CAMBRIDGE
International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/32**

Paper 3 Advanced Theory **October/November 2025**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the October/November 2025 series for most Cambridge IGCSE, Cambridge International A and AS Level components, and some Cambridge O Level components.

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptions for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5: Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).
GENERIC MARKING PRINCIPLE 6: Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Annotations guidance for centres

Examiners use a system of annotations as a shorthand for communicating their marking decisions to one another. Examiners are trained during the standardisation process on how and when to use annotations. The purpose of annotations is to inform the standardisation and monitoring processes and guide the supervising examiners when they are checking the work of examiners within their team. The meaning of annotations and how they are used is specific to each component and is understood by all examiners who mark the component.

We publish annotations in our mark schemes to help centres understand the annotations they may see on copies of scripts. Note that there may not be a direct correlation between the number of annotations on a script and the mark awarded. Similarly, the use of an annotation may not be an indication of the quality of the response.

The annotations listed below were available to examiners marking this component in this series.

Annotations

Annotation	Meaning
	Correct
	Incorrect
	To indicate where a key word/phrase/code is missing.
	Not relevant or used to separate parts of an answer.
	Indicates a part of the answer that is incorrect.
Highlighter	To draw attention to a particular aspect or to indicate where parts of an answer have been combined.
	Too vague.
	Repetition
	No examples or not enough.

Annotation	Meaning
BOD	Benefit of Doubt.
NAQ	Not Answered Question.
SEEN	Indicates that work or a page has been seen including blank answer spaces and blank pages.
FT	Follow through.
I	Ignore

Question	Answer	Marks
1(a)(i)	`DECLARE Car1 : Car`	**1**

1(a)(ii)	**One** mark for each correct answer
	
	Example answer
	`Car1.Colour ← "Blue"`
	`Car1.IntoStock ← 21/10/2025`
		2

1(b)(i)	**One** mark per mark point (**Max 2**) MP1 `TYPE Body =` MP2 `(Convertible, Hatchback, Saloon, SUV)` Example answer `TYPE Body = (Convertible, Hatchback, Saloon, SUV)`	**2**
1(b)(ii)	`DECLARE BodyStyle : Body`	**1**

Question	Answer	Marks
2(a)	**Two** marks for working • correct calculation/application of exponent seen • correct method to find the final answer Working: Exponent = 8 + 2 + 1 = 11 // =0.111100101 x 2^{11} // =11110010100.0 (moving bp 11 places to right) Method to find the answer // 1024 + 512 + 256 + 128 + 16 + 4 **One** mark for correct answer Denary value: 1940 Example of solution using fractions:	**3**
2(b)	**One** mark per mark point (**Max 3**) MP1 correct **method** to find the binary number MP2 correct **use** of exponent MP3 correct answer in the space provided Working: 26.6875 converted to binary (0)11010.1011 // 16+8+2+0.5+0.125+0.0625 movement of binary point by 5 places **Mantissa** 0 1 1 0 1 0 1 0 1 1 **Exponent** 0 0 0 1 0 1	**3**

Question	Answer	Marks
3(a)	**One** mark per mark point (**Max 2**) • HTTP to send and receive / transfer web pages / hypertext / images / videos • IMAP allows users to access/read their emails **from any device without removing** the message from the mail server // **synchronises** emails **on any device**	**2**
3(b)	**One** mark per mark point (**Max 4**) MP1 Files are shared over a **peer-to-peer** network MP2 A small file, a torrent (descriptor file), is initially created by a peer MP3 The torrent (descriptor file) contains metadata about the file to be shared MP4 The whole file must initially be located on at least one peer MP5 The file is broken into **equal sized** pieces MP6 Other peers who wish to download the file first obtain the torrent (descriptor file) and connect to a tracker MP7 A tracker acts like a server that holds all the data about all the computers connected to it / the swarm MP8 Each **peer** downloads parts of the file from other **peers** until the entire file has been downloaded MP9 As each peer receives a piece of the file, they become a source for that piece of the file MP10 A peer who has a **complete file** and is sharing it is a seed // Once a peer has **completely downloaded** the file and made the file available to others in the swarm, they become a seed.	**4**

Question	Answer	Marks
4(a)	**One** mark for a benefit of circuit switching (**Max 1**) MP1 Once the connection is made, it is available until the end of the transmission (suitable for long continuous transmission) MP2 The dedicated path ensures a steady rate of data transmission, once connection has been made (because the whole bandwidth is available) MP3 The dedicated path ensures data is less likely to be lost. MP4 No intermediate delays once the circuit is established, enables real time transmission MP5 No additional time/delay to reorder needed because data arrives in the order it was sent / data sent as a continuous stream **One** mark for a benefit of packet switching (**Max 1**) MP6 No need to tie up a communication line // path available to multiple users // bandwidth can be shared by multiple users MP7 Possible to overcome failed / faulty / busy lines by re-routing packets MP8 Individual packets can be resent if lost/damaged MP9 Users only charged for the duration of connectivity MP10 High rate of data transmission possible MP11 Always uses digital networks so data is transmitted directly to the destination MP12 Good security because all packets can take different routes // A more secure method because all packets can take different routes	**2**

