



Please write clearly in block capitals.

Centre number

--	--	--	--	--

Candidate number

--	--	--	--

Surname

Forename(s)

Candidate signature

I declare this is my own work.

A-level COMPUTER SCIENCE

Paper 2

Wednesday 18 June 2025 Morning Time allowed: 2 hours 30 minutes

Materials

For this paper you must have:

- a calculator.

Instructions

- Use black ink or black ball-point pen.
- Fill in the boxes at the top of this page.
- Answer **all** questions.
- You must answer the questions in the spaces provided. Do not write outside the box around each page or on blank pages.
- If you need extra space for your answer(s), use the lined pages at the end of this book. Write the question number against your answer(s).
- Do all rough work in this book. Cross through any work you do not want to be marked.

Information

- The marks for questions are shown in brackets.
- The maximum mark for this paper is 100.

Advice

- In some questions you are required to indicate your answer by completely shading a lozenge alongside the appropriate answer as shown. 
- If you want to change your answer you must cross out your original answer as shown. 
- If you wish to return to an answer previously crossed out, ring the answer you now wish to select as shown. 

For Examiner's Use	
Question	Mark
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
TOTAL	



J U N 2 5 7 5 1 7 2 0 1

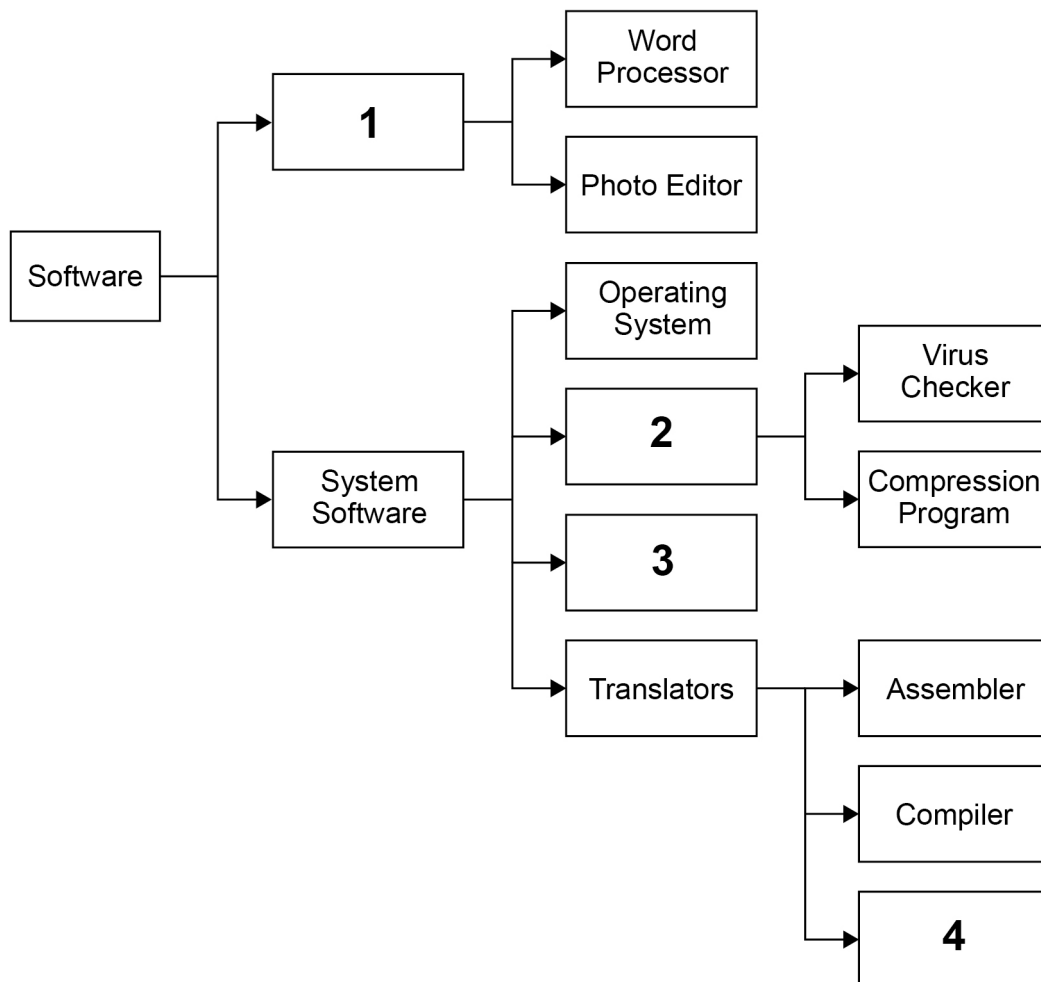
Answer **all** questions.

