Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/31**

Paper 3 Advanced Theory **May/June 2023**

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 May/June 2023 series for most Cambridge IGCSE, Cambridge International A and AS Level and Cambridge Pre-U 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 descriptors 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.

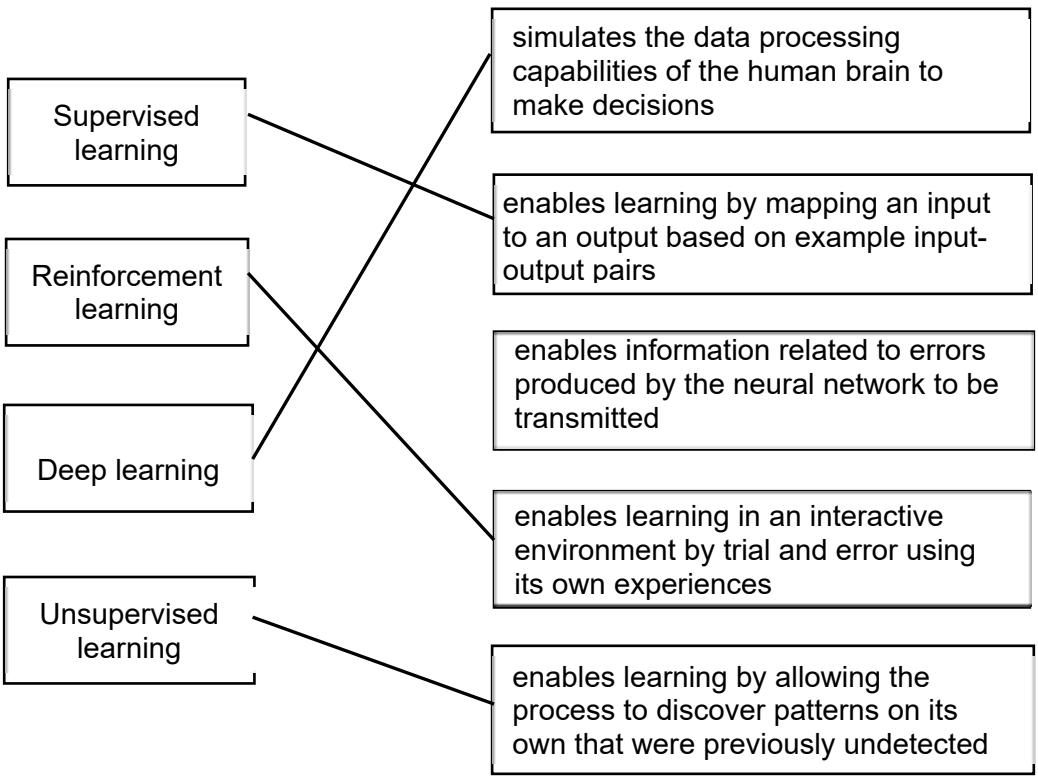
GENERIC MARKING PRINCIPLE 5:

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.

Question	Answer	Marks
1(a)	**One** mark per mark point (**Max 4**) • conversion of 113.75 to binary seen 1110001.11 • exponent for normalisation 7 converted to binary 111 // evidence of binary point moved 7 places // evidence of finding exponent = 7 • system 1 answer • system 2 answer showing correct version from system 1 **System 1** **Mantissa**: 0 1 1 1 0 0 0 1 1 1 **Exponent**: 0 0 0 1 1 1 **System 2** **Mantissa**: 0 1 1 1 0 0 0 1 **Exponent**: 0 0 0 0 0 1 1 1	**4**
1(b)	**One** mark per mark point (**Max 2**) • the mantissa in system 2 does not have enough bits to store the whole binary number // 10 bits required and only 8 bits available • so precision is lost / the number is truncated	**2**

Question	Answer	Marks
2(a)	**One** mark for each correct line connecting a machine learning technique to its most appropriate description (**Max 4**). **Machine learning category** **Description**  	**4**