Question	Answer	Marks
4(b)	**One** mark per difference (**Max 2**) MP1 Circuit switching requires a dedicated line to be connected before the data transfer takes place, in packet switching, data transfer commences directly/ straight away MP2 In circuit switching, each data unit knows the entire path address, but in packet switching, each data unit only knows the final address (routers decide intermediate paths) MP3 Circuit switching uses the whole bandwidth of the transmission path, but packet switching shares it with other users // circuit switching uses a constant high bandwidth. Packet switching has variable bandwidth MP4 In circuit switching each data unit follows the same route, but in packet switching, packets can follow any route MP5 In circuit switching data arrives in the correct order but in packet switching it has to be reordered MP6 Data remains intact / a continuous stream in circuit switching but is segmented in packet switching MP7 Lost data in circuit switching requires the **whole transmission to be resent**, but **individual packets can be resent** in packet switching MP8 Circuit switching doesn't suffer data loss but packets can be lost in packet switching.	**2**

Question	Answer	Marks
5(a)	**One** mark per mark point (**Max 2**) MP1 The low-level scheduler manages the handling of interrupts based on priority MP2 … ensuring that critical events are handled without delay MP3 It uses an Interrupt Vector Table (IVT) / Interrupt Descriptor/Despatch Table (IDT) / Interrupt Service Routine (ISR) lookup MP4 … **to map the interrupt to the specific handling routine** / ISR.	**2**
5(b)	**One** mark per correct answer (**Max 3**)	**3**

Process state	Reason
running	CPU time has been allocated and the process is being executed.
ready	The process is waiting (in the ready queue) for a slice of CPU time. It could run. // It has been displaced by a higher priority process and could otherwise still run.
blocked	The process is waiting for an I/O operation / some event to take place / be completed.

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(a)</td><td>

- **One** mark for working, (all four columns P, Q, R and S)
- **One** mark for first four rows of column Z
- **One** mark for second four rows of column Z

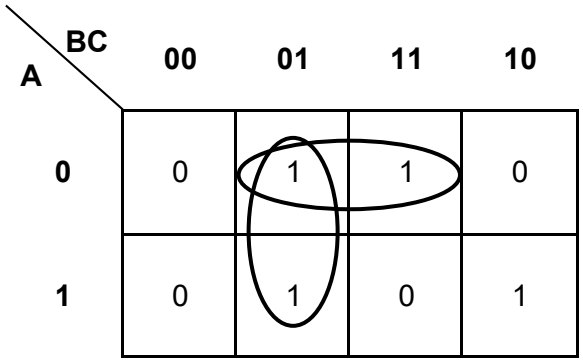
<table>
<tr><th colspan="3"></th><th colspan="5">Working space</th></tr>
<tr><th>A</th><th>B</th><th>C</th><th>P</th><th>Q</th><th>R</th><th>S</th><th>Z</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>

</td><td>3</td></tr>
<tr><td>6(b)(i)</td><td>

Two marks if no errors present
One mark if only one error present

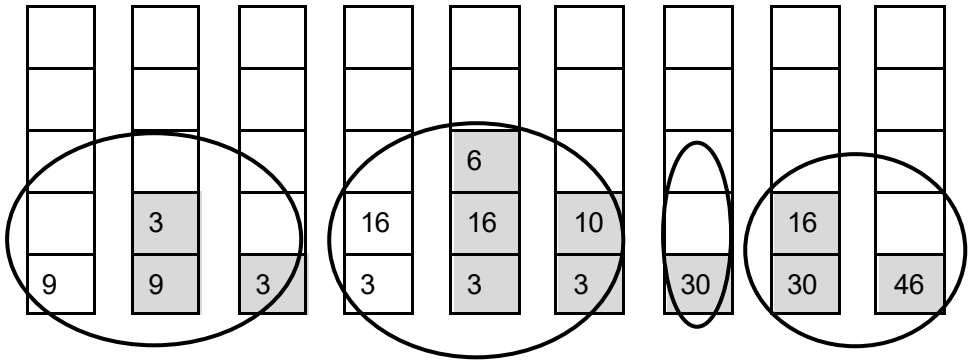
<table>
<tr><th>A \ BC</th><th>00</th><th>01</th><th>11</th><th>10</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
</table>

</td><td>2</td></tr>
</table>

Question	Answer	Marks
6(b)(ii)	**One** mark for each correct loop **(Max 2)**  	**2**
6(b)(iii)	**One** mark for each mark point (**Max 2**) • One correct Boolean term with a + / OR sign • All Boolean terms and operators correct and no other terms present $\overline{A}.C + \overline{B}.C + A.B.\overline{C}$	**2**