Do not write
outside the
box

0 1 . 1

Figure 1 shows the classification of various types of software and some examples of these types.

Figure 1



Complete **Table 1** by naming the correct types of software that should be placed at positions **1** to **4** in **Figure 1**.

[2 marks]

Table 1

Number	Correct label
1	
2	
3	
4	

0 1 . 2

One function of the operating system is to allocate I/O devices to competing processes.

Describe **two other** functions of an operating system.

[2 marks]

Function 1 _____

Function 2 _____

Question 1 continues on the next page

Turn over ►



[4 marks]

[illegible]

8



0 2 . 1

A sound file, produced by sampling sound, requires 1.5 megabytes of storage space.

The sound file has a duration of 50 seconds and was recorded at a sample rate of 20 000 samples per second (Hertz).

Calculate the sample resolution that was used when the sound was sampled.

[2 marks]

Answer _____ bits

0 2 . 2

Describe the principles of operation of an analogue to digital converter (ADC).

[2 marks]

Turn over ►



0 2 . 3

Explain **one** reason why a digital sound file that has been recorded using sampling may not accurately represent the original analogue signal produced by the microphone.

[2 marks]

0 2 . 4

One advantage of representing a piece of music using MIDI instead of as sampled sound is that the file produced requires less storage space.

Explain why.

[2 marks]



In your response, you should cover how data is represented on a disk and how the data would be read.

[illegible]

0	3	.	2
---	---	---	---

State **two** reasons why most modern computers do not have an optical disk drive.**[2 marks]**

Reason 1 _____

Reason 2 _____

8



0	4
---	---

It is difficult to write and enforce laws relating to the use of computers.

Explain why.

[3 marks]

3

Turn over for the next question

Turn over ►



Explain why a client-server network would be more suitable for use in a school than a peer-to-peer network.

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.



Turn over for the next question

*Do not write
outside the
box*

**DO NOT WRITE ON THIS PAGE
ANSWER IN THE SPACES PROVIDED**

Turn over ►



0 6

A school uses a relational database to store information about merits that are given to students. A merit is an award made for outstanding work or conduct.

At the end of each term, prizes are awarded to the students who achieve the most merits.

The database contains the four relations shown in **Figure 2**.

Figure 2

Student(StudentID, FirstName, LastName, YearGroup, House)

Teacher(TeacherID, FirstName, LastName)

Merit(MeritID, StudentID, TeacherID, DateAwarded, Reason)

Term(TermName, TermYear, StartDate, EndDate)

The Student relation stores information about the students who attend the school. An example of the data stored about a student is:

StudentID	FirstName	LastName	YearGroup	House
20965	Sophie	Latham	12	Kitchener

The Teacher relation stores information about the teachers who work at the school and who award merits to students.

The Merit relation stores information about the merits that teachers have awarded to students, when they were awarded and for what reason.

The Term relation stores information about the dates of each term. An example of the data stored about a term is:

TermName	TermYear	StartDate	EndDate
Autumn	2024	04/09/2024	20/12/2024

The StudentID, TeacherID, MeritID, YearGroup and TermYear attributes are all numeric values.



06.1

When the Merit relation was first designed, it did not contain the MeritID field. Instead, the values in the StudentID, TeacherID and DateAwarded attributes were used together as a composite entity identifier.

Describe the limitation that would have occurred if the entity identifier for the Merit relation consisted of the StudentID, TeacherID and DateAwarded attributes.

[1 mark]

06.2

The diagram below shows three of the entities from the school merit database.

Draw the relationships that exist between the three entities shown on the diagram.

You **must** clearly show the degree of each relationship.

[1 mark]

Student

Teacher

Merit

Turn over ►



Figure 2 is repeated below to help you answer Questions **06.3** and **06.4** without having to turn back in the question paper.

Figure 2 (repeated)

Student(StudentID, FirstName, LastName, YearGroup, House)

Teacher(TeacherID, FirstName, LastName)

Merit(MeritID, StudentID, TeacherID, DateAwarded, Reason)

Term(TermName, TermYear, StartDate, EndDate)

06.3

A new student has joined the school. The student is called Ethan Smith and their year group is 7. They are placed in Hulme house and given the StudentID number 17423.

Write SQL code to add the student's details to the Student table.

[2 marks]



IB/G/Jun25/7517/2

0	6	.	5
---	---	---	---

The merits database can be accessed on any computer within the school.

