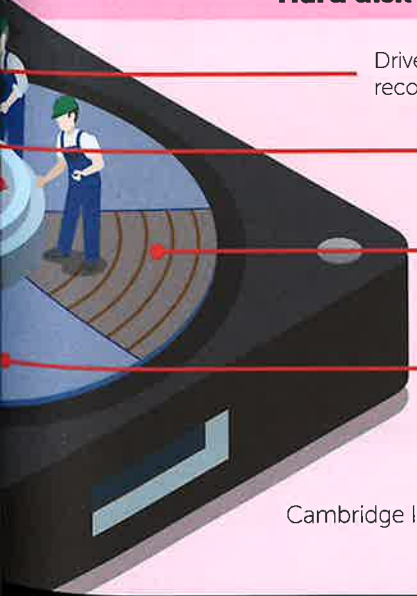
SSDs look like a standard circuit board. They use **NAND** or **NOR** gates (see **page 106**) in electrical circuits to persistently store data. These use microscopic **transistors** as **control gates** and **floating gates** to control the flow of current. Current flowing is represented by 1. No current flowing is represented by 0.

Optical drives (CD / Blu-ray)

An optical drive uses a laser to reflect light off the surface of the disk. One long spiral track contains pits and lands. When the laser beam hits the curved start or end of a pit, the light is refracted and a 1 is recorded. Where light is reflected back directly from the flat bottom of a **pit**, or from an area of the track with no pit (a **land**) a 0 is recorded.

Describe the operation of a hard disk drive. [5]

Hard disk drives (HDD)



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

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

Data is stored on magnetic platters, divided into tracks and sectors. Data is read or written using electromagnets in the drive head.

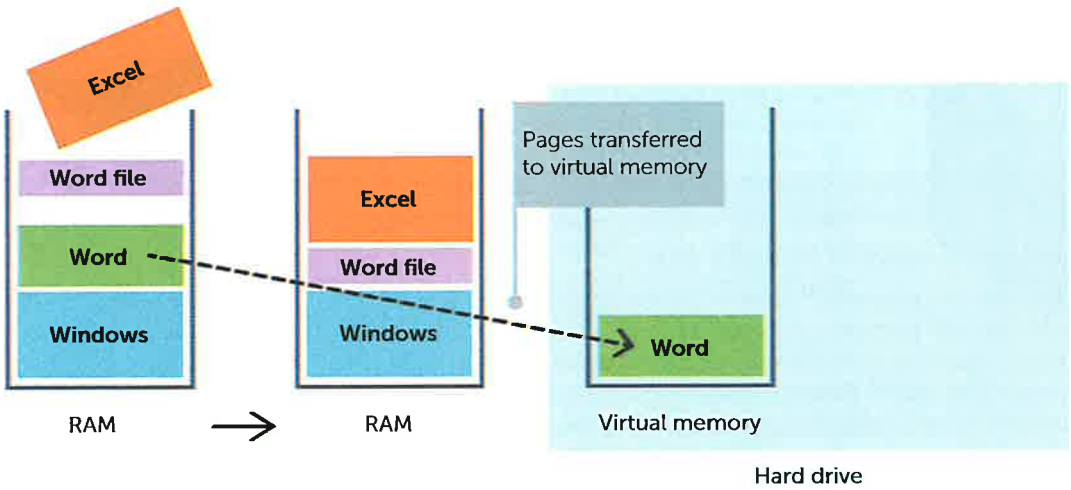
Iron particles on the disk are magnetised to be either north or south, representing 0 or 1.

VIRTUAL MEMORY

Virtual memory is used when there isn't enough space in RAM for all of the programs and data you are currently working on.

A small section of the hard disk is reserved to act like RAM. When the program in virtual memory is needed, it is swapped with something else in RAM. This swapping may cause your computer to noticeably slow down since access to the hard disk isn't nearly as fast as RAM.

In the example below, Windows, Word and a Word document are opened and loaded into RAM from the hard disk. This fills up the available RAM completely. If Excel is then required, the CPU must first move some of the **pages** of data that comprise Word out of RAM into virtual memory to free up enough space for Excel. RAM is divided into fixed size pages of typically 4KiB each. A file currently in RAM may be held in several non-contiguous (ie non-adjacent) pages. A page table saves the number and location of each page.



Imagine what happens when you put some revision material on the floor when you run out of desk space. Access to it gets slower.

1. Explain why getting more RAM would be preferable to using virtual memory. [2]
2. Animation software uses significant amounts of data. Explain why a computer running animation software might use virtual memory. [2]

CLOUD STORAGE

Cloud storage refers to remote file storage on someone else's servers. Often this is with a major third-party company such as Google, Microsoft or Dropbox. Physical servers and storage are needed to store data in cloud storage.

Advantages compared to local storage	Disadvantages compared to local storage
No need to purchase local storage devices or employ maintenance staff. Online storage capacity can shrink or grow as you need it.	May need to commit to an annual subscription.
Files are automatically backed up in the cloud. Cloud services may be used to backup local files.	Backup and restore may be slower than local storage depending on your Internet connection speed.
Greater fault tolerance than local storage as all hardware and backup issues are handled externally.	Security of the data will be managed by a third party - outside of the user's control.
Remote access to files from any Internet connected device, anywhere in the world.	Need an Internet connection to access online data.
Enables file sharing and collaboration.	Limited or delayed support from the cloud storage provider.
One cloud storage centre is more environmentally friendly than millions of individual servers.	You may be responsible for any security breaches over the data, even though you have no direct control over its security.

Cloud storage has become increasingly popular with individuals using mobile devices.

Explain the reasons for the significant growth in usage over the past few years. [4]

NETWORK HARDWARE

Network Interface Cards (NICs)

A **Network Interface Card** (**NIC**) is required to connect any **network-enabled device**. These devices include, for example, home computers, smartphones, printers and Internet-enabled light bulbs. A NIC is a physical component which can operate with a wireless connection or a wired connection using a standard Ethernet cable.

MAC addresses

A (Media Access Control) **MAC address** is used to identify a device with a **unique hexadecimal identification number** assigned to every **Network Interface Card** used in networked devices. The **IP address** assigned to a computer or device may change, however, the MAC address is **set by the manufacturer** and is **static** which means it will always be the same.

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

The first three bytes makes up a unique manufacturer code.

The second three bytes make up the unique serial code.

In 2021 there were about 21.5 billion interconnected devices in the world.

Sam is confused by some networking terms.

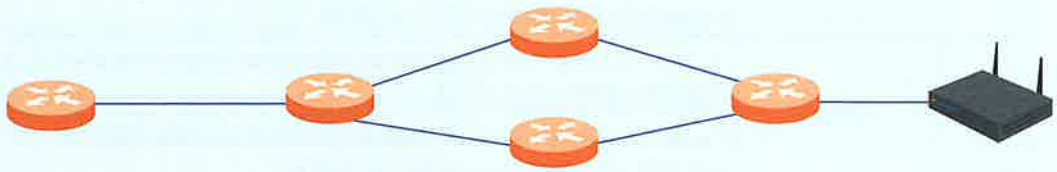
Draw a line to match each of the following terms to the appropriate description. [4]

INTERNET PROTOCOL (IP) ADDRESSES

An **IP address** is a unique **public address** for the **router** or gateway of a network. **Private addresses** are not unique and the IP address of a portable device such as a laptop will change when it is moved, for example between towns. A router can assign IP addresses.

Routers

Routers use IP addresses to direct data packets from one router to another between start and end points on a network. They sit between **local networks** and the **Internet** to join them together with a public IP address for the Internet and a private IP address for the local network. Any local data is passed on to the Internet by the router. Any inbound data from the Internet is received and directed internally to a specific device.



Static and dynamic addresses

The IP address for your whole network is assigned by your **Internet Service Provider**. All computers within a network have IP addresses **assigned** by your own internal network router. **Static** addresses do not change. **Dynamic** addresses do change and are assigned from a list of available addresses at the time they are required. This is because the IPv4 system does not have enough static addresses for everyone.

IPv4 and IPv6 addresses

An **IPv4** address is commonly four numbers (that are each stored using 8 bits) each separated by a full stop.

Your home router is likely to have a private IP address such as 192.168.0.1. This system however is running out of possible addresses owing to the huge rise in networked devices. To solve this problem, a new system called **IPv6** has been developed. This uses a 128-bit address which is usually represented in hexadecimal. This will provide enough address permutations to cater for all devices on the planet.

IPv4 addresses have four groups of digits e.g.: 212.58.244.66

IPv6 addresses have eight groups of digits e.g.: 2001:0000:4136:e378:8000:63bf:3fff:fdd2

Give **three** characteristics of an IP address. [3]

EXAMINATION PRACTICE

1. The CPU is one of the components of a computer system. It contains the following components:

   ALU (Arithmetic Logic Unit)
   CU (Control Unit)
   Clock
   Registers
   Address, data and control buses

   (a) Place each of these components in the correct row of the table. [5]

CPU Component	Function
	Regulates the speed and timing of all signals and computer functions
	Used to transfer data, instructions, memory addresses and control signals from one component to another
	Temporarily store memory addresses, instructions or data that are being processed on the CPU
	Carries out mathematical and logical operations including AND, OR, and NOT. It compares values held in registers
	Coordinates all the CPU's actions in the fetch-decode-execute cycle and decodes instructions

   (b) Special-purpose registers in the CPU include:

   Program Counter (PC)
   Memory address register (MAR)
   Memory data register (MDR)
   Current instruction register (CIR)
   Accumulator (ACC)

   Place each of these registers in the correct row of the table. [4]

Register	Function
	Holds data or a program instruction when it is fetched from memory or data that is waiting to be written to memory
	Holds the results of operations carried out in the ALU
	Holds the memory address of the next instruction to be processed

2. Increasing the amount of cache memory can improve the performance of the CPU.
   State **two** other ways of improving the performance of a CPU. [2]

3. Most modern washing machines are embedded systems.
   (a) Describe **two** characteristics of an embedded system. [4]
   (b) Give **two** other examples of embedded systems. [2]

4. For each of the following devices, state whether they are input or output devices. [5]

Device	Input device	Output device
Barcode scanner		
Speaker		
Projector		
3D printer		
Keyboard		

5. Roger is seeking to purchase a projector for his home cinema room.
   He understands that there are two main types.
   Explain the advantages and disadvantages of a Liquid Crystal Display (LCD) projector compared to those of a Digital Light Processing (DLP) projector. [3]

6. A scientific laboratory requires a sensor to trigger an alarm if the environmental conditions in a chamber change.
   For each requirement, suggest **one** suitable sensor that could be used to measure the condition:
   (a) Absolute darkness: [1]
   (b) Low oxygen: [1]

7. State **one** use of ROM (Read-Only Memory). [1]

8. A tablet computer is being designed with 256 GB storage.
   Suggest **one** suitable storage device. Justify your answer. [3]

9. ArcAccounts is a small accountancy firm. They are considering whether to move all their data from their own server to a cloud-based service provider.
   (a) Discuss the issues the company should consider before a decision is made. [6]
   (b) Hardware accessing the cloud service will need a network interface card with a MAC address.
   Explain what is meant by a MAC address. [2]

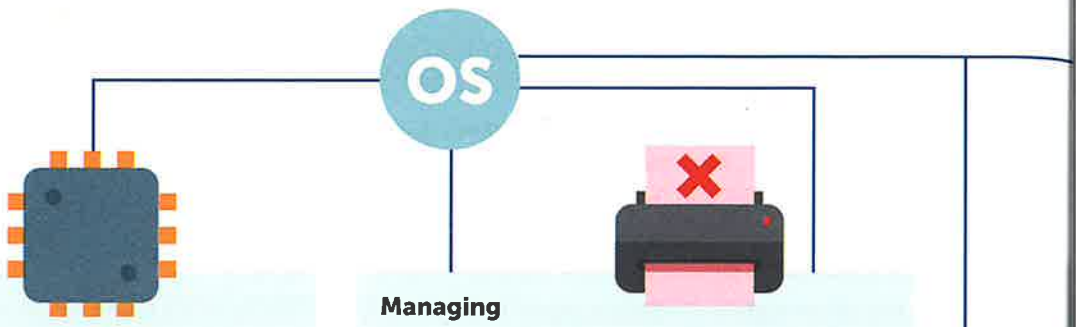
10. Describe the operation of a hard disk drive. [4]

APPLICATION AND SYSTEM SOFTWARE

Application software runs on the operating system and provides the services that the user requires, for example, a word processing package, stock control software or an in-car navigation system.

System software programs are those that are needed to enable the computer to function, including the operating system, utilities, library routines and programming language translators.

Major operating systems include **Windows®**, **Linux®**, **MacOS®**, **Apple iOS®** and Google's **Android®**. An **Operating System** (**OS**) is a group of programs that is essential for managing the computer's resources. It handles several crucial tasks:



Managing multitasking:

The OS coordinates the CPU to **schedule processes**. All processes currently in memory waiting to be executed are held in a circular queue. The CPU allocates each process a **time slice** and switches to the next process very quickly, making it seem as though they run at the same time. This is known as **multi-tasking**.

Each process has exclusive use of the CPU during its allocated time slice. Several time slices may be required to complete a process, so an unfinished process must await its next turn at the back of the queue once its time slice expires.

Managing peripherals and drivers:

Peripheral devices connected externally to the CPU include **printers**, **keyboards** and **monitors**. Users must communicate with devices via the OS. "Out of paper" messages for example, must come via the OS. A **driver** is software used to provide an interface for a hardware device.

Managing files:

The OS allows users to create, delete, move, save and copy files, or allocate them to folders. It can search for files, restore deleted files, free up space for new files and prevent conflicts when two users attempt to modify the same file at once. **Access rights** to individual files may also be managed.

Managing user accounts:

Different users will each be provided with an account with their own **user name** and **password management**. Each account can be granted different levels of access depending on their needs and levels of security. The OS can also monitor **login activity** and log out users after set periods of inactivity.

Handling interrupts:

An **interrupt** is a signal from a software program or hardware device to the CPU. A **software interrupt** occurs when an application program terminates or requests certain services from the OS. A software interrupt also occurs with a program error such when a **division by zero** occurs, or when two processes try to access the same memory location.

A **hardware interrupt** may occur, for example, when an I/O operation is complete or when a new USB device is plugged in and detected.

A hardware interrupt also occurs when a key is pressed on the keyboard or the mouse is moved.

Interrupts are also triggered regularly by the timer, to indicate that it is the turn of the next process to have processor time. It is because a processor can be interrupted that **multi-tasking** can take place.

Software called the **Interrupt Service Routine** (**ISR**) determines how to handle each interrupt.