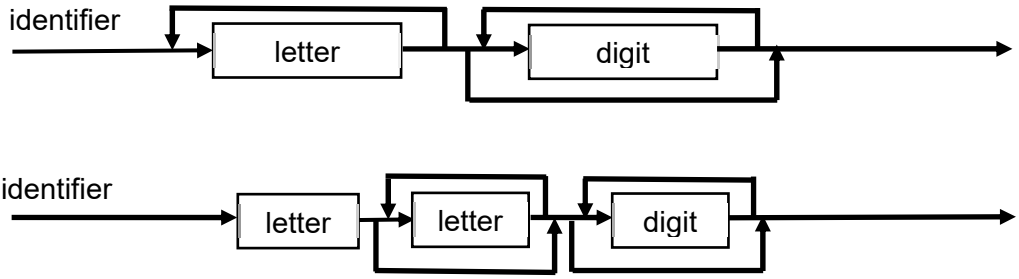
Question	Answer	Marks
2(b)	**One** mark per mark point (**Max 2**) • to find the optimal / shortest / most cost-effective route • … between two nodes in a • … based on distance / cost / time.	**2**

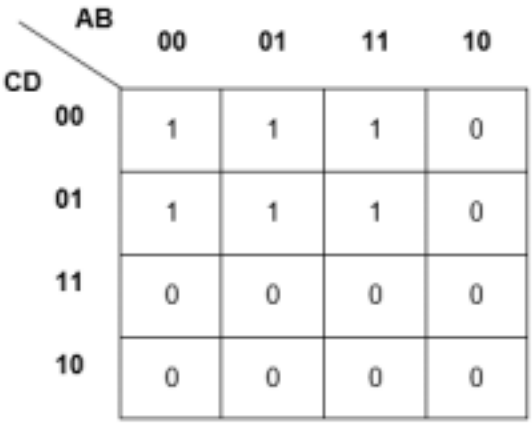
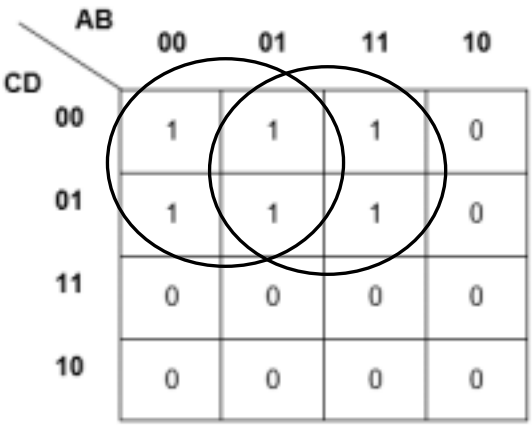
Question	Answer	Marks
3(a)	**One** mark for each correct hash value (**Max 2**) **Record key** / **Hash value** 1030 / 1 1050 / 0 1025 / 2	**2**
3(b)	**One** mark per mark point (**Max 4**) MP1 A collision occurs when the record key doesn't match the stored record key MP2 … this means the determined storage location has already been used for another record. **If the record is to be stored** MP3 Search the file linearly MP4 … to find the next available storage space (closed hash) MP5 Search the overflow area linearly MP6 … to find next available storage space (open hash) **If the record is to be found** MP7 … search the overflow area linearly (open hash) until the matching record key is found MP8 … search linearly from where you are (closed hash) until the matching record key is found MP9 If not found record is not in file	**4**

Question	Answer	Marks
4(a)	**One** mark per mark point (**Max 2**) • `TYPE Prime` • `= (2, 3, 5, 7, 11, 13, 17, 19)` Example answer `TYPE Prime = (2, 3, 5, 7, 11, 13, 17, 19)`	**2**
4(b)	**One** mark per mark point (**Max 2**) • `TYPE TDayPointer` • `= ^STRING //^DayOfWeek` Example answer `TYPE TDayPointer = ^STRING // ^DayOfWeek`	**2**