It is possible that two users might try to edit the same record simultaneously.

Describe how two users trying to edit the same record could cause a problem if the database system did not implement a method to manage concurrent access.

[2 marks]

12



07.1

State the stored program concept.

[2 marks]

07.2

Describe the roles of the following **three** components of a processor.**[6 marks]**

Control unit _____

General purpose registers _____

Status register _____

Turn over ►

A particular processor represents machine code instructions in 32 bits.

The number of bits allocated to the opcode varies depending upon the operation that the instruction represents. The remaining bits are allocated to the operands. An instruction can have between zero and three operands.

Figure 3 shows the format of an `ADD` instruction in assembly language and machine code.

Figure 3

ADD Instruction: `ADD R2, R5, #17`

Opcode												Operands																				
Basic Machine Operation											M	Operand A				Operand B				Operand C												
1	1	1	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1		

- **Operand A** and **Operand B** are register numbers.
- The bit labelled **M** in **Opcode** indicates the addressing mode to use for **Operand C**.
- **Operand C** can refer to either a register number or a numeric value, depending upon the addressing mode that is being used.

07.3

All of the processor's general purpose registers can be referenced using each of the operands in the `ADD` instruction in **Figure 3**.

Calculate the maximum number of general purpose registers that the processor could have.

[1 mark]

Answer _____



07.4

When the immediate addressing mode is used:

- the value in **Operand C** in **Figure 3** represents a numeric value that will be processed by the `ADD` instruction
- the value in **Operand C** is represented as a **two's complement binary integer**.

In **decimal**, calculate the range of numbers that **Operand C** can represent.

[1 mark]

Most positive _____

Most negative _____

10

Turn over for the next question

Turn over ►



The Internet Protocol version 4 (IPv4) addressing system does not provide enough unique addresses for every device that is connected to the Internet.

In the short term, the use of public and private IP addresses, NAT and DHCP is helping to deal with this problem.

- why IPv4 does not provide enough unique addresses and how IPv6 will overcome this problem
- the difference between a public (routable) and a private (non-routable) IP address
- how NAT works and how it helps to deal with this problem
- how DHCP works and how it helps to deal with this problem.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



Do not write
outside the
box

Handwriting practice area with 20 horizontal lines.

Turn over ►



0 9

Figure 4 shows a logic circuit. The circuit has seven inputs and two outputs. The inputs B_6 to B_0 represent the ASCII code of a character, where B_6 is the most significant bit and B_0 is the least significant bit. The circuit outputs are Q_1 and Q_2 . The outputs of six of the gates in the circuit have been labelled U , V , W , X , Y and Z .

