CAMBRIDGE
International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/33**

Paper 3 Advanced Theory **May/June 2024**

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 2024 series for most Cambridge IGCSE, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

PUBLISHED

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.

Question	Answer	Marks
1(a)	**One** mark per mark point (**Max 3**) MP1 conversion of exponent 001001 to 9 MP2 application of exponent to mantissa to go from 0.100111100 to 100111100 // 256 + 32 + 16 + 8 + 4 seen // 64/128 + 8/128 + 4/128 + 2/128 + 1/128 = 79/128 // 1/2 + 1/16 + 1/32 + 1/64 + 1/128 = 79/128 MP3 correct answer = 316	**3**
1(b)	**One** mark per mark point (**Max 3**) MP1 number converted to binary 10011001.01 // number converted to positive 102.75, reversed bits and 1 added. (0)1100110.11 ⇨ 10011001.00 ⇨ 10011001.01 // -128 + 16 + 8 + 1 + 0.25 = –102.75 MP2 exponent = 7 // Moving binary point the correct number of places MP3 correct answer **Mantissa** **Exponent** 1 0 0 1 1 0 0 1 0 1 0 0 0 1 1 1	**3**

Question	Answer	Marks
2(a)	**Two** marks for all protocols in correct position **One** mark for at least two protocols in correct position Application Transport Internet Link	**2**
2(b)	**One** mark per mark point (**Max 2**) MP1 The transport layer is responsible for delivery of data from the source host to the destination host MP2 It is where data is broken up into packets **and** sent to the internet layer MP3 Adds the sequence number to the packet header MP4 It establishes end to end contact MP5 It ensures data arrives error free // It retransmits packets if lost.	**2**
2(c)	**One** mark for name of protocol and one mark for expansion (**Max 2**) HTTP(S) – responsible for correct transfer of files / hypertext documents that make up web pages on the world wide web FTP – used when transferring files from a server to a client on a network POP3 – handles the receiving of emails IMAP – handles the receiving of emails SMTP – handles the sending of emails BitTorrent – provides peer-to-peer file sharing	**2**

Question	Answer	Marks
3(a)	**One** mark per mark point (**Max 2**) *non-composite data types* MP1 Non-composite data types can both be user-defined **or** primitive MP2 Non-composite data types do not refer to other data types in their definition / contain one data type in their definition MP3 Non-composite data types can be primitive/enumerated/pointer **One** mark per mark point (**Max 2**) *composite data types* MP4 Composite data types can be user-defined **or** primitive MP5 Composite data types refer to other data types in their definition/contain more than one data type in their definition MP6 Composite data types can be record/set/class	**3**

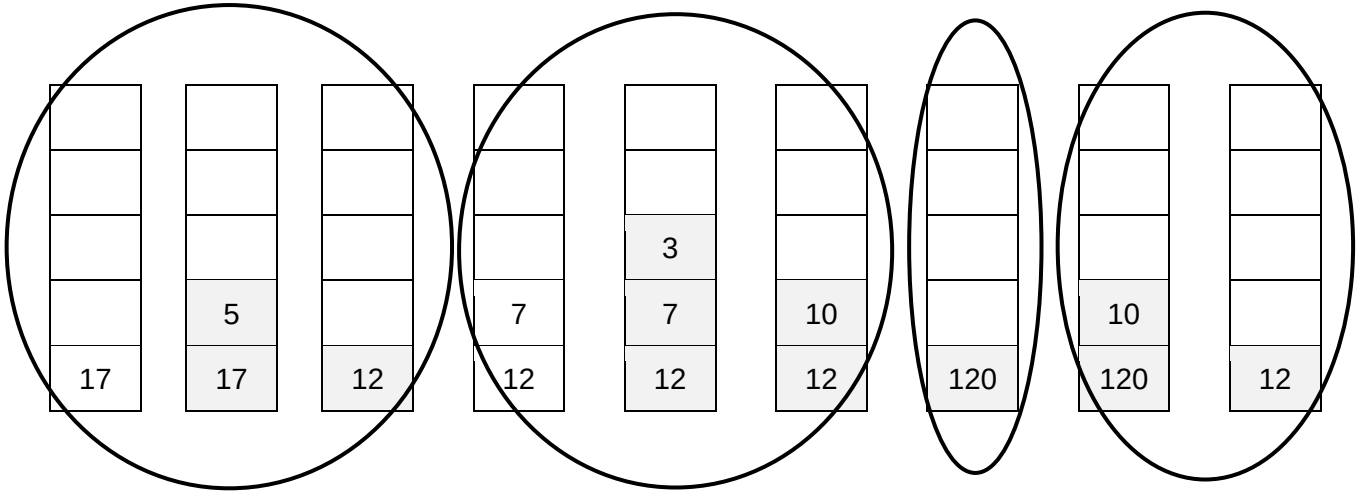
3(b)	**One** mark for `TYPE FootballClub` and `ENDTYPE` correct
	One mark for every two correct declarations
	