Providing an interface:

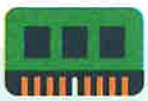
The OS provides a means of interacting with the computer, often through buttons, keyboards, touchscreens or mice. An interface may be purely text using a **command line interface**, or it may be a **Graphical User Interface** (**GUI**) / **WIMP interface**.

WIMP stands for Windows, Icons, Menus, Pointer.

Providing a platform for running applications:

When a new application is installed on your computer, the OS will manage this process. The OS will also allocate memory space for the application and control the application's access to data or devices. User access to programs is also managed.

Managing memory:



Files and programs need to be in memory for the CPU to perform tasks which use them. The OS moves programs and files between memory and storage (e.g. hard drive) if virtual memory is required.

Providing system security:

The OS controls user access to prevent users from accessing files or folders they should not have access to. **Security updates** from software manufacturers are also downloaded to help fix bugs and improve security against malware. Files on the hard disk may also be encrypted.

The bootloader

When a computer is turned off, the operating system and all programs and data in use are lost from RAM. The OS remains in non-volatile storage, e.g. hard disk. The **bootloader** (also known as the **bootstrap loader**) is a small piece of software (**firmware**) that is responsible for starting up (booting) a computer. The firmware is held in ROM (Read Only Memory) and automatically starts up when the computer is switched on. It loads the operating system into main memory (RAM).

Explain why the bootloader instructions are stored in ROM and not in RAM. [1]

TYPES OF PROGRAMMING LANGUAGE

High- and low-level programming languages

A **high-level language** has a **syntax** and structure similar to English, that is designed to be **understood by humans**. Most programs are written in high-level languages for this reason.

For example:

```
speed = distance / time
print(speed)
```

Python, Visual Basic and C# are examples of high-level languages. These are **hardware independent**, meaning they can run on any system. The programmer can therefore concentrate on their own task rather than concerning themselves with the architecture of the computer.

High-level code must be **compiled** or **interpreted** before it can be run.

Assembly language is a **low-level language** which is typically used to control specific hardware components. Low-level commands are **hard to read** but they execute very quickly. Software for embedded systems is often written in assembly language, because the machine code produced by the assembler is fast, efficient and occupies less memory than that produced by translating the equivalent high-level instructions.

Machine code

Processors execute machine code, the binary instructions produced by the compiler or interpreter. Each type of processor has its own specific machine code instruction set so machine code produced for one type will not work on another.

A programmer is writing a revision app for a mobile phone. The program is written in a high-level language.

Describe **two** differences between high-level and low-level languages. [4]

TRANSLATORS

There are three common types of translator: a **compiler**, an **interpreter** and an **assembler**. These work in different ways, each having different advantages.

Compiler	Interpreter
Translates a high-level language program (source code) in one go to produce object code	Translates and executes one line of a high-level language program at a time
A compiled program executes faster than an interpreted one as it is already in machine code	Takes more time to execute as each instruction is translated before it is executed
Produces an executable file so the original code does not need to be compiled again	Source code will be interpreted or translated every time it is run
No need for the compiler to be present when the object code is run	The interpreter must be installed to run the program
After compilation, the compiler provides an error report for the whole code if any errors are detected	An interpreter stops execution as soon as an error is found. This is useful when developing a program

Languages such as Python and JavaScript are interpreted so need an interpreter to run, whereas languages such as C# and VB need to be compiled before they are run.

1. Describe the need for a translator when using a high-level language. [2]

Assembler

An **assembler** converts assembly language into machine code. This is a simple conversion as, in general, every assembly language instruction is translated into a single machine code instruction.

Example: The assembly code instruction LDA #17 loads the value 17 into the accumulator.

This uses the **mnemonic** 'LDA' which means "Load into the accumulator".

The assembler translates this into the machine code instruction, e.g. 11010100 00010001.

2. Explain **two** reasons why assembly code might be used in an embedded system such as a digital camera. [4]

THE INTEGRATED DEVELOPMENT ENVIRONMENT (IDE)

An **IDE** is software used to enter and edit source code. It will also compile programs to machine code and have debugging features.

There are several tools or facilities of an IDE that are useful to programmers.

Error diagnostics (Debugging and error detection)

An **interpreter** or **compiler** highlights syntax errors in the code as you type or when you try to compile or run the code. Runtime errors will also be flagged up often giving the location of the error in the code and a description of the error.

Breakpoints can be set to pause a program at a certain point. This enables the programmer to check the value of variables at specific points in the code to find errors. A **watch window** is commonly used to display the value of specific variables as they change with each line of code. **Stepping** enables the programmer to step through line by line as they monitor changes in values.

Runtime environment

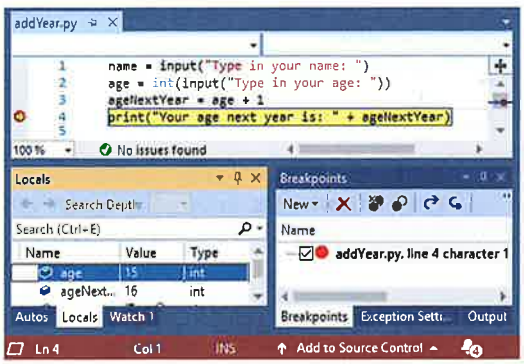
This enables a program to be run. Checks for runtime errors and other testing can be carried out.

Translators

The IDE will use either a **compiler** or **interpreter** to translate the high-level code into machine code so that it can be run.

Code editors

The **editor** enables programmers to write and edit their code. **Auto-completion** and **auto-correction** functions often offer to complete code or suggest corrections. **Keywords** and different types of syntax are commonly **highlighted** in different colours making the code easier to read. **Prettyprint** features enable code formatting, indentation and syntax highlighting to be displayed.



Explain what is meant by a runtime error. [2]

EXAMINATION PRACTICE

1. Describe what is meant by the following terms, and give an example of each.
   (a) Application software [3]
   (b) System software [3]

2. (a) Describe what is meant by firmware. [2]
   (b) Give an example of its use. [1]

3. The operating system is responsible for managing multitasking and for providing system security.
   (a) Describe **one** task performed by the operating system to help maintain the security of a computer system. [2]
   (b) List **three** other functions of the operating system apart from managing multitasking and providing system security. [3]

4. Which **one** of the following is an example of utility software? Tick (✓) one box. [1]
   A. ☐ Payroll software
   B. ☐ Compression software
   C. ☐ A mobile game app
   D. ☐ A word processor

5. Which **one** of the following best explains why an embedded system cannot be modified once manufactured? Tick (✓) one box. [1]
   A. ☐ The system uses secondary storage
   B. ☐ The casing is sealed
   C. ☐ The code is written in a low-level language
   D. ☐ The system is stored in ROM

6. Shona is using a high-level language to learn programming.
   Explain **two** features of an IDE (Integrated Development Environment) that can help Shona to find or prevent errors in her programming code. [4]

7. (a) Explain the difference between a compiler and an interpreter. [4]
   (b) PoundSoft is a software company selling accountancy software.
   The software is compiled rather than interpreted.
   Explain **two** reasons why they sell the software in this form. [4]

8. (a) Give **two** examples of events that will cause a software interrupt. [2]
   (b) Give **two** examples of events that will cause a hardware interrupt. [2]

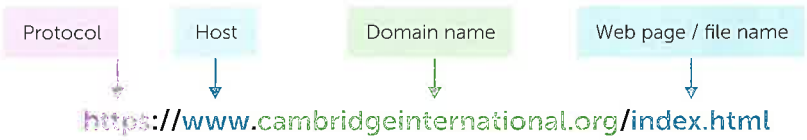
THE INTERNET AND THE WORLD WIDE WEB

The **Internet** is a global **infrastructure** of interconnected networks. The **World Wide Web** is all the web pages that are accessible via the Internet.

Uniform Resource Locator (URL)

Web pages are held on computers connected to the Internet. The World Wide Web (WWW) uses the Internet as a service to communicate the information held on these pages. Every web page has its own unique text-based address, known as the **Uniform Resource Locator (URL)**.

These pages are accessed using a program called a **web browser** such as Google Chrome™, Microsoft® Edge or Apple® Safari. A typical web page address is shown below:



http and https protocols

A **protocol** is a set of rules for communication between devices. It covers standards for physical connections, cabling, mode of transmission, data format and error detection. It allows equipment from different suppliers, all following the standard communication protocol, to work together.

HTTP

HTTP (HyperText Transfer Protocol) is the standard protocol used to communicate across the Web. When you visit a site which has an address starting with **http**, any data you enter into the site, such as your name, address or bank details, is sent in plaintext and is susceptible to hacking.

HTTPS

HTTPS (HyperText Transfer Protocol Secure) is a secure protocol, meaning that any data transmitted between the website and your browser is encrypted and cannot be understood by anyone intercepting it. The web server you are connecting to is also **authenticated** to ensure the site is genuine.

Functions of a browser

In addition to rendering and displaying web pages, the functions of a web browser include:

- Storing bookmarks and favourites
- Recording user history
- Allowing use of multiple tabs
- Storing cookies
- Providing navigation tools
- Providing an address bar

LOCATING AND DISPLAYING A WEB PAGE

The main purpose of a web browser is to render the **Hypertext Markup Language** (**HTML**) used to code a web page, and to display the resulting web page as the user should see it.

IP addresses

Every public-facing, networked computer or computing device in the world has a separate **IP address**.

An IPv6 address is 128 bits long, arranged in eight groups of 16 bits each. Each group is four hexadecimal digits, separated by colons. See page 45 for more detail.

Domain name server

The role of a **Domain Name Server** is to convert a website address (**URL** e.g. www.google.co.uk) into an **IP address** so that a browser on a client machine can make a request to the correct **web server** hosting that web page. If the DNS server does not have an entry for the domain name, it passes the request to another more authoritative DNS server. An error is sent back if no match is found.

HTML

HTML is the standard markup language for creating web pages using elements in <tags>:

```
<body>
<h1>Hello!</h1>
<p>Welcome</p>
</body>
```

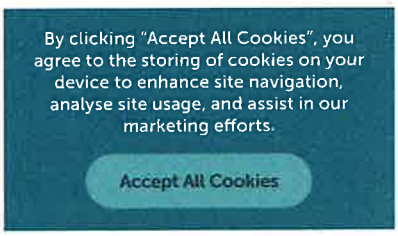
Be your own DNS! Try typing 172.217.14.195 into a browser instead of www.google.co.uk.

172.217.14.195

Cookies

Cookies are text files with small pieces of data They are used to track user preferences, hold items in a shopping cart, store login details and other personal details.

Session cookies are created and replaced each time a user visits a website. **Persistent cookies** are created and saved on the first visit and retained until they expire.



Ahmed creates a new account on an online shopping site. He accepts the use of cookies.

(a) List **two** advantages to the retailer of using cookies. [2]

(b) List **two** advantages to Ahmed of accepting the use of cookies. [2]

DIGITAL CURRENCY

A digital currency is any money-like asset that only exists in digital form and is exchanged digitally, with no physical banknotes or coins in circulation.

Bitcoin

Bitcoin is the most well-known digital currency. It can be thought of as digital gold. People can buy fractions of a Bitcoin, just like they can with gold, and use that to trade with others for goods, services and other currencies at the latest market price. Some shops, car manufacturers and even nations are now accepting Bitcoin as payment.

Explain why digital currencies are gaining in popularity. [3]

Blockchain

Blockchain database technology is used to form a **digital ledger** that is a time-stamped series of transaction records sequentially linked in a chain, tracking the movement of a digital currency.

- Each new transaction is added onto a chain of transaction blocks.
- Each transaction carries the user's unique ID number and a reference to the previous block. The reference is a **hash total** calculated with a 'hashing' algorithm using data from the previous block.
- A change to data in any block in the blockchain would create a ripple effect of incorrect hash totals throughout the blockchain.
- The blockchain is copied and stored on multiple different servers, so that if a change to a data block is not reflected in all copies, it is rejected. This maintains the integrity and trustworthiness of the data.

Example of a hashing algorithm:

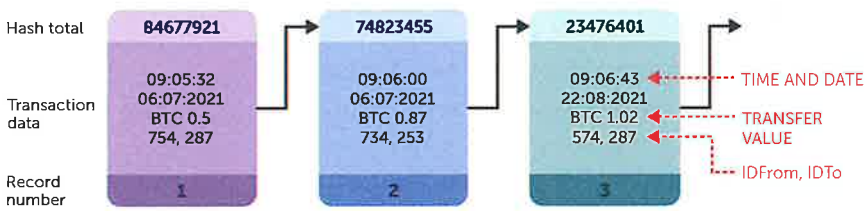
Given two numbers between 1 and 10,000, generate a hash total between 1 and 1000.

Multiply the first number by 3, the second by 5 and divide the result by 997. The remainder is the hash total.

*e.g. 56, 721: (56 * 3 + 721 * 5) / 997*

= 3 remainder 784

784 is the hash total



Hash algorithm applied to transaction 1:
(84677921, 090532, 06072021, BTC 0.5, 754, 287) * algorithm = 74823455 (Hash total)

CYBER SECURITY THREATS

Cyber security relates to the processes, policies and practices designed to protect against any threats or unauthorised access to networks, computers, programs or data. A combination of strategies should be employed to provide a robust defence against these threats.

Forms of attack and defence

Hacking

Automated or manual attempts to gain unauthorised access to programs or data.

Firewall to block access by unauthorised devices

Brute-force attacks

Automated or manual attempts to gain unauthorised access to secure areas by trying all possible password or key combinations.

Strong passwords with limited attempts allowed

Data interception and theft

Data may be intercepted during transmission, but physical theft can occur where storage devices or data files are left insecurely.

Encryption / Physical locks / Passwords

Malware: Spyware

Software that hides on your computer recording your activities to send back to a third party for analysis.

Anti-malware / Anti-spyware software

Social engineering: Phishing

Phishing emails redirect a user to a fake website where they trick the reader into divulging confidential information such as passwords that can be used fraudulently.

Network policy / Firewall / User awareness of phishing 'clues'

Malware: Pharming

A form of cyberattack that sends you to a fake website that looks like the real thing. The user types in a legitimate address and is redirected to a fake website.

Checking the URL, making sure the site says HTTPS and not just HTTP. Using a trusted Internet Service Provider (ISP)

Social engineering: Shouldering

Looking over someone's shoulder when they enter a password or PIN.

Concealing your password or PIN entry

Suspicious links

Links in emails may not be genuine.

Hovering over a suspicious link should reveal the link's destination before clicking it.

1. List **three** clues that may reveal that an email is a fraudulent "phishing" attempt to get the user to divulge personal information. [3]
2. Give **three** guidelines that should be used in order to create a strong password. [3]

MALICIOUS CODE (MALWARE)

Malware describes malicious software written to cause inconvenience or damage to programs or data.