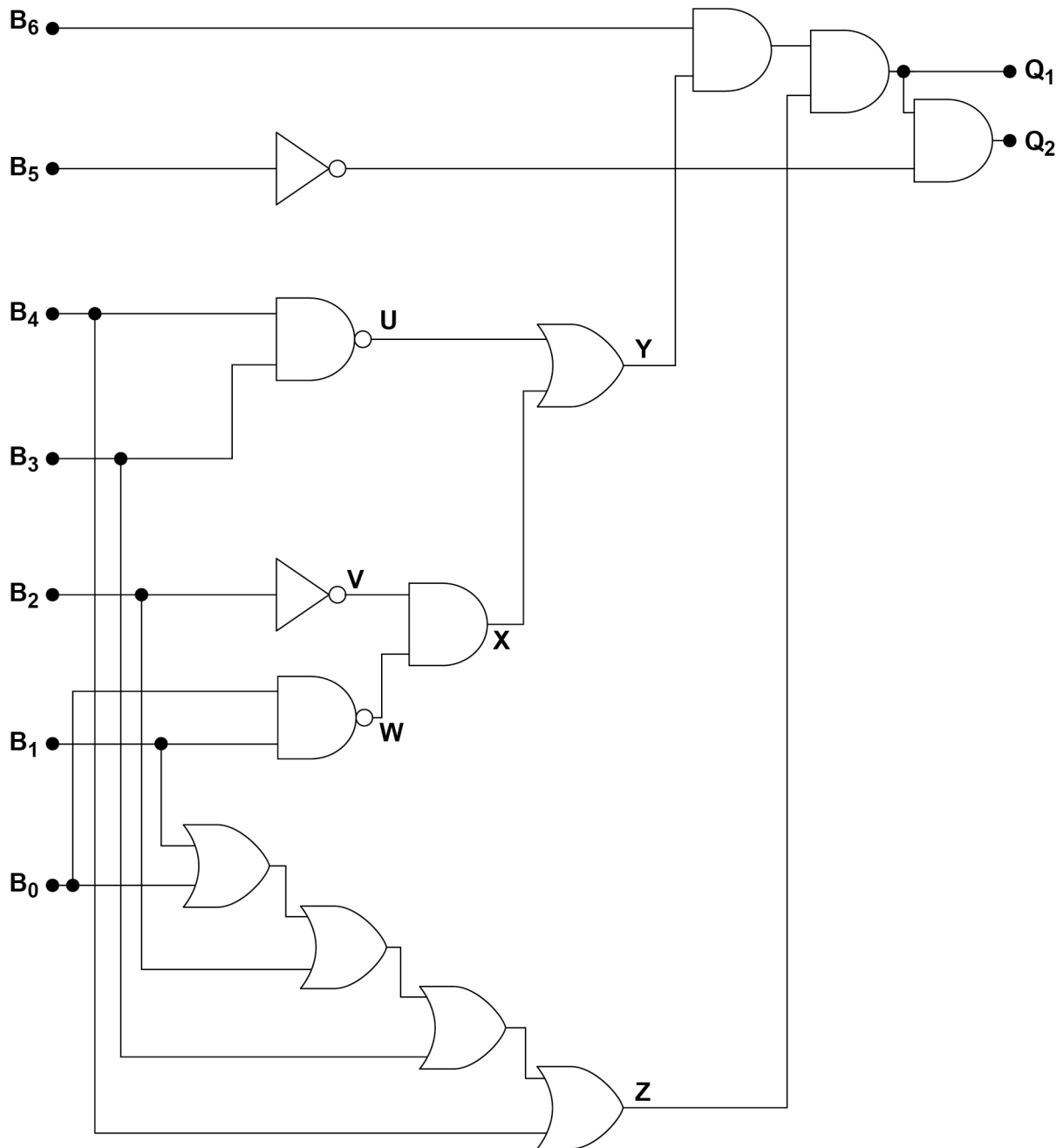
Figure 4

Figure 5 shows the ASCII code for printable characters.

Figure 5

ASCII code		ASCII code		ASCII code		ASCII code	
Char	Binary	Char	Binary	Char	Binary	Char	Binary
SP	0100000	8	0111000	P	1010000	h	1101000
!	0100001	9	0111001	Q	1010001	i	1101001
"	0100010	:	0111010	R	1010010	j	1101010
#	0100011	;	0111011	S	1010011	k	1101011
\$	0100100	<	0111100	T	1010100	l	1101100
%	0100101	=	0111101	U	1010101	m	1101101
&	0100110	>	0111110	V	1010110	n	1101110
'	0100111	?	0111111	W	1010111	o	1101111
(	0101000	@	1000000	X	1011000	p	1110000
)	0101001	A	1000001	Y	1011001	q	1110001
*	0101010	B	1000010	Z	1011010	r	1110010
+	0101011	C	1000011	[	1011011	s	1110011
,	0101100	D	1000100	\	1011100	t	1110100
-	0101101	E	1000101	]	1011101	u	1110101
.	0101110	F	1000110	^	1011110	v	1110110
/	0101111	G	1000111	_	1011111	w	1110111
0	0110000	H	1001000	`	1100000	x	1111000
1	0110001	I	1001001	a	1100001	y	1111001
2	0110010	J	1001010	b	1100010	z	1111010
3	0110011	K	1001011	c	1100011	{	1111011
4	0110100	L	1001100	d	1100100		1111100
5	0110101	M	1001101	e	1100101	}	1111101
6	0110110	N	1001110	f	1100110	~	1111110
7	0110111	O	1001111	g	1100111		

0 9 . 1

Complete the truth table below for the input of the ASCII codes of the characters * @ F W \ and f into the logic circuit shown in Figure 4.