	Example answer
	`TYPE FootballClub`
	`    DECLARE TeamName : STRING`
	`    DECLARE DateOfJoining : DATE`
	`    DECLARE MainTelephone : STRING`
	`    DECLARE ManagerName : STRING`
	`    DECLARE NumberOfMembers : INTEGER`
	`    DECLARE LeaguePosition : INTEGER`
	`ENDTYPE`
		4

Question	Answer	Marks
4(a)	**One** mark per mark point (**Max 2**) MP1 Sequential access method searches for **records** one after the other MP2 … from the physical start of the file until the **record** is found/the end of file.	**2**
4(b)	**One** mark per mark point (**Max 3**) MP1 For serial files, records are stored in chronological order MP2 … every record needs to be checked until the record is found, or all records have been checked. MP3 For sequential files, records are stored in order of a key field/index, and it is the key field/index that is compared. MP4 … every record is checked until the record is found, or the key field of the current record is greater than the key field of the target record.	**3**

Question	Answer	Marks

5(a)	**One** mark per correct term (**Max 3)**
	`(5 + 2)`
	`/ (9 – 3)`
	`* 3`
	
	Complete correct answer
	`((5 + 2) / (9 - 3)) * 3`
		3

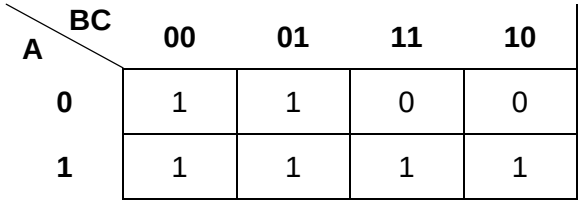
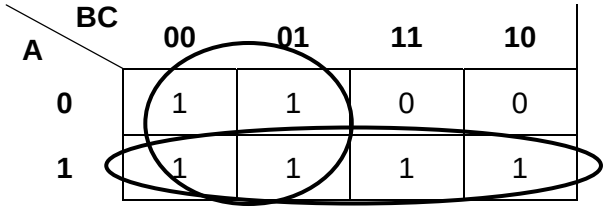
5(b)	**One** mark `7 3 +` **One** mark `2 8 * - 6 /` **Complete answer** `7 3 + 2 8 * - 6 /`	**2**

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

<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="4">Working space</th><th></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>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</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>0</td><td>0</td><td>0</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>0</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>0</td><td>0</td><td>1</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>1</td><td>1</td></tr>
</table></td><td>3</td></tr>
<tr><td>6(b)</td><td>Two marks for all five correct terms and no extras
One mark for any three correct terms

(Z =) $\overline{A}.\overline{B}.\overline{C} + \overline{A}.\overline{B}.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$</td><td>2</td></tr>
</table>