Viruses and worms

A **virus** is a program that is installed on a computer without the user's knowledge or permission with the purpose of doing harm. It includes instructions to replicate (copy itself) automatically on a computer and to other computers.

A **worm** can cause damage similar to a virus, corrupting files and stealing information, but it does not self-replicate.

Anti-virus software runs on a computer checking all files for known viruses so it must be kept up to date.

Trojan horse

A **Trojan**, named after the famous Ancient Greek story of the Trojan Horse, is a program which masquerades as having one legitimate purpose but actually has another. It is normally spread by email. The user is invited to click on a link for some everyday or interesting purpose, which then executes a program which may, for example, give the controller unauthorised access to that computer.

Ransomware and Adware

Ransomware is a form of malware that encrypts a victim's files. A ransom is then demanded from the attacker in return for instructions on how to decrypt the files. The ransom may range from a few hundred to hundreds of thousands of dollars, often payable to the cybercriminals in Bitcoin.

Adware is malware that automatically displays onscreen advertisements. It is not always dangerous but may redirect the browser to an unsafe site, or contain spyware.

Spyware enables a remote user to gather activity, passwords and payment information from another user's computer.

Distributed denial of service (DDOS) attack

A DDOS attack is a malicious attempt to disrupt the normal traffic of a targeted server with a flood of Internet requests. Typical targets include Internet shopping sites, or any organisation such as a bank or credit card company that depends on providing online services

The DDOS attack uses multiple interconnected devices in different locations to establish a **botnet**.

The attacker is then able to direct the attack by sending remote instructions to each bot to send requests to the target's IP address. This results in the server becoming overwhelmed, so that genuine requests cannot be handled. Eventually the server fails or times out.

Because each bot is a legitimate device, it is difficult to separate the fraudulent requests from genuine ones. A firewall can help to protect against a DDOS attack.

KEEPING DATA SAFE

Biometrics

Biometrics use pre-recorded human characteristics to authenticate authorised users of a system or device.

Methods include:

- Facial recognition
- Iris and retina scanners
- Fingerprint or palm recognition
- Voice recognition

The user is required to look at a camera, press their fingerprint to a sensor or speak into a microphone. The image or recording is taken and analysed for key features. This is compared with several pre-recorded examples for that individual and if a match is found, access is granted.

Two-step verification

Two-step verification involves two authentication methods performed one after the other to prove someone's identity. For example, a 4-digit code may be sent in an email or text message to your phone, which you have to type in to complete a login process.

Passwords

Password systems are effective at preventing access to systems. Passwords should be secure and changed regularly.

Automatic software updates

Automatic software updates will keep applications and the operating system constantly up-to-date. This enables bugs and security flaws in the software to be fixed, as soon as they are known about, with a **patch**.

Secure socket layer (SSL) protocol

SSL provides a secure channel between two computers or devices operating over the Internet or an internal network. SSL can be used to secure communication between a web browser and a web server. This will then use an HTTPS address for the website. The 'S' stands for 'secure'.

Privacy settings

Privacy settings allow a person to control what information about them is shared on a social media platform. A user can control who has access to what they post, and who can see their full profile.

Windows also allows you to control privacy settings – by default, Microsoft will monitor the websites you browse and the apps you use in order to personalise your Windows experience.

You can tell if a site is using SSL when a padlock is displayed. Clicking on the padlock will reveal the digital certificate information.

Proxy servers

A **proxy server** is any machine that checks and forwards traffic between networks or protocols. It's an intermediary server separating end-user clients from the destination sites that they browse. A proxy server can provide varying levels of functionality, security and privacy depending on your needs or company policy.

EXAMINATION PRACTICE

1. A bank holds data on a database kept on the organisation's server about each of its account holders, including personal data, credit rating, credit limit and current balance.

   Social engineering techniques have been used by callers contacting their call centre.

   (a) Explain what is meant by social engineering. [2]

   (b) Give **one** possible consequence of such an attack on each of:

   (i) The bank's customers [1]

   (ii) The bank's staff [1]

2. Sylvia runs a small accounting business from home. She keeps details of all her customers on her desktop computer.

   Describe **two** ways in which this data may be put at risk and suggest a way of reducing the risk in each case. [4]

3. Blockchain is a type of peer-to-peer database ledger.

   (a) Explain what is meant by blockchain technology. [2]

   (b) Give **two** advantages of blockchain technology. [2]

4. Roshin has received an email from an unrecognised sender.

   (a) Give **three** ways in which a phishing attack by email may be recognised. [3]

   (b) Describe **one** possible consequence of a phishing attack. [2]

5. Five cyber security terms are given below. For each row in the table below, choose the letter A, B, C, D or E that best matches the description. Letters should not be used more than once. [3]

   A. Pharming
   B. Trojan
   C. Virus
   D. Biometrics
   E. DDOS attack
   F. Two-step verification

Description	Letter
Malware that automatically replicates itself on a computer.	
A cyber-attack designed to redirect legitimate website traffic to a fake website.	
Use of human features for authentication.	

6. A user enters the URL www.bbc.co.uk into their browser. It appears in the address bar as:

   A protocol is used by the browser to download the web page from the web server.

   (a) State what is meant by a protocol in the context of data transmission. [1]

   (b) State which protocol the web browser will use to download the web page information from the web server. [1]

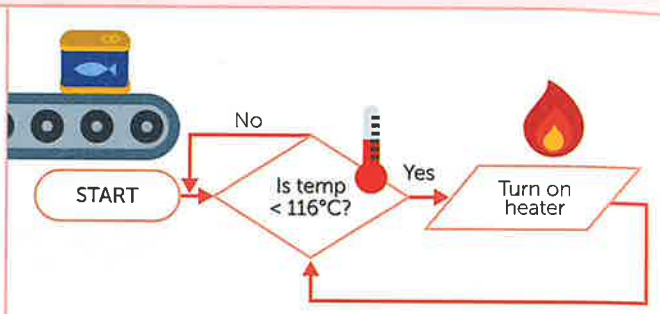
   (c) Explain why a standard protocol is necessary. [2]

AUTOMATED SYSTEMS

An **automated system** usually comprises **sensors** to **take measurements** of their environment, **microprocessors** to process the information and make decisions based on the readings, and **actuators** to make a **physical change** to the environment such as a motor opening a door, shutting a valve or turning on an alarm.

Industry

The canned food industry relies on bringing the contents of tins to a certain temperature under pressure before they are then cooled, and the food inside is deemed safe to eat. The temperature of the cans is measured on the production line and a response is made if the temperature falls below a pre-set minimum.



An automated process means that far greater accuracy of measurements can be achieved. Even a 99.9% accuracy level on a production line with 100,000 cans per day means that 100 cans will contain potentially harmful bacteria. Automated systems are more expensive to install, but they do not require salaries and can work 24/7, significantly increasing productivity.

Agriculture

Humidity, pH and moisture sensors in agriculture ensure that the crops are grown and stored in the optimum conditions. Microprocessors will process any readings from the sensors and cause actuators to turn on sprinkler systems or open vents, or turn on heaters, for example.

Transport

Proximity sensors in self-driving cars can automatically apply the brakes if a car gets too close to another in front. Aircraft are equipped with thousands of sensors and actuators to adjust the wing flaps, throttle, rudder etc., as part of the autopilot and safety systems. .

A scientific laboratory is conducting tests on 50,000 test tubes containing different concentrations of a new vaccine. They are considering using an automated system to conduct the tests.

Describe **two** disadvantages of installing an automated system in a laboratory. [4]

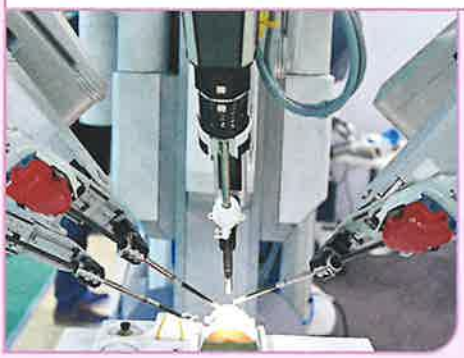
ROBOTICS

Robots include factory equipment used on a production line, domestic robots such as lawnmowers and vacuum cleaners, medical robots, agricultural robots and drones.

Further examples

- In agriculture, automated planting, irrigation, application of nutrients and harvesting on predefined paths maximise yields.
- In medicine, robots are used in surgery, actuated and sensory prostheses (false limbs), disinfecting rooms, clinical training and many other applications.

600 drones were used in China to create images in the air.



1. Give **two** characteristics of a robot. [1]
2. Describe **one** way in which a robot could be used for entertainment. [2]
3. Vehicle manufacturers use robots for welding and spraying car panels. Explain why a manufacturer may choose to use robots rather than humans for these tasks. [4]

ARTIFICIAL INTELLIGENCE

Artificial intelligence involves the **simulation** of **intelligent behaviours** by computers. An early application of artificial intelligence was teaching computers how to play chess. A set of rules was written down to evaluate possible moves and explore the most promising ones.

Expert systems

An expert system simulates a human expert. These systems have a **knowledge base**, a **rule base**, an **inference engine** and an **interface** through which we can interact with them.

Expert systems are used in many fields such as medical diagnosis, telecommunications, identifying plants or rocks, video games and transportation. An early application of an expert system was medical diagnosis. The computer is first 'trained' by being given the symptoms of hundreds of thousands of patients, together with a correct diagnosis for each one. From this **knowledge base**, a set of rules (the **rule base**) can be formulated and a program (the **inference engine**) used to apply the rules to diagnose a new patient with particular symptoms. The interface allows the doctor to interact with the expert system, answering questions posed by the system and receiving results with associated probabilities.

Machine learning

Instead of giving the computer the rules for a task, using machine learning the computer learns from first principles the structure of the data and potentially, a solution to the problem.

For example, billions of spam emails are sent every day – that's a majority of all emails. Without spam filters inboxes would drown in spam. The classic AI method would be to come up with a list of spammy words such as "free", "opportunity", "won", "prize", "congratulations" and a set of rules to decide whether the email is spam. The problem is that all of these words also occur in legitimate emails.

The spam filtering program is fed a large amount of email which it uses as training data. It then creates rules which it has 'learnt' for itself - hence it is known as machine learning. The rules are then applied to identify spam.

More spam is continuously fed into the program so that it constantly improves its rules.

Deep Blue was a chess computer developed by IBM. It is famous for defeating the chess world champion, Garry Kasparov, in 1997.

Global applications of AI include optimising harvests, interpreting medical images, fraud detection and deciding on sentences for offenders found guilty in a court of law.

State **three** advantages and **one** disadvantage in using AI to grade students' exam papers. [4]

EXAMINATION PRACTICE

1. Automated systems are used in many countries to issue flood warnings. Sensors may be placed at various points along a river to measure the depth of water and speed of flow.

   Describe **two** other devices other than sensors which would be needed as part of an automated system for flood defence. [4]

2. Miller and Co have installed an automated lighting system in their small office block.

   (a) Give **two** advantages to the company of an automated lighting system. [2]

   (b) Give **two** advantages to the employees of an automated lighting system. [2]

3. Automated systems are used in some intensive care units.

   (a) Describe **two** advantages of using an automated system to dispense medication. [4]

   (b) Describe **one** disadvantage of an automated system to dispense medication. [2]

4. Robots are being introduced into a factory which makes computer components.

   (a) Describe **three** ways in which this could affect workers. [3]

   (b) The management thinks that using robots will reduce costs.

   Give **two** ways in which costs may be reduced. [2]

   (c) Give **two** other advantages to management of introducing robots. [2]

5. Robots are increasingly being used in manufacturing production lines. However, there is limited use of domestic robots.

   Explain **two** reasons why this is so. [4]

6. An expert system is used by a travel website.

   (a) Explain what is meant by an expert system and list its major components. [4]

   (b) Give **two** other examples of the use of expert systems. [2]

TOPICS FOR PAPER 2
ALGORITHMS, PROGRAMMING AND LOGIC

Information about Paper 2

Written exam: 1 hour and 45 minutes
75 marks
50% of IGCSE

Specification coverage

Algorithm design and problem-solving; programming concepts; databases; Boolean logic.

The content for this assessment will be drawn from subject content sections 7 to 10 of the specification.

Questions

This paper consists of short-answer and structured questions, and a scenario-based question. All questions are compulsory. Calculators are not permitted.

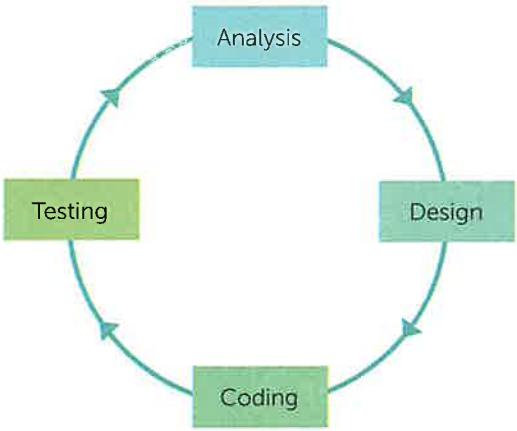
THE PROGRAM DEVELOPMENT LIFE CYCLE

There are four main stages in the program development life cycle:

Example

The four main stages in the program development life cycle are:

- **Analysis:** abstraction, decomposition of the problem, identification of the problem and requirements
- **Design:** decomposition, structure diagrams, flowcharts, pseudocode
- **Coding:** writing program code and iterative testing
- **Testing:** testing program code with the use of test data



Abstraction involves identifying the key parts of the problem and removing any unnecessary detail so that it becomes easier to solve.

For example, if a program is to be written to simulate a card game, the first task to be accomplished may be 'shuffle the cards'. This is an abstraction – implementing it will involve specifying a way to randomise the order of 52 values representing the cards. We can refer to 'shuffle' throughout the program without specifying how it will be done.

A self-driving car is being developed. The software has to be capable of distinguishing between an animal and a person crossing the road in front of the car.

(a) Define what is meant by abstraction. [2]

(b) Give **one** example of how abstraction could be used in developing this software. [1]

STRUCTURE DIAGRAMS

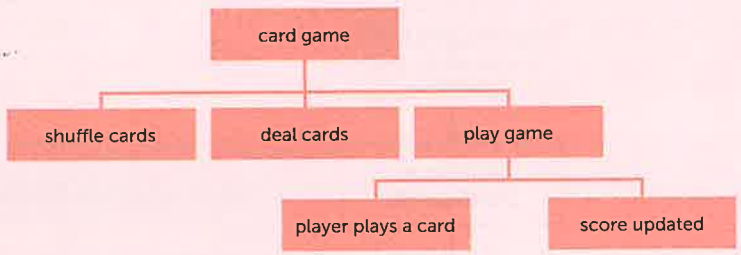
Identifying inputs, processing, outputs and storage is the first stage in the **decomposition** of a problem. Each of these stages may then be broken down.