[4 marks]

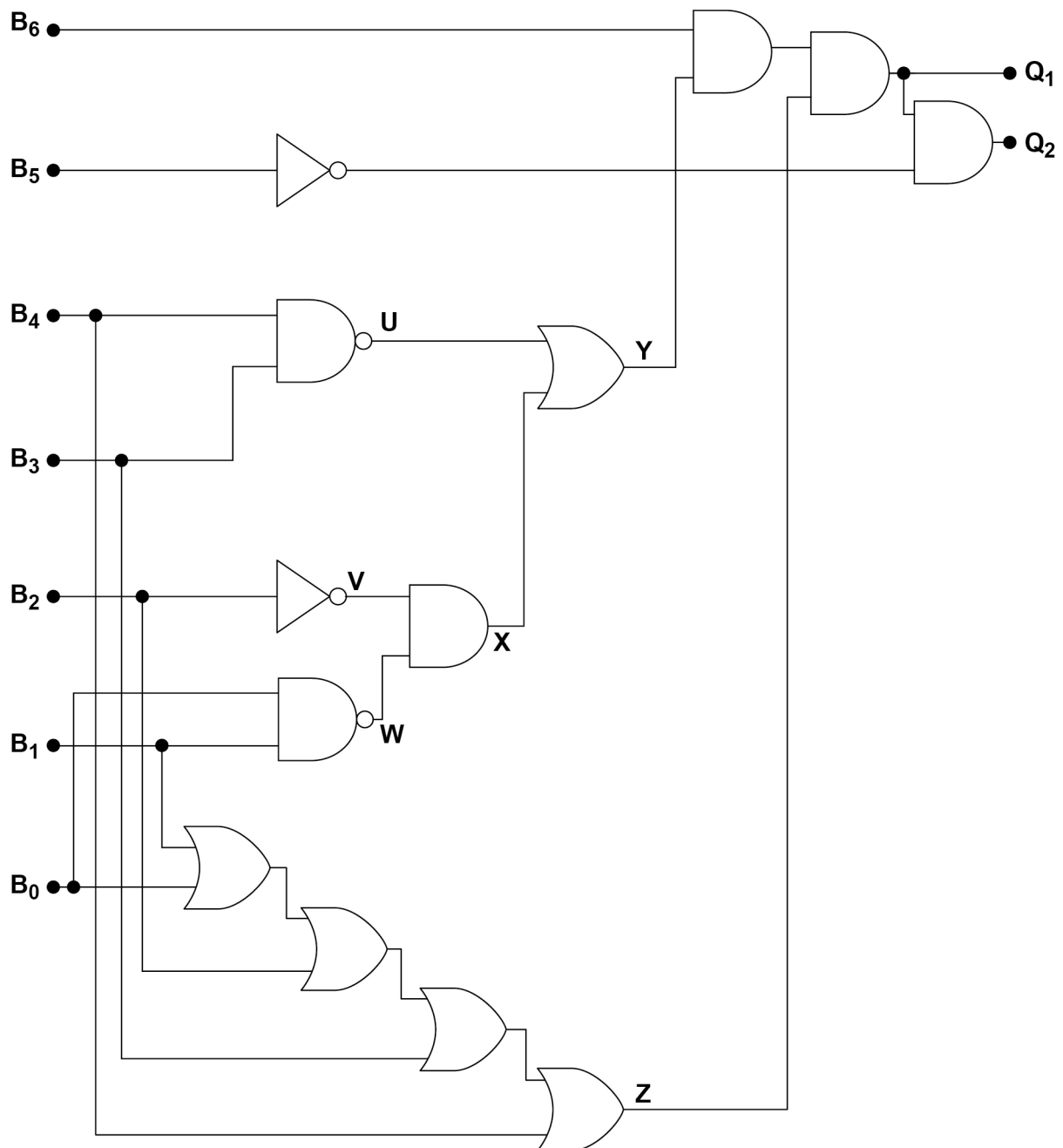
		Inputs												Outputs		
Char	Binary	B ₆	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	U	V	W	X	Y	Z	Q ₁	Q ₂
*	0101010	0	1	0	1	0	1	0								
@	1000000	1	0	0	0	0	0	0								
F	1000110	1	0	0	0	1	1	0								
W	1010111	1	0	1	0	1	1	1								
\	1011100	1	0	1	1	1	0	0								
f	1100110	1	1	0	0	1	1	0								

Turn over ►



Figure 4 is repeated below to help you answer Questions **09.2** and **09.3** without having to turn back in the question paper.

Figure 4 (repeated)



0 9 . 2

Describe the purpose of the outputs Q_1 and Q_2 of the circuit in **Figure 4**.

[2 marks]

 Q_1 _____

 Q_2 _____

0 9 . 3

Shade **one** lozenge to indicate which of the Boolean equations correctly represents the output Q_1 in **Figure 4**.