Question	Answer	Marks
5(a)	**One** mark per mark point (**Max 2**) • Circuit switching is used where a dedicated path needs to be sustained throughout the call / communication // where the **whole** bandwidth is required // where a real time communication is used. • A typical application is standard voice communications / video streaming / private data networks	**2**
5(b)	**One** mark per benefit (**Max 2**) MP1 Whole of bandwidth is available MP2 Dedicated communication channel increases the quality of transmission MP3 Data is transmitted with a fixed data rate MP4 No waiting time at switches MP5 Suitable for long continuous communication MP6 Fast method of data transfer MP7 Data arrives in the same order as it was sent MP8 Data can't get lost MP9 Data all follows the same path / route MP10 Better for real-time MP11 Simple method of data transfer. **One** mark per drawback (**Max 2**) MP1 A dedicated connection makes it impossible to transmit other data even if the channel is free MP2 Not very flexible MP3 No alternative route in case of failure MP4 The time required to establish the physical link between the two stations can be too long MP5 The need to establish a dedicated path for each connection can have cost implications MP6 Dedicated channels require the whole bandwidth / bandwidth can't be shared	**4**

Question	Answer	Marks
6(a)	**One** mark per correct valid/invalid and reason combination (**Max 3**) `DPAD99$` – Valid Reason – 4/multiple letters followed by 2/multiple digits followed by a symbol. `DAD#95` – Invalid Reason – The symbol comes before the digits – it should be after. `ADY123?` – Invalid Reason – The ? is not a valid symbol.	**3**
6(b)	`<symbol> ::= $ | % | & | @ | #` `<letter> ::= A | D | P | R | Y`	**1**

Question	Answer	Marks
6(c)	**One** mark per mark point (**Max 4**) • begins with a letter • letter can repeat and digit present • digit can repeat or can be bypassed • correct structure – name, boxes and arrows (in and out). Example answers:  	**4**

Question	Answer	Marks
7(a)	**Two** marks if no errors present **One** mark if one error present 	**2**
7(b)	**One** mark for each correct loop **(Max 2)** 	**2**

Question	Answer	Marks
7(c)	**One** mark for each mark point (**Max 2**) • Any correct Boolean term • Boolean terms and operator correct and no other terms present (Z =) $\bar{A}\bar{C} + B\bar{C}$	**2**
7(d)	**One** mark for simplest form (**Max 1**) (Z =) $\bar{C}(\bar{A} + B)$	**1**

Question	Answer	Marks
8	**One** mark per mark point (**Max 3**) MP1 A **large number** of computer processors / separate computers connected together MP2 … simultaneously performing a set of coordinated computations // collaborative processing MP3 network infrastructure **MP4** communicate using a message interface / by sending messages.	**3**

Question	Answer	Marks
9(a)	**One** mark per mark point (**Max 2**) MP1 To provide better security MP2 … by using two different keys / a public key and a private key MP3 One of the keys is used to encrypt the message MP4 … the **matching key** is used to decrypt the message.	**2**
9(b)	**One** mark per benefit (**Max 2**) MP1 Provides security based on laws of physics rather than mathematical algorithms, so more secure. MP2 To protect the security of data transmitted over fibre optic cables. MP3 Virtually unhackable. MP4 The performance of quantum cryptography is continuously improved, making it suitable for most valuable government/industrial secrets. MP5 Longer keys can be used MP6 Eavesdropping can be detected **One** mark per drawback (**Max 2**) MP1 Lacks many vital features such as digital signature, certified mail, etc. MP2 High cost of purchasing / maintaining equipment required. MP3 Currently only works over relatively short distances. MP4 Error rates are relatively high as technology is still being developed. MP5 Polarisation of light can change during transmission. MP6 Allows criminals and terrorists to hide their communications.	**4**

Question	Answer	Marks

10	**One** mark for each correctly completed line (**Max 5**)
	
	`DECLARE Account : STRING`
	`OPENFILE "ActiveFile.txt" FOR READ`
	`OPENFILE "ArchiveFile.txt" FOR WRITE`
	`WHILE NOT EOF("ActiveFile.txt")`
	`  READFILE "ActiveFile.txt", Account`
	`  IF Account = "" THEN`
	`  WRITEFILE "ArchiveFile.txt", "Account not present"`
	`  ELSE`
	`  WRITEFILE "ArchiveFile.txt", Account`
	`  ENDIF`
	`ENDWHILE`
	`CLOSEFILE "ActiveFile.txt"`
	`CLOSEFILE "ArchiveFile.txt"`
		5