Stages in decomposition

Decomposition means breaking down a complex problem into smaller, manageable parts which are easier to solve. This comprises the following steps:

- Identify the main problem
- Identify the component parts of inputs, processes, outputs and storage
- List the main sub-problems, sub-systems, functions or tasks
- Break these down into smaller sub-problems or sub-tasks which can then be completed separately.

A **structure diagram** may be used to show the structure of a problem, its subsections and links to other subsections. Applied to a card game, this could look like the following:



A program is to be written to record new customer details for an online store. A new customer registering for an account enters their name, address and email address, and is given a unique username by the system. They then enter a password, which is validated and must contain a mixture of uppercase and lowercase letters and numbers. Once a valid password is entered, it is encrypted and stored with the username and personal details on a permanent storage device.

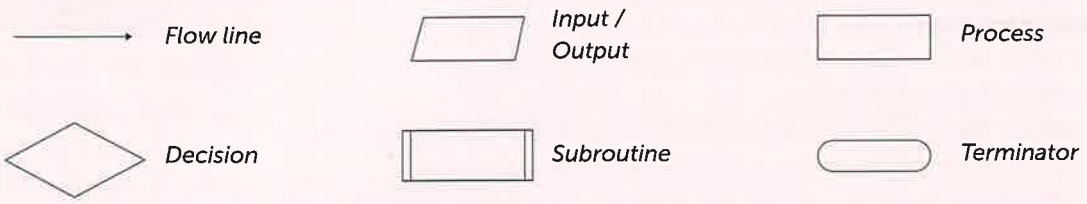
(a) Describe how **abstraction** could be used in the initial design of the program. [2]

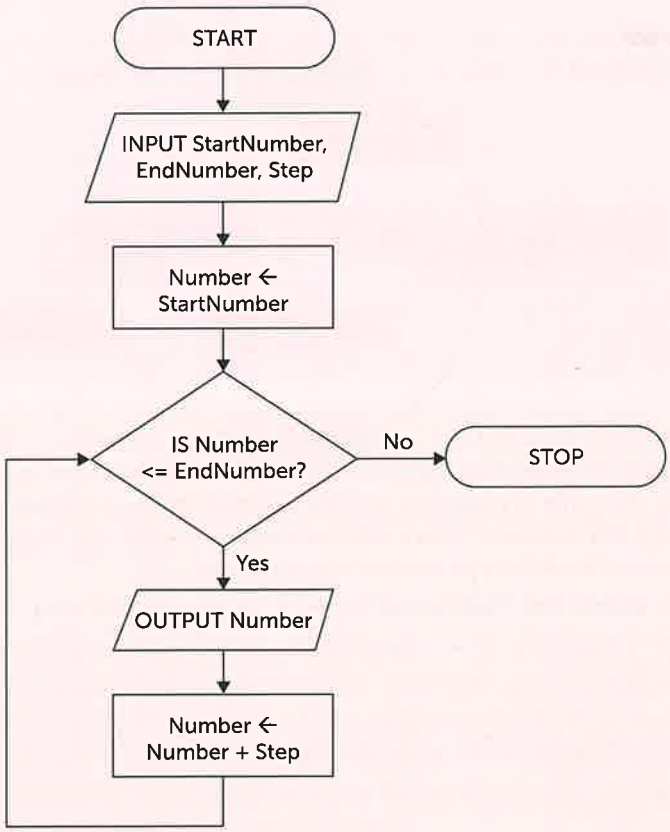
(b) Draw a structure diagram to show how program modules relate to each other. [4]

FLOWCHARTS

Flowcharts are a useful tool that can be used to develop solutions to a problem.

Standard flowchart symbols are shown below:





Look at the flowchart above.

(a) What will be output if the user enters 7, 50, 10 for the three input values? [2]

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

PSEUDOCODE

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

Using pseudocode

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

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

```
INPUT StartNumber
INPUT EndNumber
INPUT Step
Number ← StartNumber
WHILE Number <= EndNumber DO
    OUTPUT Number
    Number ← Number + Step
ENDWHILE
```

Note that if there are three values to be input, you must use three input statements if you are asked to use pseudocode or a programming language rather than pseudocode. Each input statement is used to input a single value and assign it to a variable.

IGCSE standard pseudocode

There is a published version of pseudocode which will be used in exams. It is defined in the syllabus for Cambridge IGCSE (9-1) Computer Science, downloadable from the Cambridge website.

Note the following points in the pseudocode used in this course:

- Comments start with two forward slashes // and continue to the end of the line
- Identifiers contain only letters and digits 0–9 and start with an uppercase letter
- A variable may be declared with a statement such as

```
DECLARE StartNumber : INTEGER
DECLARE Total : REAL
DECLARE ValidPassword : BOOLEAN
```

- A constant may be declared with a statement such as

```
CONSTANT VAT ← 0.2
```

- The assignment operator is ←

(In some word processors, try typing < followed by two hyphens -- to display ←)

Write pseudocode to display on the screen the values of three variables x, y and z. [1]

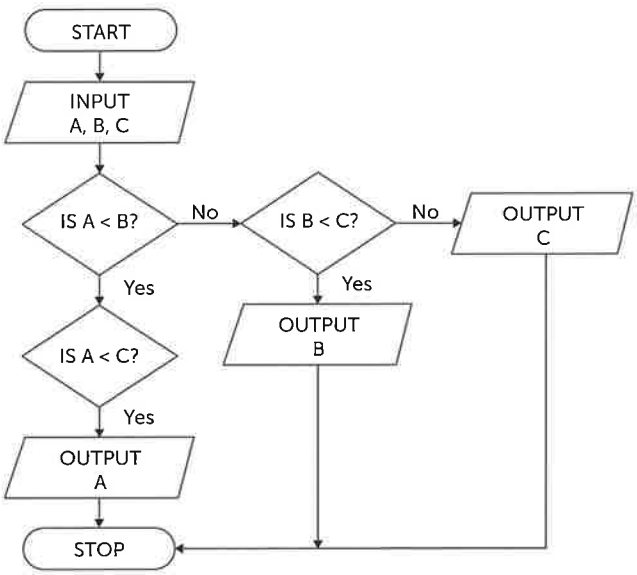
EXPLAIN THE PURPOSE OF A GIVEN ALGORITHM

Every problem to be solved using a computer involves input, processing and output.

Input, process, output

Many problems involve reading data from or writing data to a permanent storage device.

- The **input** may be typed by someone at a keyboard, or it may be a reading from a sensor such as a moisture, pressure or temperature sensor, or some other form of input.
- The data then has to be **processed** in some way – for example by sorting a list, performing calculations or using temperature readings to predict ice on the roads.
- **Output** is the end result after processing. This could be, for example, a printed report, a valve opened or closed, graphics displayed on a screen or data stored on a storage device.



(a) Explain the purpose of the algorithm shown on the left. [1]

(b) Write the algorithm using pseudocode instead of a flowchart. [4]

This algorithm uses a **nested selection** structure.

The IF statement has an IF statement nested inside it. In this example, the ELSE statement also has a nested IF statement.

LINEAR SEARCH

Using a linear search, each item will be checked one by one in the list until the item is found, or the whole list has been searched without success.

A list of 10 names is shown below.

Harry	Jo	Anne	Zoe	Peter	Ken	Steve	Geri	Fiona	Bob

1. (a) State which items are examined when looking for **Zoe** using a linear search. [1]
   (b) State how many items will be examined when looking for **Dave** using a linear search. [1]

A linear search algorithm

```
//Perform a linear search of an array
DECLARE AList : ARRAY[1:10] OF INTEGER
AList ← [14, 2, 3, 11, 1, 9, 5, 8, 10, 6]
OUTPUT "List to be searched:", AList
Found ← FALSE
Index ← 1
INPUT SearchItem
WHILE Found = FALSE AND Index <= LENGTH(AList) DO
   IF AList[Index] = SearchItem
      THEN
        Found ← TRUE
      ELSE
        Index ← Index + 1
   ENDIF
ENDWHILE
IF Found = TRUE
   THEN
      OUTPUT SearchItem, "in position", Index, "of the list"
   ELSE
      OUTPUT "Item not found"
ENDIF
```

2. (a) State the purpose of the algorithm given below. [1]

```
AList ← [14, 2, 3, 11, 1, 9, 5, 8, 10, 6]
X ← AList[1]
FOR i ← 2 TO LENGTH(AList)
   IF AList[i] < X
      THEN
         X ← AList[i]
   ENDIF
NEXT i
OUTPUT X
```

2. (b) State **two** ways of making the algorithm easier to understand [2]

BUBBLE SORT

A bubble sort works by repeatedly going through the list to be sorted, swapping adjacent elements if they are in the wrong order.

To sort a list of n items, a maximum of n – 1 passes is required. (The items may be alphabetical or numeric.)

Example

A list of 5 numbers 7, 3, 5, 9, 4 is to be sorted. Show the state of the list after each pass.

List	7	3	5	9	4	
Pass 1	3	7	5	9	4	
	3	5	7	9	4	
	3	5	7	9	4	
	3	5	7	4	9	Examine 5 items

After the first pass through the list, the largest number has 'bubbled' to the end of the list. In the second pass, we only need to compare the first four items.

Pass 2	3	5	7	4	9	
	3	5	7	4	9	
	3	5	4	7	9	Examine 4 items
Pass 3	3	5	4	7	9	
	3	4	5	7	9	Examine 3 items
Pass 4	3	4	5	7	9	Examine 2 items

The list is now sorted.

1. The list of animals hamster, rabbit, dog, cat, goldfish, is to be sorted in alphabetical order using a bubble sort.

   Show the state of the list after:

   (a) Pass 1 [1]

   (b) Pass 2 [1]

The bubble sort algorithm is not efficient for large lists. Note that in some cases, the algorithm may have sorted the list before performing the full number of passes. If no swaps are made during a particular pass, then the list must already be sorted.

This condition could be tested and the sorted list output without performing any more passes.

Pseudocode for the bubble sort

The algorithm for the bubble sort is given below.

```
01 // Bubble sort with flag set when no swaps are made
02 AList ← [17, 3, 7, 15, 13, 23, 20]
03 // Get number of items in the array
04 NumItems ← LENGTH(AList)
05 Comparisons ← NumItems - 1
06 SwapMade ← TRUE
07 WHILE Comparisons > 0 AND SwapMade = TRUE
08   SwapMade ← FALSE
09   FOR i ← 1 TO Comparisons
10      IF AList[i] > AList[i + 1]
11          THEN
12             Temp ← AList[i]
13             AList[i] ← AList[i + 1]
14             AList[i + 1] = Temp
15             SwapMade ← TRUE
16      ENDIF
17   NEXT i
18   Comparisons ← Comparisons - 1
19 ENDWHILE
20 OUTPUT "Sorted list:", AList
```

2. Look at the bubble sort algorithm above.

   (a) State the line numbers of the Bubble Sort code given above which swaps two items in the list. [1]

   (b) State the purpose of the variable named `Temp`. [1]

   (c) Explain the purpose of the variable named `SwapMade`. [3]

VALIDATION AND VERIFICATION

Data validation techniques are used to check the validity of data entered by the user.

Validation checks

You should be able to write simple routines to validate input data.

The following validation checks are examples of simple data validation:

Range check:	Data must lie within a given range.
Length check:	A string input must be greater than or equal to a minimum length.
Type check:	Data must be of the correct data type, e.g. integer, Boolean
Presence check:	A string should not be empty.
Format check:	E.g. a postcode must conform to one of a number of set formats.
Check digit check:	*See opposite page, and* ***Page 22***

Note that validation can only check that the data entered is sensible and reasonable. It cannot check the accuracy of the data.

For example, it can check that no data other than S, M, L or XL is entered in a field for dress size, or check that a number between –5 and 35 is entered for a temperature in a country where the temperature is always within these limits. However, an incorrect value may still be entered.

1. A user is required to enter a 6-character ID in order to register on a website.

   Complete line 02 in the pseudocode below to ensure that a valid ID has been entered. [2]

```
01 INPUT UserID
02 WHILE ..............................
03    OUTPUT "ID must be 6 characters: please re-enter: ")
04    INPUT UserID
05 ENDWHILE
06 OUTPUT "UserID accepted"
```

2. A validation check ensures that an 'age' figure entered is between 11 and 18.

   (a) State what type of validation check is this. [1]

   (b) Write pseudocode to perform this validation. [4]

Check digits

A **check digit** is an additional digit at the end of a product code designed to check that a valid product code has been entered. An example of how to calculate and check the check digit on a product code is given on Page 22.

3. A date of birth field has been validated.
   (a) Explain how the validated date of birth field may still be incorrect. [2]
   (b) Give **one** example of an invalid date of birth that should not be accepted. [1]
4. A product code with a total of 7 digits includes a check digit calculated using the modulus-11 system.
   Explain how the check digit may be used to check that the product code has been correctly read by a barcode scanner. [3]

Verification checks

Double-entry verification is the input of data twice. The two entries are compared and if they match, the entry is accepted as valid. Setting a new password commonly uses verification.

```
OUTPUT "Please enter password: "
INPUT Password
OUTPUT "Confirm password: "
INPUT PasswordAgain
IF Password = PasswordAgain
    THEN
       OUTPUT "Password accepted"
    ELSE
       OUTPUT "Invalid - Passwords don't match"
ENDIF
```

Tip: For a Describe question worth 4 marks like here, provide two answers and then give full descriptions of each.

5. Describe **two** ways to ensure that personal data input to a database is accurate. [4]

TESTING

Selecting and using suitable test data

Test data should include:

- **normal (typical) data**, using examples of typical data that the program is designed to handle
- **extreme data**, i.e. the largest and smallest acceptable value (e.g. 1–50)
- **boundary data** includes both ends of the allowed range (e.g. 1–50) as well as invalid data that should not be allowed, just outside this range. For example, if a range of 0 to 50 needs to be tested, then the boundary data would be –1, 0, 50, 51
- **abnormal** or **erroneous data**, i.e. data of the wrong type, for example non-numeric characters in a numeric field

A test plan, which includes the expected results for each test, should be drawn up even before the program is coded.

Example: John has written a program to input and validate daily temperatures recorded in a particular month. The temperature is never above 40 or below 0.

No.	Test purpose	Test data	Expected outcome	Actual outcome
1	Check normal (typical)	12	Input is accepted	
2	Check extreme data	0, 40	Input is accepted	
3	Check invalid boundary data	–1, 41	Error message displayed and user asked to enter number again	
4	Check erroneous entry	xx, 2+6	Error message displayed and user asked to enter number again	

A customer registering with an online store for the first time is required to input a new password. The password must be between 8 and 12 characters and contain at least one uppercase and one lowercase letter.