[1 mark]

A $Q_1 = B_6 \cdot (\overline{B_4 \cdot B_3} + \overline{B_2 \cdot B_1 \cdot B_0}) \cdot (B_4 + B_3 + B_2 + B_1 + B_0) \cdot \overline{B_5}$ ☐

B $Q_1 = B_6 \cdot (\overline{B_4 \cdot B_3} + \overline{B_2 \cdot B_1 \cdot B_0}) \cdot (B_4 + B_3 + B_2 + B_1 + B_0)$ ☐

C $Q_1 = B_6 \cdot (\overline{B_4 \cdot B_3} + \overline{B_2 \cdot B_1 \cdot B_0}) \cdot (B_4 + B_3 + B_2 + B_1 + B_0)$ ☐

D $Q_1 = B_6 \cdot \overline{B_4 \cdot B_3} + \overline{B_2 \cdot B_1 \cdot B_0} \cdot (B_4 + B_3 + B_2 + B_1 + B_0)$ ☐

E $Q_1 = B_6 + (\overline{B_4 + B_3 \cdot B_2 + B_1 + B_0}) + (B_4 \cdot B_3 \cdot B_2 \cdot B_1 \cdot B_0)$ ☐

7

Turn over for the next question

Turn over ►



1 0 . 1

State **two** reasons why data is often compressed.**[2 marks]**

Reason 1 _____

Reason 2 _____

1 0 . 2

Two different methods of compressing data are run length encoding (RLE) and dictionary-based methods.

Explain why RLE is more suitable for compressing images and dictionary-based methods are more suitable for compressing text.

[2 marks]

4

Turn over for the next question

*Do not write
outside the
box*

**DO NOT WRITE ON THIS PAGE
ANSWER IN THE SPACES PROVIDED**

Turn over ►



1 1

The Fibonacci sequence is a number sequence in which the first two numbers are both 1 and every other number in the sequence is the sum of the previous two numbers. The first ten values in the Fibonacci sequence are:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55

Figure 6 shows a function written in a functional programming language to calculate the n th term in the Fibonacci sequence.

Figure 6

```
fibonacci 1 = 1
fibonacci 2 = 1
fibonacci n = fibonacci (n - 1) + fibonacci (n - 2)
```

1 1 . 1

Shade **all** of the lozenges for the statements that are **true** about the `fibonacci` function in **Figure 6**.

[1 mark]

- | | | |
|----------|--|--------------------------|
| A | The function has one argument. | ☐ |
| B | The function is a first-class object. | ☐ |
| C | The function outputs a list. | ☐ |
| D | The function uses a higher-order function. | ☐ |

1 1 . 2

Shade **one** lozenge to indicate the function type of the `fibonacci` function in **Figure 6**.

[1 mark]

- | | | |
|----------|-------------------------------------|--------------------------|
| A | $\mathbb{N} \rightarrow \mathbb{N}$ | ☐ |
| B | $\mathbb{Q} \rightarrow \mathbb{Z}$ | ☐ |
| C | $\mathbb{R} \rightarrow \mathbb{Z}$ | ☐ |
| D | $\mathbb{Z} \rightarrow \mathbb{N}$ | ☐ |

1 1 . 3

Explain why the recursive method used by the code in **Figure 6** is **not** an efficient method of calculating the n th term in the Fibonacci sequence.

[2 marks]

1 1 . 4

Describe what partial function application is.

[2 marks]