Question	Answer	Marks

11(a)	**One** mark per mark point (**Max 3**)
	• correctly defined constant
	• correctly defined array
	• three correctly defined integers
	
	`CONSTANT MaxSize = 60`
	
	`DECLARE Queue : ARRAY[1:60] OF STRING // DECLARE Queue : ARRAY[0:59] OF STRING // DECLARE Queue : ARRAY[1:MaxSize] OF STRING // DECLARE Queue : ARRAY[0:MaxSize - 1] OF STRING`
	
	`DECLARE FrontPointer : INTEGER`
	`DECLARE RearPointer : INTEGER`
	`DECLARE Length : INTEGER`
		3

Question	Answer	Marks

11(b)	**One** mark for each correctly completed line (**Max 4**)
	
	`FUNCTION Dequeue RETURNS STRING`
	`  DECLARE Item : STRING`
	`  IF Length > 0 THEN`
	`  Item ← Queue[FrontPointer]`
	`  FrontPointer ← FrontPointer + 1`
	`  Length ← Length – 1`
	`  IF Length = 0 THEN`
	`  CALL Initialise // procedure to reset the pointers`
	`  ELSE`
	`  IF FrontPointer > MaxSize THEN`
	`  FrontPointer ← 1`
	`  ENDIF`
	`  ENDIF`
	`  ELSE`
	`  OUTPUT "The print queue was empty – error"`
	`  Item ← ""`
	`  ENDIF`
	`  RETURN Item`
	`ENDFUNCTION`
		4

11(c)	**One** mark per mark point (**Max 4**) MP1 (Two stacks are required) so that the second stack can reverse the order of the first stack. MP2 Stack 1 operates as the queue with the newest elements at the bottom. Stack 2 is empty. MP3 To add an element, pop all the elements from stack 1 and push onto stack 2. MP4 Push the new element onto either stack. MP5 Pop all the elements of stack 2 back onto stack 1.	**4**

Question	Answer	Marks
12(a)	**One** mark per mark point (**Max 2**) • A process using a function or procedure defined in terms of itself / calls itself. • A recursive process must have a base case (which is a way to return without making a recursive call) // terminating solution // concept of unwinding described • There must (also) be a general case where the recursive call takes place.	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>12(b)</td><td>

One mark per mark point (**Max 5**)
- `Call number` column correct
- `Function call` and `Number` columns correct
- `Result` column down to base case (Winding) rows 1–6 correct
- `Result` column down from base case (Unwinding) rows 7–10 correct
- Return value column correct

<table>
<tr><th>Call number</th><th>Function call</th><th>Number</th><th>Result</th><th>Return value</th></tr>
<tr><td>1</td><td>Fib(5)</td><td>5</td><td>Fib(4) + Fib(3)</td><td></td></tr>
<tr><td>2</td><td>Fib(4)</td><td>4</td><td>Fib(3) + Fib(2)</td><td></td></tr>
<tr><td>3</td><td>Fib(3)</td><td>3</td><td>Fib(2) + Fib(1)</td><td></td></tr>
<tr><td>4</td><td>Fib(2)</td><td>2</td><td>Fib(1) + Fib(0)</td><td></td></tr>
<tr><td>5</td><td>Fib(1)</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>6</td><td>Fib(0)</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>(4)</td><td>Fib(2)</td><td>2</td><td>1 + 0</td><td>1</td></tr>
<tr><td>(3)</td><td>Fib(3)</td><td>3</td><td>1 + 1</td><td>2</td></tr>
<tr><td>(2)</td><td>Fib(4)</td><td>4</td><td>2 + 1</td><td>3</td></tr>
<tr><td>(1)</td><td>Fib(5)</td><td>5</td><td>3 + 2</td><td>5</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table>

</td><td>5</td></tr>
</table>