Complete the table with four tests, each with a different purpose, that could be used to check valid and invalid password entries. [4]

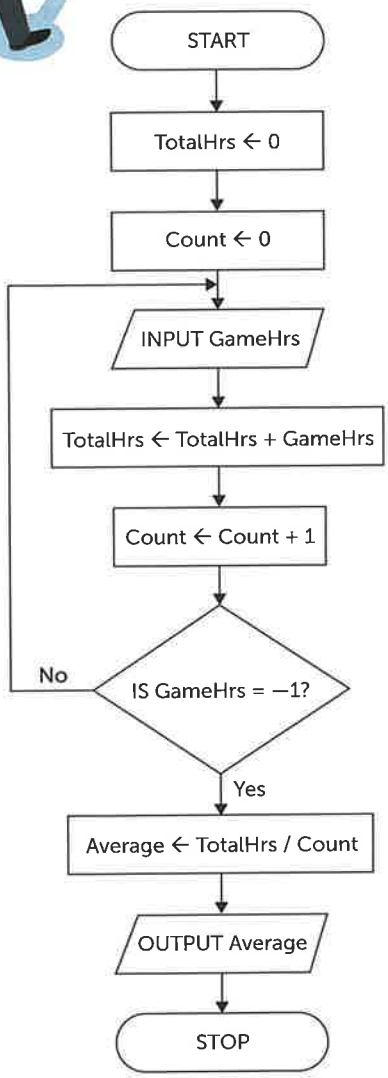
No.	Test purpose	Test data	Expected outcome
1			
2			
3			
4			

TRACE TABLES

A trace table is used to document how the values of variables change during a dry-run of an algorithm.

As each instruction is executed, the current value of any variable or logical expression that is changed is written in the appropriate column of the table below. It is not necessary to fill in a cell if the value has not changed from the line above.

Example: Ben designs a flowchart for an algorithm to calculate the average number of hours students spend per week playing computer games. He uses test data for 3 students spending respectively 8, 10 and 12 hours playing games. This should result in an average of 10 hours.



A trace table, shown below, has identified an error with the flowchart.
Describe how the algorithm could be corrected. [3]

GameHrs	TotalHrs	Count	GameHrs = –1?	Average
	0	0		
8	8	1	No	
10	18	2	No	
12	30	3	No	
–1	29	4	Yes	7.25

Oops! The algorithm must be incorrect, since it produces the wrong answer.

TYPES OF ERROR

Syntax, logic and runtime errors

There are three types of error that may prevent a program from working correctly, or from running at all.

A **syntax error** will prevent your program from running. It is caused by a mistake in the spelling or 'grammar' of your code. For example `primt("Hello World")`.

A **logic error** is harder to spot. Your program will still run but may give an incorrect or unexpected output. Common scenarios include the use of greater than or less than symbols, for example: using `X > 5` instead of `X >= 5` which could affect loop conditions or range checks. Missing brackets in mathematical calculations can also affect the logic, for example:

`VAT ← (OrderTotal - Discount) * TaxRate` gives a different answer to
`VAT ← OrderTotal - Discount * TaxRate`

Using a well thought-out test plan, with the expected results manually calculated first, should reveal any logic errors.

A **runtime error** is one which will be detected when the program is running. It may be caused by a logic error, erroneous user input, or by the program not allowing for some conditions, such as a user inputting zero, or entering no data at all. Division by zero will also cause a runtime error.

1. Look at the following pseudocode algorithm.

```
Total ← 0
X ← 0
WHILE X <> 100 DO
   Total ← Total + X
   X ← X + 3
OUTPUT "Total = ", X
```

(a) Explain what will happen when the corresponding program code is executed. [2]

(b) What type of error causes this to happen? [1]

2. The code below is intended to add the even numbers between 1 and 50.

```
Count ← 0
WHILE Count <= 50 DO
   Count ← Count + 2
   Sum ← Sum + Count
OUTPUT Total, Sum
```

Find **three** errors in the code. State in each case whether each is a syntax error, a logic error which will result in a wrong answer being output, or an error which will cause execution to be abandoned with an error message (runtime error). [6]

EXAMINATION PRACTICE

1. A pseudocode algorithm is given below.

```
01  AList ← [3,6,7,9,13,15,16,19,20,24,26,29,36]
02  Found ← False
03  N ← 1
04  INPUT X
05  WHILE Found = FALSE AND N <= LENGTH(AList) DO
06    OUTPUT aList[N]
07    IF AList[N] = X
08      THEN
09        Found ← TRUE
10      ELSE
11        N ← N + 1
12    ENDIF
13  ENDWHILE
14  IF Found = TRUE
15    THEN
16      OUTPUT X, N
17    ELSE
18      OUTPUT "Invalid number"
19  ENDIF
```

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

(b) The user enters 9 at line 04. What is printed at line 06 the first 3 times the WHILE...ENDWHILE loop is performed? [1]

(c) State what will be printed at line 16 if the user enters the number 9. [1]

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

2. An algorithm for a bubble sort is given below.

```
01  Names ← ["Edna ", "Adam ", "Victor ", "Charlie ", "Jack ", "Ken", "Maria"]
02  N ← LENGTH(Names)
03  Comparisons = N - 1
04  SwapMade ← TRUE
05  WHILE Comparisons > 0 AND SwapMade = TRUE DO
06    SwapMade ← False
07    FOR Index ← 1 TO Comparisons
08      IF Names[Index] > Names[Index + 1]
09        THEN
10            <swap the names>
11
12
13            SwapMade ← TRUE
14      ENDIF
15    NEXT Index
16    Comparisons ← Comparisons - 1
17  ENDWHILE
18  OUTPUT Names
```

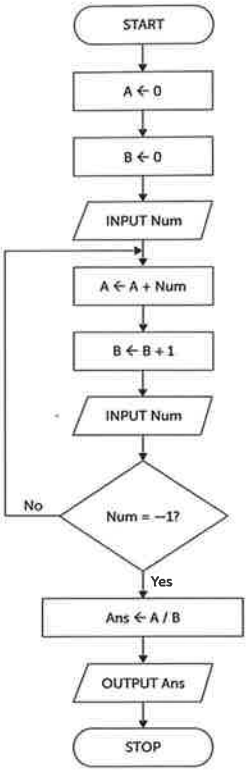
(a) Explain the purpose of the variable **SwapMade** in the algorithm. [2]

(b) Write the pseudocode statements for **<swap the names>** at lines 10–12. [3]

(c) Write the contents of the array after each of the first two times the WHILE loop is executed. [2]

(d) Explain how many times the WHILE loop will be executed before the program terminates. [2]

3. (a) An algorithm is given below.



Complete the trace table to show how the variables change, and what will be output, if the numbers 3, 8, 2, 5, –1 are entered. [5]

Num	A	B	Ans
	0	0	
3	3	1	
8			

(b) State the purpose of the algorithm. [1]

4. (a) Andrew enters a PIN to secure his new phone.
The pseudocode shown below checks the PIN.

```
INPUT PIN
D ← 0
X ← PIN
WHILE X >= 1 DO
   D ← D + 1
   X ← X/10
ENDWHILE
IF D <> 4
   THEN
      OUTPUT "PIN INVALID"
   ELSE
      OUTPUT "PIN ACCEPTED"
ENDIF
```

(a) Complete the trace table below to show what would be output if Andrew enters **52390**. [5]

(b) What type of validation check does the algorithm perform? [1]

PIN	D	X	OUTPUT
52390	0	52390	

5. The pseudocode algorithm below is intended to find the largest of 10 positive integers entered by a user. There are errors in the code.
Identify the errors and state what type of error each one is. [4]

```
01 MaxNumber ← 1000
02 FOR N ← 1 TO 10
03    INPUT Num
04    IF Num > MaxNumber
05      THEN
06        Num ← MaxNumber
07    ENDIF
08    NEXT N
09 OUTPUT MaxNum
```

6. Joanna has written a program which allows a user to enter the percentages obtained by each student who sat an exam, and outputs the highest mark, lowest mark and average mark.
Give **four** items of test data that she could use to test the validation of input data.
Specify the purpose of each test and the expected outcome. [8]

VARIABLES, CONSTANTS, ASSIGNMENTS

Data used in a program is stored in memory locations while the program is running. A **variable** is a data item held in memory which may change value during program execution. It is referred to by its **identifier** (variable name).

Data types, variables and constants

The table below shows the different data types found in programming languages such as Python or VB.

Note that in the IGCSE pseudocode, strings will be delimited by double quotes, and single characters by single quotes.

Data type	Type of data	Examples
INTEGER	A whole number	5, –267, 153489
REAL	A number with a decimal point	3.142, 56.0, –0.75
CHAR	A single character enclosed in single quotes	'A', '#', '6', '&'
STRING	Zero or more characters enclosed in double quotes	"Yes", "Hi, John"
BOOLEAN	The logical values TRUE and FALSE	TRUE, FALSE

Some programming languages allow the use of **constants**. The value of a constant cannot change during the execution of the program. A constant can be declared in pseudocode like this:

```
CONSTANT VAT ← 0.2
```

A **variable name** (**identifier**) can be a mixture of letters and numbers, but should start with a letter. Uppercase letters and lowercase letters may be used. The **value** of a variable (not its name) can change during program execution. Although variable names often start with a lowercase letter, in the pseudocode used by IGCSE, they will start with an uppercase letter, and will be considered case-insensitive.

Pascal case is used in IGCSE to separate words making up a variable name; for example, `TotalCost`, `StudentName`.

Using a naming convention such as this helps to reduce errors in referring to variables in a program. If no convention has been used, it can be difficult to remember whether, for example, a total cost is named `totalCost`, `totalcost`, `TotalCost`, `total_cost` or something else. It also aids other programmers who may need to update a pre-existing program.

Assignments

A **variable** is **assigned** a value in pseudocode using the ← sign.

```
CostPrice ← 15.65
Count ← Count + 3
Under12 ← True
StudentName ← "Higgins, P"
Response ← 'Y'
```

A game is being programmed. Use pseudocode to answer the following questions.

1. (a) Choose a meaningful variable name for the highest score in the game. [1]
   (b) Set the highest score to 25. [1]
2. (a) Choose a meaningful variable name for the name of a player. [1]
   (b) Set the player's name to "Santos". [1]

INPUT / OUTPUT

When data needs to be input, the user is typically prompted to type something, and whatever they type is assigned to a variable.

Example

In pseudocode:

```
OUTPUT "Please enter your name: "
INPUT Name
```

When writing an algorithm using pseudocode, the prompt given by the OUTPUT statement is typically assumed, and therefore omitted.

The OUTPUT statement is used to output text or data to the screen

Example: Write pseudocode which asks the user to enter two numbers representing the length and width of a rectangle, then calculates and prints the area.

```
INPUT Length
INPUT Width
OUTPUT "Area = ", Length * Width
```

1. Write a pseudocode algorithm to ask the user to enter their first name and their last name. Then output on one line their last name followed by their first name. [4]
2. Write a pseudocode algorithm to allow the user to enter a series of numbers between 0 and 100. The end of the data is signalled by an entry of –1. Output the maximum of the numbers and the average of all the numbers. [8]

ARITHMETIC AND LOGICAL OPERATORS

Arithmetic operators	
Symbol	**Meaning**
+	Addition
–	Subtraction
/	Division
*	Multiplication
^	Raised to the power of
MOD	Modulus
DIV	Quotient

Logical operators	
Symbol	**Meaning**
=	Equal to
<	Less than
<=	Less than or equal
>	Greater than
=>	Greater than or equal
<>	Not equal to

MOD returns the remainder when one integer is divided by another. For example,

```
A ← 22 MOD 5  will assign the value 2 to A
B ← 15 MOD 5  will assign the value 0 to B
```

DIV returns the quotient

```
C ← 22 DIV 5  will assign the value 4 to C
D ← 15 DIV 5  will assign the value 3 to D
```

Normal rules of precedence mean that in an expression such as

$$10 + 3 * (18 - 2) / 2 ^ 3$$

the operations are carried out in the order: Parentheses (brackets), Exponents, Multiplication and Division, Addition and Subtraction. The result of this expression is 16.

Boolean expressions

A **Boolean expression** uses one or more logical operators and evaluates to either **TRUE** or **FALSE**.

Evaluate the following Boolean expressions:

(a) (2 + 5 * 3) > (2 + 5) * 3 [1]

(b) 25 DIV 4 >= 25/4 [1]

(c) "Fred" <> "fred" [1]

SEQUENCE AND SELECTION

There are three basic control structures in all high level imperative languages such as Python, C# or VB.NET. These are **sequence**, **selection** and **iteration**.

Sequence

Sequence is simply two or more statements written and executed one after the other in sequence.

Selection

A **selection** statement comprises an **IF** or **CASE** statement and a **logical expression**. Variables of the Boolean data type can only be TRUE or FALSE.

Boolean operators

- The logical operators **AND** and **OR** are used to combine two or more conditions.
- The logical operator **NOT** may be used to test whether a condition is NOT satisfied.

1. State the values of `ValidName1` and `ValidResult`, given that `Name = "Mia"`

   (a) `ValidName1 ← (Name = "Meera") OR (Name = "Mia")` [1]

   (b) `ValidResult ← (1 < 4) AND (7 <= 7)` [1]

2. What will be output by the following statements? [1]

```
A ← 5
B ← 2 * A
C ← A + B
IF A = B OR B <= C
   THEN
      OUTPUT "True"
   ELSE
      OUTPUT "False "
ENDIF
```

3. Given that A = 3, B = 3, C = 7 and D = 8, what will be output by the following statements? [1]

```
X ← NOT (A = B AND C <> D)
OUTPUT X
```

4. Write a Boolean condition to test whether `Result` is between 1 and 10. [2]

5. Write a pseudocode statement to output **"X is valid"** if X is divisible by 7, but not divisible by 3. Otherwise, print the message **"X is invalid"**. [3]

IF AND CASE STATEMENTS

Selection statements

Selection statements include:
IF ... THEN ... ENDIF, **IF ... THEN ... ELSE ... ENDIF** and **CASE OF... OTHERWISE ... ENDCASE** statements.

Nested IF statements

An **IF statement** may be **nested** inside another IF statement. For example, suppose we wanted to output the largest of three numbers entered. The pseudocode could be written:

```
IF Num1 >= Num2 AND Num1 >= Num3
    THEN
       OUTPUT Num1
    ELSE
       IF Num2 >= Num1 AND Num2 >= Num3
          THEN
             OUTPUT Num2
          ELSE
             OUTPUT Num3
       ENDIF
ENDIF
```

The CASE statement

CASE statements allow one of several branches of code to be executed, depending on the value of a variable.

```
INPUT MemberType
CASE OF MemberType
    "Junior"     : EntryFee ← 2.0
    "Senior"     : EntryFee ← 3.0
    "Special"    : EntryFee ← 0.0
    OTHERWISE OUTPUT "Invalid member type" //Other conditions not met
ENDCASE
```

Write a selection statement to test an integer variable called `Mark`. If `Mark` is 75 or over, output `"Distinction"`. If `Mark` is between 60 and 74, output `"Pass"`. Otherwise, output `"Fail"`. [4]

COUNT-CONTROLLED LOOPS

Iteration means repetition. Statements to be repeated are placed inside a loop structure. A **FOR ... NEXT** loop is a **count-controlled loop**. A counter is automatically incremented each time the loop is performed.