Question	Answer	Marks
7(a)	**One** from Symmetric (cryptography / encryption) Quantum (cryptography)	**1**
7(b)	**One** mark per mark point (**Max 4**) MP1 The two keys held by the organisation are a private key and a public key MP2 The **organisation** makes the public key available to anyone who wishes to send them secure transmissions // The **sender** obtains the organisation's public key MP3 The **sender** uses the organisation's public key to encrypt the message / plain text // The **sender** uses the organisation's public key to turn the message into cipher text MP4 The **organisation** uses its **private key** to decrypt the message.	**4**

Question	Answer	Marks
8(a)	To improve the code by making it use minimum resources (CPU, memory/storage, time) // Answer by example: To improve the code by • minimising program storage • minimising CPU time • minimising memory use • minimising program execution time (includes peripheral use as well as CPU)	**1**
8(b)	**One** mark `(a - b + c)` **One** mark `* (c - a)` **One** mark `/ d` Complete answer `(a - b + c) * (c - a) / d`	**3**

Question	Answer	Marks
8(c)	**One** mark per ring (**Max 4**) 	**4**

Question	Answer	Marks
9(a)(i)	**One** from: e.g. • Virtual assistants // AI Agents • Chatbots • Composing music • Self-driving cars • Natural language processing • Fraud detection • Image colouring // photograph enhancement • Visual/Image recognition/facial recognition • Text mining • Computer assisted translation • Large language models (LLMs) • Playing chess • Grading student exams • Weather forecasting	**1**
9(a)(ii)	Use / introduce more **hidden** layers	**1**

Question	Answer	Marks
9(b)	**One** mark per mark point (**Max 4**) MP1 Initial outputs are compared to expected outputs MP2 … weightings are adjusted to minimise the difference between actual and expected outputs MP3 Calculus is used to find the error gradient in the obtained outputs MP4 … the results are fed back into the neural network MP5 … weightings of each neuron / node are adjusted **as a result of the feedback** MP6 … the process repeats until results are more accurate	**4**

Question	Answer	Marks
10(a)	**One** mark per mark point (**Max 2**) MP1 Use an **exception handling** routine (to respond to unwanted / unexpected events when the program is running) MP2 Use of try … except / catch // Generate some form of error message	**2**
10(b)	**One** mark per mark point (**Max 2**) MP1 Coding errors MP2 User errors MP3 Hardware failure // Losing connection to a device e.g. a printer	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>11</td><td>

One mark per mark point (**Max 7**)
MP1 `LDD 300` seen
MP2 Correct use of `STO` seen (at least once)
MP3 Correct use of `LDD 420` seen
MP4 Correct use of `LDI B`
MP5 Correct use of `ADD A`

<table>
<tr><th>Opcode</th><th>Operand</th></tr>
<tr><td>LDD</td><td>300</td></tr>
<tr><td>STO</td><td>A</td></tr>
<tr><td>LDD</td><td>420</td></tr>
<tr><td>STO</td><td>B</td></tr>
<tr><td>LDI</td><td>B</td></tr>
<tr><td>ADD</td><td>A</td></tr>
<tr><td>STO</td><td>Answer</td></tr>
</table>

MP6 Correct labelling of three addresses `A:`, `B:` and `Answer:`
MP7 Correct contents in `A:` and `B:`
MP8 Correct value in `Answer:`

<table>
<tr><th>Label</th><th>Contents</th></tr>
<tr><td>A:</td><td>86</td></tr>
<tr><td>B:</td><td>150</td></tr>
<tr><td>Answer:</td><td>112</td></tr>
<tr><td></td><td></td></tr>
</table>

</td><td>**7**</td></tr>
</table>

Question	Answer	Marks

12(a)(i)	**One** mark for each correctly completed line (**Max 4**)
	
	`PROCEDURE Push(NewData : STRING)`
	`  IF Top < Max - 1 THEN`
	`  Top ← Top + 1`
	`  StackArray[Top] ← NewData`
	`  ELSE`
	`  OUTPUT "Stack full; new data cannot be added"`
	`  ENDIF`
	`ENDPROCEDURE`
		4

12(a)(ii)	**One** mark per mark point (**Max 2**)
	MP1 Input with variable, with or without prompt
	MP2 Procedure call for Push with parameter used matching input variable
	
	Example answer
	`INPUT MyData`
	`CALL Push(MyData)`
		2

12(b)	**One** mark per mark point (**Max 3**) MP1 Stacks store data in Last In First Out (LIFO) / First In Last Out (FILO) order MP2 Each time a recursive algorithm calls itself data is pushed onto the stack MP3 When the recursive algorithm reaches its base case / starts to unwind MP4 … data is popped from the stack in the **reverse order** to which it was pushed onto it.	**3**