6

Turn over for the next question

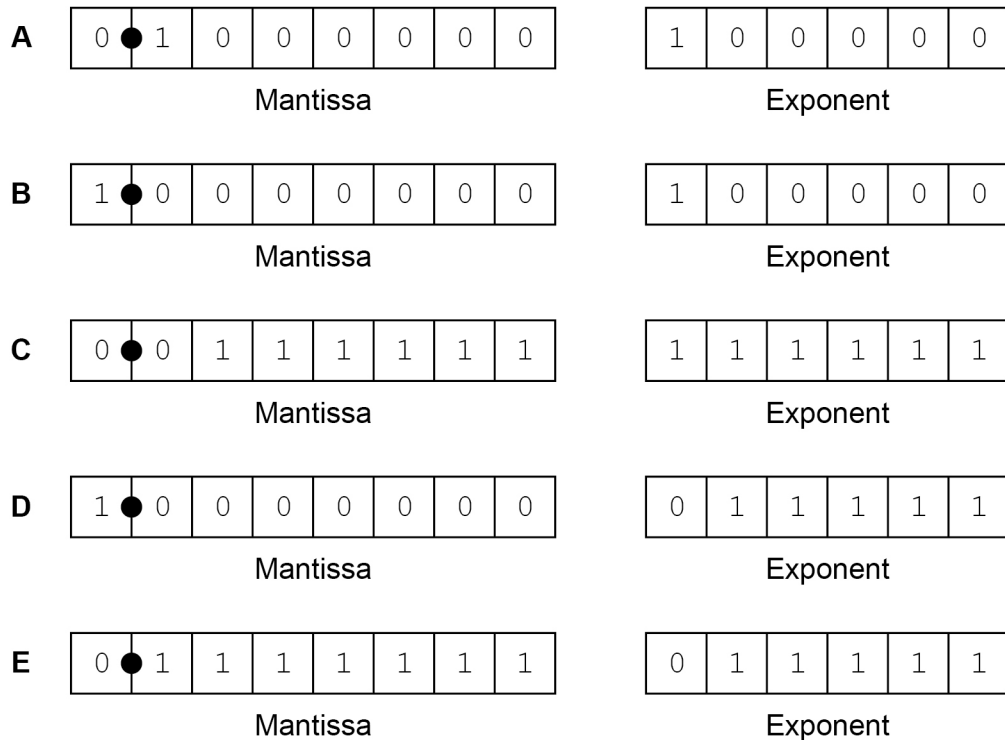
Turn over ►



1 2

A computer represents numbers using a **normalised** floating point representation with an **8-bit mantissa** and a **6-bit exponent**, both stored using **two's complement**, as shown in **Figure 7**.

Five different bit patterns that are stored in this computer's memory are listed in **Figure 7** and are labelled with the letters **A** to **E**. Four of the bit patterns are valid normalised floating point numbers and one is not.

Figure 7

1 2 . 1

Shade **one** lozenge to indicate which bit pattern (**A–E**) in **Figure 7** does not represent a normalised value.

[1 mark]

A ☐ **B** ☐ **C** ☐ **D** ☐ **E** ☐

1 2 . 2

Shade **one** lozenge to indicate which bit pattern (**A–E**) in **Figure 7** shows the most negative normalised value that can be represented.

[1 mark]

A ☐ **B** ☐ **C** ☐ **D** ☐ **E** ☐



1 2 . 3

State **one** reason why values stored using a floating point representation are usually normalised.

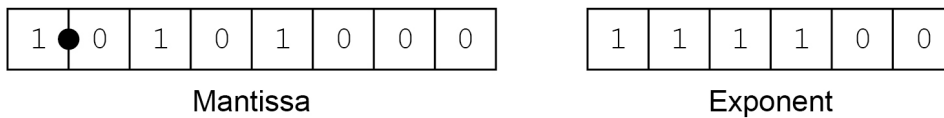
[1 mark]

Question parts **12.4** and **12.5** use a **normalised** floating point representation with an **8-bit mantissa** and a **6-bit exponent**, both stored using **two's complement**.

1 2 . 4

Figure 8 shows a floating point representation of a number:

Figure 8



Calculate the decimal equivalent of the number in **Figure 8**.

Express your answer to at least four decimal places or as a fraction.

You should show your working.

[2 marks]

Answer _____

Question 12 continues on the next page

Turn over ►



Question parts **12.4** and **12.5** use a **normalised** floating point representation with an **8-bit mantissa** and a **6-bit exponent**, both stored using **two's complement**.

1 2 . 5

The decimal value 43 057 152 cannot be represented exactly using this floating point system.

Write the closest possible normalised floating point representation of the decimal value 43 057 152 in the boxes below.

You should show your working.

[3 marks]

Answer

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

Mantissa

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

Exponent

1 2 . 6

Calculate the **relative error** that occurred when you represented 43 057 152 as closely as possible in question part **12.5**.

You should show your working.

Express your answer as a percentage to at least four decimal places.

[1 mark]

Answer _____

9



$$\overline{A \cdot B + \bar{A} \cdot C + B \cdot C + \bar{B} \cdot A}$$

[4 marks]

[illegible]

4

1 4

An assembly language program is used to control the operation of a robot.

The program uses the AQA assembly language instruction set in **Table 3** on page 36.

The value in memory location 100 is used to read input from the robot's sensors and to provide output to the robot's motors. The memory location can hold an 8-bit value.

Figure 9 shows an example 8-bit value in memory location 100.

Figure 9

Outputs			Inputs				
B₇	B₆	B₅	B₄	B₃	B₂	B₁	B₀
1	1	0	1	0	0	1	0

The purpose of each bit is:

- **B₇** – Controls the motor that drives the robot's left wheel.
- **B₆** – Controls the motor that drives the robot's right wheel.
- **B₅** – Controls the motor that drives the robot's accessory arm.
- **B₄** – Represents the input from an ultrasonic sensor.
- **B₃** – Represents the input from a light sensor.
- **B₂** – Represents the input from touch sensor **X**.
- **B₁** – Represents the input from touch sensor **Y**.
- **B₀** – Represents the input from touch sensor **Z**.