Example

This code will print all the numbers between 1 and 10:

```
FOR Count ← 1 TO 10
    OUTPUT Count
NEXT Count
```

An increment for the counter can be specified. The following code will output the numbers 2, 5, 8.

```
FOR Count ← 2 TO 10 STEP 3
    OUTPUT Count
NEXT Count
```

1. Complete the following pseudocode, which allows a user to enter a start number and an end number, and outputs all the numbers in between which are divisible by either 3 or 7 or both. [4]

```
INPUT StartNum
INPUT EndNum
FOR N ......................
   IF ..................
     THEN
        ..................
   ENDIF
.............
```

Nested iteration

You can have one loop nested inside another.

Example: Display all the multiplication tables between 2 and 10.

```
FOR Table ← 2 TO 10
    FOR N ← 1 TO 10
       Answer ← Table * N
       OUTPUT Answer
    NEXT N
NEXT Table
```

2. In the code to the left, how many times will the OUTPUT statement be executed? Explain your answer. [2]

CONDITION-CONTROLLED LOOPS

WHILE...DO...ENDWHILE and **REPEAT...UNTIL** are examples of **indefinite**, or **condition controlled**, iteration.

WHILE...DO...ENDWHILE loop

A `WHILE...DO...ENDWHILE` loop is a **pre-condition loop**, controlled by a Boolean condition which is checked **before** the loop is entered.

Example

The algorithm below calculates the average number of visitors per day at an attraction over a number of days.

In the algorithm below, if `Visitors = -1` before entering the `WHILE ... ENDWHILE` loop, none of the statements in the loop will be executed. `Visitors` and `Days` will be 0 and the program will crash with an **runtime error** when it reaches the last line.

```
Total ← 0
Days ← 0
INPUT Visitors
WHILE Visitors <> -1 DO
    Total ← Total + Visitors
    Days ← Days + 1
    INPUT Visitors
ENDWHILE
Average ← Total / Days
OUTPUT "Average daily number of visitors"
OUTPUT Average
```

REPEAT...UNTIL loop

A REPEAT...UNTIL loop is a **post-condition loop**, controlled by a Boolean condition which is checked at the *end* of the loop. It is therefore always performed at least once.

1. Explain what will happen when the following pseudocode is coded and executed: [2]

```
Total ← 0
X ← 0
WHILE X <> 100 DO
   X ← X + 3
   Total ← Total + X
ENDWHILE
OUTPUT "Total = ", X
```

2. (a) Which of the following gives the number of times the REPEAT loop given below is performed? [1]

A: 9 B: 10 C: 500

```
Number ← 1
REPEAT
   Number ← Number * 2
   OUTPUT Number
UNTIL Number > 1000
```

(b) Rewrite the algorithm using a WHILE loop. [2]

STRING HANDLING

A **string** is defined as zero or more characters enclosed in quote marks, for example "My name is Jodie Bell". An empty string is written "".

A single character is referred to in IGCSE pseudocode as a **char** and is delimited by single quotes, e.g. 'a', '&', '7'.

String indexing

Each character in a string can be referenced by its index, starting at 0 or 1, depending on the programming language used. The starting index will always be 1 unless specified otherwise in any exam question on strings.

Example

```
DECLARE StudentName : STRING
StudentName ← "Hatty Gibson"
```

The first character of the string is `StudentName[1]`.

`StudentName[1]` contains `"H"` and `StudentName[7]` contains `"G"`.

String operations (functions)

Operation	Example	Returns
`LENGTH()`	`LENGTH("The cow jumped over the moon")`	`28`
`LCASE()`	`LCASE("ABCdef123")`	`abcdef123`
`UCASE()`	`UCASE("Sabrina Ngozi")`	`SABRINA NGOZI`
`SUBSTRING()`	`SUBSTRING("Sabrina Ngozi",4,7)`	`rina Ng`

1. Write a pseudocode algorithm which allows a user to enter a name consisting of a firstname and a surname, and outputs the name in uppercase letters and the length of the name. [3]
2. Write a pseudocode algorithm which allows a user to enter a product code, for example AFG191214796 and outputs characters 4–6 and the last character of the code. [3]

PROCEDURES AND FUNCTIONS

A procedure is a named block of code, separate from the main program, which can be called and executed using a statement of the form: `CALL <procedure name>`.

Procedure

A procedure does not return a value. Using pseudocode, control is passed to the procedure using the `CALL` statement, with any **parameters** in brackets. These parameters are passed to the procedure, and placed in variables specified in the procedure heading.

1. Write a statement to call the procedure greeting(aName), passing it the parameter "Helen". What will be output? [2]

```
PROCEDURE greeting(aName:STRING)
   OUTPUT "Hello " + aName
ENDPROCEDURE
```

Function

A **function** is another type of subroutine. It returns a value to a variable specified in the statement which calls it. Programming languages have many built-in functions, similar to the pseudocode string functions `LENGTH()`, `LCASE()` and `SUBSTRING()` shown on the previous page. Notice how these functions are called:

`SLength ← LENGTH("This is a string")` will return 16 in `SLength`.
`Chars4To6 ← SUBSTRING(ProductCode,4,3)`

Parameters

Parameters are values passed to the subroutine, declared in parentheses within the call instruction.

2. Name the parameters passed to the functions (a) `LENGTH()` and (b) `SUBSTRING()` in the examples above. [2]

Example

This function converts a Celsius temperature to Fahrenheit:

```
FUNCTION ConvertToF(CTemp) RETURNS REAL
   DECLARE FTemp : REAL
   FTemp ← (CTemp * 9/5) + 32
   RETURN FTemp
ENDFUNCTION
```

To call the function and assign the Fahrenheit temperature to `TempInF`:

```
INPUT CelsiusTemp
TempInF ← ConvertToF(CelsiusTemp) //This is the call statement
```

Tip: Procedures and functions are essential tools in decomposition – breaking down a complex problem into smaller sub-problems or sub-tasks. ***(See Pages 67 and 92)***

LOCAL AND GLOBAL VARIABLES

All variables have a scope which defines the parts of the program in which they are recognised. In the function `ConvertToF()`, `FTemp` is a local variable. It is not recognised outside the subroutine and if you try to use it in the main program, you will get an error message.

The scope of a **global variable** is the whole program including all its subroutines.

Example

The following function returns the average of two numbers `x` and `y` passed as parameters.

```
FUNCTION Average(x:REAL, y:REAL) RETURNS REAL
    DECLARE Total, Avg:REAL  //Declare local variables, Total, Avg
    Total ← x + y
    Avg = Total / 2
    RETURN Avg
ENDFUNCTION
```

The function could be called with the statement

```
Mean ← Average(317, 654)
```

`Total` and `Avg` are local variables inside the function. If you try to print either of them outside the function, you will get an error message.

LIBRARY ROUTINES

The ROUND function

`X ← ROUND(Num1,3)` returns `Num1` rounded to 3 decimal places and assigns it to `X`.

For example:

```
X ← 5.2651
Y ← ROUND(X,2) returns 5.27 in Y
```

Random number generation

The `RANDOM()` function generates a random floating point number between 0 and 1 inclusive.

Write a program that generates 10 random real numbers between 0 and 10, rounds each of them to 2 decimal places, and prints both the original and the rounded numbers. [4]

Note that MOD and DIV are arithmetic operators, not functions. **(See Page 84)**.

CREATING A MAINTAINABLE PROGRAM

Whether a program consists of just a few lines, or contains tens of thousands of lines, it's important to make it as easy to read and understand as possible.

This helps to create a **maintainable program** which can be amended if necessary and used over and over again.

Meaningful identifiers

Meaningful identifiers should be used for variables, constants, arrays, procedures and functions. Having meaningful identifiers using a **consistent format** helps readability and maintainability, e.g. `GrandTotal`, `YearJoined`, `ConvertToCelsius`.

Commenting

Commenting should be used to document for yourself and others:

- program name
- the purpose of the program
- complex areas of code and the purpose of each procedure and function used.

This helps ensure that the program can still be understood and maintained long after it was written. It should also indicate who wrote the program and when.

Rewrite the pseudocode below to make it easier to understand. Assume that the values entered represent a set of exam marks between 0 and 100. [4]

```
X ← 0
INPUT Y
N ← 1
WHILE Y <> -1 DO
   X ← X + Y
   N ← N + 1
   INPUT Y
ENDWHILE
Z ← X / N
OUTPUT Z
```

Decomposition

Decomposition of a long, complex program using **subprograms** (**procedures and functions**) makes debugging and maintaining a program easier:

- Subprograms are usually no more than a page of code
- Subprograms can be tested separately and shown to be correct
- A particular subprogram can be used several times in the same program, and saved in a subroutine library to be used in other programs

ARRAYS

An **array** is a data structure used to hold several elements of the same data type.

Each element of the array can be accessed by its index number, with the index of the first element (in IGCSE pseudocode) being 1.
An array index may start at either 0 or 1 depending on the programming language.

Example 1

Using pseudocode, an array named `StudentName` which will hold the names of 10 students could be declared like this:

```
DECLARE StudentName : ARRAY[1:10] OF STRING
```

To assign the name `"Morris, J"` to the third element of the array we could write

```
StudentName[3] ← "Morris, J"
```

Example 2

Write pseudocode for a program to enter the number of customers visiting a shop each day of the week, and then print out the number of customers each day and the total number of customers for the week.

```
01  DECLARE Day : ARRAY[1:7] OF STRING
02  DECLARE Customers : ARRAY[1:7] OF INTEGER
03  Day ← ["Sun","Mon","Tue","Wed","Thu","Fri","Sat"]
04  TotalCustomers ← 0
05  FOR N ← 1 TO 7
06     OUTPUT(Day[N])
07     INPUT Customers[N]
08     TotalCustomers ← TotalCustomers + Customers[N]
09  NEXT N
10  FOR N ← 1 TO 7
11     OUTPUT Day[N], Customers[N]
12  NEXT N
13  OUTPUT "Total Customers ", TotalCustomers
```

1. What will be output at line 06 the third time the `FOR...NEXT` loop is executed? [1]
2. Define an array called `numbers` holding the numbers 37, 76, 55, 91, 23. Write pseudocode for a program to reverse the order of the numbers, storing them in a second array called `ReverseNumbers`.
   Print out the contents of `ReverseNumbers`. [5]

TWO-DIMENSIONAL ARRAYS

Example

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

Four quarters (Jan–March, April–June, July–Sep, Oct–Dec)

	Index	1	2	3	4
Three branches A, B, C	1	56	87	92	43
	2	167	206	387	54
	3	22	61	52	14

The index for both row and column of the array starts at 1. The array is defined with the statement:
`DECLARE Sales : ARRAY[1:3, 1:4] OF INTEGER`.

The number of properties sold by Branch A in Quarter 4 is held in `Sales[1,4]` with the value 43.

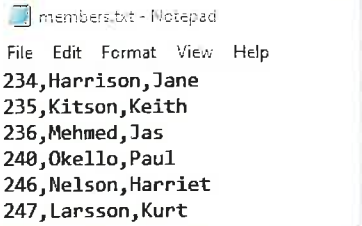
1. The three branches of the estate agency are known as Branch A, Branch B and Branch C.
   (a) Write pseudocode to output the sales figure for Branch C for the period April–June. [1]
   (b) What will be output? [1]
2. Write a program to ask a user to enter the name and five race times in seconds for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

FILE HANDLING

Typically, data such as student names and details, sales records or daily weather readings need to be stored permanently in non-volatile storage such as a hard disk or SSD. In this case, the data is written to a **file**. The file can then be read and the data used in many different programs.

Text files

Text files contain text in lines. Each line of the text file comprises a **record**, and the different **fields** in the record are separated by commas. For example, a file containing the membership number, surname and first name of club members could look like this when stored in a text file:



Creating, opening, writing to and closing a file

The pseudocode below creates a new file and writes records to it.

```
OPENFILE MemberFile.txt FOR WRITE          // Opens a new file
INPUT MemberID, Surname, Firstname
WHILE MemberID <> "END" DO
    MemberRecord ← MemberID + "," + Surname + "," + Firstname
    WRITEFILE Memberfile.txt, MemberRecord
    INPUT MemberID, Surname, Firstname
ENDWHILE
CLOSEFILE MemberFile.txt
```

Reading and printing all records in a file

```
OPENFILE MemberFile.txt FOR READ          //open an existing file
READFILE Memberfile.txt,MemberRecord      //read the first record
WHILE NOT Memberfile.endOfFile DO         //while not end of file
    OUTPUT MemberRecord                   //print record
    READFILE Memberfile.txt,MemberRecord  //read next record
ENDWHILE
CLOSEFILE MemberFile.txt                  //close the file
```

Write a pseudocode algorithm to enter the 10 most popular films of the previous year and store them in a text file. You should create a new file named "Films.txt" in WRITE mode, add the records and then close the file. [5]

EXAMINATION PRACTICE

1. Write an algorithm in pseudocode to input 20 numbers, and output numbers which are NOT between 97 and 99 inclusive with a message "Not in normal range". Numbers inside this range are output with no message. [5]

2. Explain what happens when a program based on the following pseudocode algorithm is executed, if the user enters –1 at line 4. [3]

```
01  TotalPrice ← 0
02  NumItems ← 0
03  OUTPUT "Enter price of next item, -1 to finish data entry"
04  INPUT Price
05  WHILE Price <> -1 DO
06    TotalPrice ← TotalPrice + Price
07    NumItems ← NumItems + 1
08    OUTPUT "Enter price of next item, -1 to finish data entry"
09    INPUT Price
10  ENDWHILE
11  AveragePrice ← TotalPrice / NumItems
12  OUTPUT "Average item price", AveragePrice
```

3. To determine whether a year is a Leap Year, with 29 days in February, the following rules are applied:
   - The year must be evenly divisible by 4 *but...*
   - if the year can also be evenly divided by 100, it is ***not*** a Leap Year,

   unless...
   - the year is also evenly divisible by 400, in which case it ***is*** a Leap Year.

   The pseudocode algorithm below will then output the year and whether it is or is not a leap year.

   According to these rules, the years 1600, 2000 are Leap Years, while 1800, 1900, 2100, are not Leap Years.