Writing a value of 1 to one of the bits **B₇** to **B₅** will cause the motor controlled by the bit to turn on. The motor will remain on until a 0 is written to the bit.

Reading a value of 1 from bits **B₄** to **B₀** indicates that the sensor associated with the bit has been activated. Reading a value of 0 indicates that the sensor has not been activated.

For example, the value 11010010 in memory location 100 in **Figure 9** indicates that the robot is driving forward, as both the left and right motors are on, and that the ultrasonic sensor and touch sensor **Y** have been activated.

The current state of the inputs can be determined by reading the value from memory location 100. When a value is written to memory location 100, only the values in bits **B₇** to **B₅** change.

The program in **Table 2** is used to make the robot carry out a task. The program has been split into six sections.

Complete the **Purpose** column of **Table 2** to explain the effect of the code in sections **2** to **6** on the robot. In your response, where appropriate, you should include the effect of the sensor inputs on the robot's actions.

[6 marks]



Table 2

Section	Code	Purpose
1	LDR R0, 100 ORR R0, R0, #192 STR R0, 100	Make the robot drive forwards.
2	loop1: LDR R0, 100 AND R1, R0, #7 CMP R1, #0 BEQ loop1	Complete rows 2 to 6 in the Purpose column
3	LDR R0, 100 AND R0, R0, #63 STR R0, 100	
4	LDR R0, 100 MOV R2, #0 MOV R3, #0 loop2: AND R1, R0, #1 CMP R1, #0 BEQ skip ADD R2, R2, #1 skip: ADD R3, R3, #1 LSR R0, R0, #1 CMP R3, #3 BNE loop2 CMP R2, #2 BEQ active HALT	
5	active: LDR R0, 100 ORR R0, R0, #32 STR R0, 100	
6	loop3: LDR R0, 100 AND R1, R0, #8 CMP R1, #0 BEQ loop3 AND R0, R0, #223 STR R0, 100 HALT	



Table 3 – Standard AQA assembly language instruction set

LDR Rd, <memory ref>	Load the value stored in the memory location specified by <memory ref> into register d
STR Rd, <memory ref>	Store the value that is in register d into the memory location specified by <memory ref>
ADD Rd, Rn, <operand2>	Add the value specified in <operand2> to the value in register n and store the result in register d
SUB Rd, Rn, <operand2>	Subtract the value specified by <operand2> from the value in register n and store the result in register d
MOV Rd, <operand2>	Copy the value specified by <operand2> into register d
CMP Rn, <operand2>	Compare the value stored in register n with the value specified by <operand2>
B <label>	Always branch to the instruction at position <label> in the program.
B<condition> <label>	Branch to the instruction at position <label> if the last comparison met the criterion specified by <condition>. Possible values for <condition> and their meanings are: EQ: equal to NE: not equal to GT: greater than LT: less than
AND Rd, Rn, <operand2>	Perform a bitwise logical AND operation between the value in register n and the value specified by <operand2> and store the result in register d
ORR Rd, Rn, <operand2>	Perform a bitwise logical OR operation between the value in register n and the value specified by <operand2> and store the result in register d
EOR Rd, Rn, <operand2>	Perform a bitwise logical XOR (exclusive or) operation between the value in register n and the value specified by <operand2> and store the result in register d
MVN Rd, <operand2>	Perform a bitwise logical NOT operation on the value specified by <operand2> and store the result in register d
LSL Rd, Rn, <operand2>	Logically shift left the value stored in register n by the number of bits specified by <operand2> and store the result in register d
LSR Rd, Rn, <operand2>	Logically shift right the value stored in register n by the number of bits specified by <operand2> and store the result in register d
HALT	Stops the execution of the program.

Labels: A label is placed in the code by writing an identifier followed by a colon (:). To refer to a label, the identifier of the label is placed after the branch instruction.

Interpretation of <operand2>

`<operand2>` can be interpreted in two different ways, depending on whether the first character is a # or an R:

- # – use the decimal value specified after the #, eg #25 means use the decimal value 25
- Rm – use the value stored in register m, eg R6 means use the value stored in register 6

END OF QUESTIONS



There are no questions printed on this page

*Do not write
outside the
box*

**DO NOT WRITE ON THIS PAGE
ANSWER IN THE SPACES PROVIDED**



[illegible]

*Do not write
outside the
box*

[illegible]