   Complete LINE A, LINE B and LINE C in the pseudocode algorithm below to determine whether a year input by the user is a Leap Year. [3]

```
// Accept a year entered by the user
INPUT Year
// Check if this is a Leap Year
IF MOD 4 = 0 AND  ..............................    //(LINE A)
 THEN
    OUTPUT Year, "is a Leap Year"
 ELSE
    IF MOD 400 = 0
      THEN
        OUTPUT Year, ..............................        //(LINE B)
      ..........                                           //(LINE C)
        OUTPUT Year, "is not a Leap Year"
    ENDIF
ENDIF
```

4. The following algorithm inputs the age of people applying to take a driving test. An input of –1 signals the end of data. The program outputs the average age of all applicants, and the number of people aged 25 and over who applied.

```
TotalAge ← 0
Applicants ← 0
Over25 ← 0
INPUT Age
WHILE Age <> -1 DO
   Applicants ← Applicants + 1
   TotalAge ← TotalAge + Age
   IF Age >= 25
      THEN
         Over25 ← Over25 + 1
   ENDIF
   INPUT Age
ENDWHILE
AverageAge ← TotalAge / Applicants
OUTPUT AverageAge, Over25
```

Complete the trace table for the following input data:

17, 20, 27, 34, 22, –1 [5]

TotalAge	Applicants	Over25	Age	AverageAge	OUTPUT
0	0	0	17		

5. The following pseudocode is intended to allow the user to enter a set of positive numbers between 1 and 100, and output the smallest number. An entry of –1 signals the end of data entry.

```
01  LowNum ← 0
02  WHILE X <> -1 DO
03    INPUT X
04    IF X < LowNum
05      THEN
06        X ← LowNum
07    ENDIF
08  ENDWHILE
09  OUTPUT LowNum
```

There are errors in the pseudocode. Locate the errors and write an amended algorithm. [4]

6. The following partially completed program validates a user password. The user is permitted three attempts to enter the password correctly before being locked out.

```
01  SavedPassword ← "Black&Bird34"
02  ValidPassword ← FALSE
03  Attempts ← 0
04  WHILE validPassword = FALSE AND ..................... DO
05    INPUT UserPassword
06    IF UserPassword .........................
07      THEN
08            ValidPassword ← TRUE
09            Attempts ← 3
10      ELSE
11            Attempts ← ........................
12            OUTPUT "Invalid password"
13  ENDWHILE
14  IF ........................
15    THEN
16      OUTPUT "Password correct… please continue"
17    ELSE
18      OUTPUT "Password incorrect… locked out"
19  ENDIF
```

Complete lines 04, 06, 11, 14 [4]

7. Write a pseudocode algorithm which allows a user to enter their firstname and surname, e.g. *Ken Harden*. It should then output in uppercase letters, the surname followed by initial, e.g. *HARDEN K*. (Use inbuilt functions `UCASE(<identifier>)`, `SUBSTRING(<identifier>, <start>, <length>))` [3]

8. For each of the following activities, state whether they are examples of abstraction, decomposition or algorithm design.
   (a) Planning a series of step-by-step instructions specifying how a computer will solve a problem. [1]
   (b) Identifying the sub-problems involved in solving a problem, and further breaking down the sub-problems into ones which can be solved more easily. [1]
   (c) Removing unnecessary details of a problem that are not relevant to a solution. [1]

9. (a) Complete the pseudocode procedure shown below named `Swap()` to swap two string variables X and Y.

```
PROCEDURE Swap(String1: STRING, String2: STRING)
  DECLARE Temp: String
  (add code here)
ENDPROCEDURE
```

[3]

   (b) Name a local variable used in the procedure. [1]
   (c) A program calls the procedure to swap two names in an array named `Pupil` of 20 names. Write a statement to declare this array. [1]
   (d) Write a statement to call the procedure `Swap()` to swap `Pupil[5]` and `Pupil[6]`. [1]

10. The function `RollDice(1,6)` returns a random integer between 1 and 6.

    Write a pseudocode algorithm for a program which simulates a 6-sided dice being thrown one hundred times. The number of times each face (1–6) appears should be saved in an array declared with the statement: `DECLARE TotalTimesThrown : ARRAY [1:6] OF INTEGER`

    The contents of the array are then output.

    (You do not have to write pseudocode for the function `RollDice()`.) [6]

11. Describe **three** ways in which you can create a program which is easy to understand and maintain. [6]

12. (a) Explain the purpose of storing data in a file to be used by a program. [2]

    (b) Explain the purpose of lines 01, 03, 04, 06 in the following pseudocode. [4]

```
01   OPENFILE Mammals.txt for READ
02   WHILE NOT END_OF_FILE DO
03     READFILE Mammals.txt, MammalRec
04     OUTPUT MammalRec
05   ENDWHILE
06   CLOSEFILE Mammals.txt
```

13. Write a pseudocode algorithm which inputs 100 numbers and outputs the total and average of the numbers. [6]

DATABASES

A **database** is a way of holding data in an organised way so that searching for data items meeting certain criteria is quick and easy.

Tables, records and fields

A database consists of one or more **tables**. Each table consists of many **records** (rows) each having an identical record structure. Each **field** (column) in a record has a defined field type such as **text/alphanumeric**, **character**, **Boolean**, **integer**, **real** or **date/time**.

Each table will have a **primary key** field that uniquely identifies each record in the table. In this table, the primary key is **ID**.

Volcano

ID	Name	Country	LastErupted	TimesErupted	ElevationMetres
1	Taal	Philippines	2020	7	311
2	White Island	New Zealand	2019	39	321
3	Shiveluch	Russia	2019	21	3283
4	Anak Krakatoa	Indonesia	2018	21	813
5	Eyjafjallajökull	Iceland	2010	2	2119
6	Etna	Italy	2021	57	3326
7	Stromboli	Italy	2019	1500	924
8	Puyehue-Cordón Caulle	Chile	2011	8	2236

The table **Volcano** above, records volcanic eruptions and the number of times each one has erupted in recent times.

(a) State how many records are featured in the **Volcano** table. [1]

(b) State how many fields are in the **Volcano** table. [1]

(c) Suggest suitable data types for the data in the **Country**, **LastErupted** and **ElevationMetres** columns. [3]

STRUCTURED QUERY LANGUAGE (SQL)

Records in a database format can be searched using Structured Query Language (SQL).

The format of an SQL statement

SELECT... list the field(s) you want displayed here
FROM... list the table or tables the data will come from here
WHERE... list search criteria here
ORDER BY... optional criteria to sort in ascending (ASC) or descending (DESC) order.

Using the **Volcano** table, the SQL statement below will return a **Results** table showing all eruptions since 2019, in alphabetical order of name.

SELECT Name, lastErupted, TimesErupted
FROM Volcano
WHERE LastErupted >= 2019
ORDER BY name **ASC**

Name	LastErupted	TimesErupted
Etna	2021	57
Shiveluch	2019	21
Stromboli	2019	1500
Taal	2020	7
White Island	2019	39

You can also use Boolean operators AND and OR in search criteria.

Example

Find all volcanoes with an elevation of less than 500 metres which have erupted since 2019. Display the results in descending order of the date they last erupted, i.e. the volcano which erupted most recently should appear first in the list.

SELECT Name, Country, LastErupted, ElevationMetres

FROM Volcano

WHERE LastErupted >= 2019 **AND** ElevationMetres < 500

ORDER BY LastErupted **DESC**

RESULTS table

Name	Country	LastErupted	ElevationMetres
Taal	Philippines	2020	311
White Island	New Zealand	2019	321

VALIDATION

When a database structure is created, **validation rules** may be assigned to any of the fields. Validation checks on each field may include a **length check**, a **format check**, a **validation rule** such as **"M"** or **"F"**, a **range check** and a **presence check**.

A table named **ClubMember** is shown below.

ClubMember

MemberID	Surname	Firstname	Gender	MemberType	DateJoined	SubPaid
0001	Campbell	Sarah	F	S	18/01/2019	Y
0005	Mansur	Rajeev	M	S	23/07/2019	N
0035	Jolly	Helen	F	J	01/09/2020	Y
0036	Gurman	Aaron	M	S	01/09/2020	Y
0150	Bull	Sean	M	J	05/07/2021	N
0152	Mason	Dina	F	S	08/07/2021	Y

1. Suggest what validation checks could be carried out when inputting data into the **ClubMember** table.
2. (a) What is the purpose of a primary key in a database? [2]
   (b) Identify the primary key in the ClubMember table. [1]
   (c) Identify the data type of the DateJoined field. [1]

SUM and COUNT statements

To find the total number of female members in the **ClubMember** table:

```
SELECT COUNT(*)
FROM ClubMember
WHERE Gender = "F"
```

To find the total number of eruptions of volcanoes in Italy recorded in the **Volcano** table:

```
SELECT SUM(TimesErupted)
FROM Volcano
WHERE Country = "Italy"
```

3. What will be the result of the SUM query above? [1]

EXAMINATION PRACTICE

1. A database table **Animal** is given below.

Animal

ID	Name	HeartRateBPM	LifeSpanYears	AnimalClass	SpeedKmH
BW1	Blue Whale	8	90	Mammal	52
GF20	Giraffe	150	25	Mammal	60
HA03	Hamster	400	2	Mammal	6
HU24	Hummingbird	1200	5	Bird	79
HU78	Human	75	79	Mammal	45

(a) Give the name of the field that would be used as the primary key. [1]

(b) Write an SQL query to find the number of records in the database table. [1]

(c) Write an SQL query to return the Name, LifeSpanYears and AnimalClass of all mammals with a lifespan of 5 years or more, in alphabetical order of Name. [4]

(d) List the names of the animals which will be returned by the query. [1]

2. A database table named **Product** is shown below.

Product

ProductCode	Description	Price	OnSale	SalePrice	QtyInStock	Department
GF165	Garden chair	54.00	Yes	45.00	10	Garden
GF201	Small table	499.00	No		5	Garden
GF230	Waterproof cushion	23.99	No		15	Garden
GM335	Picnic basket	75.00	Yes	39.00	4	Garden
HM212	Uplighter	125.00	Yes	99.00	5	Home
HM578	Large table	750.00	No		2	Home

(a) Complete the table below to show the most appropriate data type for each field in the database. [3]

Field	Data type
ProductCode	
Description	
Price	
OnSale	
SalePrice	
QtyInStock	

(b) Write an SQL query to return the total quantity in stock of products on sale in the Garden department. [2]

(c) State the result of the query. [1]

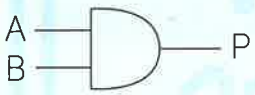
(d) Name and give an example of a validation check that could specified for **QtyInStock**. [2]

BOOLEAN LOGIC

Simple logic gates

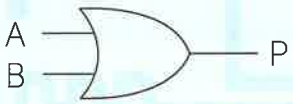
The electronic circuits in a computer are constructed from **logic gates** which can only be in one of two states: on or off, 1 or 0. Three simple logic gates are shown below. Each is represented by a diagram and a **truth table** showing the possible outputs for each possible input.

AND gate



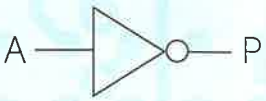
AND		
A	**B**	**P = A AND B**
0	0	0
0	1	0
1	0	0
1	1	1

OR gate



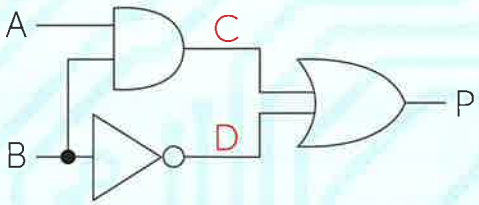
OR		
A	**B**	**P = A OR B**
0	0	0
0	1	1
1	0	1
1	1	1

NOT gate



NOT	
A	**P = NOT A**
0	1
1	0

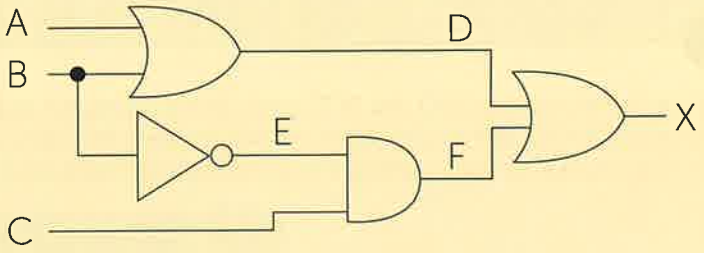
Logic gates can be combined to produce more complicated circuits. This circuit (logic diagram) can be represented by the logic expression: P = (A AND B) OR (NOT B).



The truth table is given below.

A	B	C (A AND B)	D (NOT B)	P (C OR D)
0	0	0	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

1. Below is a logic diagram.



(a) Write the logic expressions for D, E and F. [3]

(b) Write the logic expression corresponding to the logic diagram, in terms of inputs A, B and C and output X. Show your working. [4]

2. A logic circuit is being developed for a bus shelter advert that plays automatically if a passenger is detected in or around the bus stop.

- The system has two sensors, S1 and S2, that detect if a passenger is near. The advert plays if either of these sensors is activated.
- The advert should only play if it is not daytime (D).
- The output from the circuit, for whether the advert should play or not, is P.

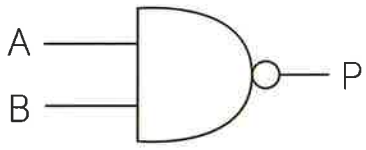
(a) Write the logic expression for this system. [2]

(b) Complete the logic circuit for this system. [3]

NAND, NOR AND XOR (EOR) GATES

NAND gate

The NAND gate is a combination of the AND and NOT gates, which inverts the output of the AND gate. Having a single type of NAND gate that can perform two separate functions can help to reduce development costs if a NAND gate is cheaper than separate AND and NOT gates.

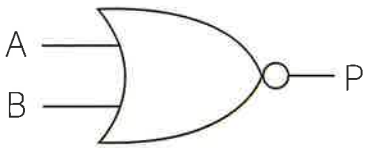


P = NOT (A AND B)

A	B	P = A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

NOR gate

The NOR gate returns true only when both inputs are false.

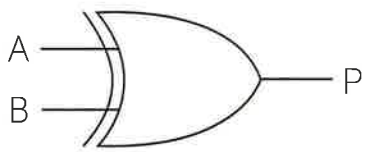


P = NOT (A OR B)

A	B	P = A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

XOR (EOR) gate

The exclusive OR gate returns true if one input is true, but not both.



P = A XOR B

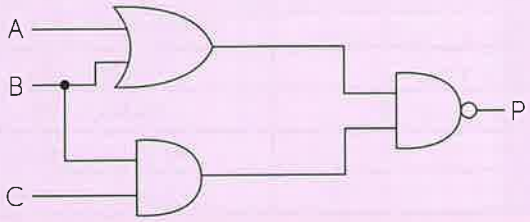
A	B	P = A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Draw a logic circuit for this logic expression: P = (A XOR B) NAND C [3]

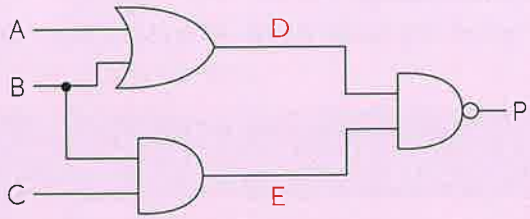
LOGIC EXPRESSIONS, LOGIC CIRCUITS AND TRUTH TABLES

Example

(a) Write a logic expression corresponding to the following logic circuit:



Solution: When there are 3 inputs, name the outputs from each gate:



Work from right to left: P = D NAND E where D = (A OR B), E = (B AND C)

Substitute for D and E: P = (A OR B) NAND (B AND C)

(b) Draw a truth table for this logic circuit.

Solution: When there are three inputs, there will be eight rows in the truth table. Notice the order in which the 0s and 1s are written in the columns for the three inputs A, B and C.

A	B	C	A OR B	B AND C	P = (A OR B) NAND (B AND C)
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	1	0

1. (a) Draw a logic circuit corresponding to the logic expression

P = (A AND NOT B) XOR C [3]

(b) Draw the truth table representing this logic circuit. [3]

2. A buzzer sounds (B = 1) when certain fault conditions are detected in a fish tank.

The inputs are:

Input	Binary value	
T	1	Temperature >= 22° Celsius
	0	Temperature < 22° Celsius
H	1	Heater is ON
	0	Heater is OFF
P	1	Air pump working
	0	Air pump not working

B = 1 if Temperature < 22° Celsius AND Heater is OFF

OR Air pump not working

Draw the logic circuit to represent the above system using AND, OR and NOT gates. [5]

EXAMINATION PRACTICE

1. Draw a truth table for the expression (A OR B) AND (A OR C). [4]

2. A mobile phone rings (R = 1) when a call is received if certain conditions are met.

INPUT	Binary value	Description
B	1	Call is from a blocked number
	0	Call is not from a blocked number
S	1	Call volume is set at > 0
	0	Call volume is set to 0
D	1	Phone is not set to "Do Not Disturb"
	0	Phone is set to "Do Not Disturb"

The phone will ring if: The call is not from a blocked number
AND Call volume is set at >0
AND Phone is not set to "Do Not Disturb"

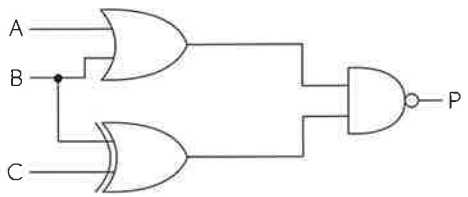
(a) Complete the truth table for this scenario. [3]

B	S	D	Working space	R
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) Draw the logic circuit for this scenario. [3]

3. (a) Draw truth tables for the logic gates
(i) A NOR B (ii) A XOR B (iii) A NAND B [3]

(b) Write the logic expression represented by this logic circuit: [3]



(c) If the inputs A, B and C are all 1, what is the value of the output P? [1]

INDEX

Symbols

2D and 3D scanners 31
2D array 94
3D printers 37
5G network 43

A

abnormal data 76
abstraction 66
access rights 48
accumulator 27
actuators 38, 61, 62
address bus 27
adware 58
agriculture 61, 62
algorithm 70
  bubble sort 73
  linear search 71
ALU 27
amplifier 37
amplitude 11
analogue sound 11
analysis 66
AND gate 104
application software 48
Arithmetic Logic Unit 26
arithmetic operators 84
array 93
artificial intelligence 63
ASCII 10
assembler 50, 51
assignment (of a variable) 82
asymmetric encryption 23
auto-correction functions 52
automated system 61
automatic repeat request 21
automatic software updates 59
average 83, 92, 94

B

barcode scanners 30, 75
binary 2
  addition 7
  representation of images 12
  shifts 8
  to denary conversion 3
  to hexadecimal 4
biometrics 59
bit 2
Bitcoin 56
bit depth 11, 12, 13
bitmap 12
blockchain 56
Boolean
  expression 84
  logic 104
  operators 85
bootloader 49
boundary data 76
breakpoint 52
browser 54
brute-force attacks 57
bubble sort 72
buses 27
byte 2

C

cache 28, 39
capacitive screens 33
CASE statements 86
CD 40
central processing unit 26
character set 10
check digit 22, 75
checksum 21
ciphertext 23
clock 27
  cycle 28
  speed 26, 28
cloud storage 43
colour depth 12, 13
command line interface 49
commenting 92
compiler 50, 51
compression 14
computer aided design 37
condition controlled loops 88
constants 82
control bus 27
control unit 26
cookies 55
cores 26, 28
COUNT statement 102
count-controlled loop 87
counting 69, 87, 102 (SQL count
CPU performance 28
current instruction register 27
cyber security threats 57

D

data 39
  bus 27
  compression 14
  interception 57
  packets 17
  theft 57
  transmission 18
database 100
debugging 52
decomposition 66, 67, 92
denary
  to binary 3
  to hexadecimal 5
device management 48
digital
  cameras 32
  certificate 59
  currency 56
  light proc. (DLP) projectors 35
  to analogue conv. (DAC) 37
distributed denial of service
    (DDOS) attack 58
DIV 84
Domain Name Server (DNS) 55
dual-core processor 28
duplex transmission 19
dynamic IP addresses 45

E

echo check 21
editors 52
embedded system 29
encryption 23
EOR gate 106
erroneous data 76
errors 78
  detection 20, 52
  diagnostics 52
  overflow 7
expert system 63
extreme data 76

F

fetch-execute cycle 26, 28
field 100
file
  handling 95
  management 48
  size 13
firmware 49
flowcharts 68, 77
format check 74
FOR ... NEXT loop 87
frequency 11
function 90

G

gibibyte 2
global variable 91
graphical user interface 49

H

hacking 57
half-duplex transmission 19
hard disk drive (HDD) 40, 41
hashing algorithm 56
header 17
Hertz 11, 28
hexadecimal 4, 6
  to binary 4, 6
  to denary 5
high-level languages 50
HTTP / HTTPS 54
HTML 55

I

identifiers 82
  meaningful 92
IDEs 52
IF statements 86
IGCSE standard pseudocode 69
images 12
indefinite loop 88
indexing
  arrays 93
  strings 89
industry 61
inference engine 63
infra-red screens 33
inkjet printers 36
input 70
  statement 83
instructions 39
instruction set 29
integrated circuit 27
interface 49, 63
Internet 54
Internet Protocol (IP) 45
interpreter 50, 51
interrupt 49
interval 11
IP addresses 45, 55
iteration 87

J

John von Neumann 26

K

keyboards 32
keyloggers 57
kibibyte 2
knowledge base 63

L

land 41
laser printers 36
LCD projectors 35
LCD screens 34
least significant bit 3
LED screens 34
ledger 56
length check 74, 102
library routines 91
linear search 71
local storage 43
local variable 91
logical operators 84
logic
  circuits 107
  diagram 105
  error 78
  expressions 107
  gates 104
low-level languages 50

M

MAC address 6, 44
machine code 29, 50, 51
machine learning 63
main memory 39
maintainable program 92
malicious code 58
malware 57, 58
MAR 27
max 83
MDR 27
mebibyte 2
medicine 62
memory
  management 49
  virtual 42
mice 32
microphones 32
microprocessor 27, 61, 62
min 70, 71
MOD 84
Modulus 11 system 22
most significant bit 3
mouse 32
multi-tasking 48

N

NAND gate 106
nested IF statements 86
nested iteration 87
network hardware 44
network interface card 44
non-volatile 39
NOR gate 106
normal data 76
NOT gate 104

O

operating system (OS) 29, 48
optical drive 41
OR gate 104
output 70
  statement 83
overflow 7, 8

P

packet switching 17
pages 42
parallel transmission 18
parameters 90
parity checks 20
passwords 59
payload 17
PC 27
peripheral management 48

persistent cookies 55
pharming 57
phishing 57
pit 41
pixel 12, 32
pixels per inch (PPI) 13
plaintext 23
post-condition loop 88
pre-condition loop 88
presence check 74, 102
prettyprint 52
primary key 100
printers 36
privacy settings 59
private key encryption 23
procedure 90
process 70
process management 48
program dev. life cycle 66
projectors 35
  DLP 35
  LCD 35
protocol 54
proxy server 59
pseudocode 69
public key encryption 23

Q

QR code scanners 31
quad-core processor 28

R

RAM 39
random number generation 91
range check 74, 102
ransomware 58
read only memory (ROM) 39
records 100
registers 27
REPEAT ... UNTIL loop 88
resistive screens 33
resolution 13
robots 62
ROM 29, 39
ROUND function 91
routers 45
rule base 63
run length encoding (RLE) 14
runtime error 52, 78

S

sample rate 11
sample resolution 11
scanners 30
science 61
scope 91
screens 34
searching 71
secondary storage 40
secure socket layer (SSL) 59
selection 85
sensors 38, 61, 62
sequence 85
serial transmission 18
session cookies 55
shifts 8
shouldering 57
sign bit 9
signed integers 9
simplex transmission 19
skewing 18
social engineering 57
software updates 59
solid-state (flash) memory 41
sorting 72
sound 11
speakers 37
spyware 57, 58
SQL 101
static IP addresses 45
stepping 52
stored program computer 26
strings 89
structure diagram 67
SUM statement 102
switches 2
symmetric encryption 23
syntax 50
  completion 52
  error 78
systems
  architecture 26
  security 49
  software 48

T

tables (database) 100
tebibyte 2
test data 76
testing 66, 76
text files 95
time slice 48
totalling 93, 102 (SQL sum)
touch screens 33
trace table 77
trailer 17
translators 51
transmission error detection 20
transmission of data 18
transport 61
Trojan 58
truth tables 104, 107
two-dimensional array 94
two's complement 9
two-step verification 59
type check 74
typical data 76

U

Unicode 10
Uniform Resource Locator 54
USB flash drive 40
USB (Universal Serial Bus) 18
user account management 48

V

validation 74, 102
variable 82, 91
  scope 91
verification 75
virtual memory 42
virus 58
volatile 39
Von Neumann architecture 26

W

watch 52
web browser 54
web server 55
WHILE ... ENDWHILE loop 88
World Wide Web 54
worm 58

X

XOR gate 106

EXAMINATION TIPS

With your examination practice, apply a boundary approximation using the following tables. These tables are calculated using an average of past years' boundaries for the 0478 (A*-G) and 0984 (9-1) IGCSE course.

Both courses are identical in content but vary in the way they are graded. Be aware that boundaries vary annually.

0478 (A*-G)	A*	A	B	C	D	E	F	G
Boundary	80%	67%	54%	41%	34%	28%	21%	15%

0984 (9-1)	9	8	7	6	5	4	3	2	1
Boundary	81%	73%	65%	56%	48%	40%	30%	23%	15%

1. Read each question carefully. Some students give answers to questions they think or hope are being asked rather than the actual question. Avoid simply rewriting a question in your answers or repeating examples that are already given in the question.
2. Be sure to write your answers in the spaces provided. Answers given outside of this space may be missed in the scanning and marking process.
3. Understand the requirements of command words at the back of the specification. If *'Describe'* or *'Explain'* questions are given you need to expand your answers. To help you justify your responses, aim to include connective words such as BECAUSE... or SO... in every answer **because** this forces you to justify your point, **so** you get additional marks. See how well it works! *'Explain'* questions such as *'Explain why this is the most appropriate...'* do not require just a list of benefits. Instead you should identify the benefits and then expand each one, applying them to the scenario or context.
4. No marks are awarded for using brand names of software packages or hardware, e.g. "MS Word", "Excel", "iPhone", "Android" or "Windows". Use generic terms where you can, e.g. word processor or smartphone.
5. Full answers should be given to questions – not just key words. Make your answers match the context of the question.
6. Generic answers are not sufficient. E.g. If a question asks for a description of the function of a router, an answer 'it connects devices together' is not sufficient. Instead, answers should describe how routers are used to receive packets from computers, read the destination address of each and then forward each packet to its destination. *Faster, bigger* and *cheaper* are not very useful responses unless you justify your points.
7. Algorithm questions require an actual algorithm, not a repetition of the question. If a question explicitly asks for an algorithm to be written in pseudocode, then it will not gain marks if it is written as a flowchart. Equally, a question that asks for an algorithm to be written as a flowchart will not gain marks if answered with pseudocode.
8. Learn and make use of the standard flowchart symbols when drawing or constructing charts.
9. Be careful when outputting strings and variables: the string must be in quotes, e.g. `OUTPUT "Hello", name`
10. Be careful with quotes around strings. E.g. `choice ← A` (which assigns a variable `A` to `choice`) is very different from `choice ← "A"` (which assigns a value `"A"` to `choice`).
11. The first element of an array has an index of 1 in IGCSE pseudocode, e.g. Day[1]. In most programming languages, e.g. Python, the index of the first array element is 0.
12. A common error in IF statements is writing `IF name = "Sam" OR "sam"`. This should be:
    `if name = "Sam" OR name = "sam"`

Good luck!

Revision, re-imagined

the ClearRevise family expands

New titles coming soon!

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

They have been expertly compiled and edited by subject specialists, highly experienced examiners, industry professionals and a good dollop of scientific research into what makes revision most effective. Past examination questions are essential to good preparation, improving understanding and confidence.

- Hundreds of marks worth of examination style questions
- Answers provided for all questions within the books
- Illustrated topics to improve memory and recall
- Specification references for every topic
- Examination tips and techniques
- Free Python solutions pack (CS Only)

Absolute clarity is the aim.

Explore the series and add to your collection at **www.clearrevise.com**

Available from all good book shops.

amazon @pgonlinepub

Cambridge IGCSE **Computer Science** 0478 / 0984

ClearRevise™

Illustrated revision and practice:

- Over 400 marks of examination style questions
- Answers provided for all questions within the book
- Illustrated topics to improve memory and recall
- Specification references for each topic
- Examination tips and techniques

Experience + science + beautiful design = better results

Absolute clarity is the aim with a new generation of revision guide. This guide has been expertly compiled and edited by successful teachers of Design and Technology, industry professionals, highly experienced examiners and a good measure of scientific research into what makes revision most effective.

PG Online have a record of significantly raising and sustaining examination results at GCSE in schools using their award-winning teaching resources.

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

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

www.clearrevise.com

ISBN 978-1-910523-38-4

9 781910 523384

PG ONLINE