Question	Answer	Marks
6(c)(i)	**Two** marks if all correct **One** mark if one error present 	**2**
6(c)(ii)	**One** mark for each correct loop **(Max 2)** 	**2**
6(c)(iii)	A + $\underline{B}$	**1**

Question	Answer	Marks
7(a)	**One** mark per point (**max 3**) MP1 A digital certificate is an electronic/online document. MP2 used to authenticate/prove the identity of a website/the online identity of an individual/organisation MP3 typically issued by a CA MP4 For example: it contains information identifying a website owner/individual and a public key	**3**
7(b)	**One** mark per point (**max 2**) MP1 The digital certificate **provides the public key** MP2 … that can be used to validate the private key associated with the organisation/website/digital signature	**2**

Question	Answer	Marks

8(a)	**One** mark for each correctly completed clause (**Max 3**)
	
	`(23) feature(sliding_doors).`
	`(24) available(sliding_doors, minivan).`
	`(25) unavailable(sliding_doors, hatchback).`
		3

8(b)	`(Options =) sunroof, reversing_camera`	**1**

8(c)	**One** mark per mark point (**Max 4**)
	MP1 `feature(F)`
	MP2 `bodystyle(B)`
	MP3 `unavailable(F, B)`
	MP4 all correct Boolean operators and punctuation (allow , for AND) and no additional lines of code
	
	Example answers
	
	`may_choose_option(F, B)`
	`IF`
	
	`feature(F) AND bodystyle(B) AND NOT unavailable(F, B).`
	
	`feature(F), bodystyle(B), NOT unavailable(F, B).`
		4

Question	Answer	Marks
9	**One** mark per mark point (**Max 3**) MP1 Deep learning learns by finding hidden patterns that are **undetectable to humans**. MP2 It structures algorithms in layers: **input** layer, **hidden** layers and **output** layer. MP3 … to create an artificial neural network to learn and make intelligent decisions on its own. MP4 It is trained using large quantities of unlabelled data. MP5 Deep learning requires/uses **a large number of** hidden layers. MP6 … the larger the number of layers, the higher the level of success.	**3**

Question	Answer	Marks
10(a)	**One** mark The elements are sorted according to the compare function / in ascending / descending order.	**1**

10(b)	**One** mark for each correctly completed line (**Max 5**)
	`DECLARE Names : ARRAY[1:100000] OF STRING`
	`DECLARE TopOfList : INTEGER`
	`DECLARE EndOfList : INTEGER`
	`DECLARE CurrentItem : INTEGER`
	`DECLARE ToFind : STRING`
	`DECLARE Found : BOOLEAN`
	`DECLARE NotInList : BOOLEAN`
	`TopOfList ← 1`
	`EndOfList ← 100000`
	
	`OUTPUT "Which name do you wish to find? "`
	`INPUT ToFind`
	`Found ← FALSE`
	`NotInList ← FALSE`
	`WHILE` **`Found = FALSE`** `AND` **`NotInList = FALSE`**
	`  CurrentItem ← (TopOfList + EndOfList) DIV 2`
	`  IF` **`ToFind = Names[CurrentItem]// Names[CurrentItem] = ToFind`** `THEN`
	`  Found ← TRUE`
	`  ELSE`
	`  IF TopOfList >= EndOfList THEN`
	`  NotInList ← TRUE`
	`  ELSE`
	`  IF ToFind > Names[CurrentItem] THEN`
	`  TopOfList ← CurrentItem + 1`
	`  ELSE`
	`  EndOfList ← CurrentItem – 1`
	`  ENDIF`
	`  ENDIF`
	`  ENDIF`
	`ENDWHILE`
		5

Question	Answer	Marks

10(b)	`IF Found = TRUE THEN`
	`  OUTPUT "Item found at position ", CurrentItem, " in array"`
	`ELSE`
	`  OUTPUT "Item not in array"`
	`ENDIF`	

10(c)	**One** mark from MP1 Big O for a binary search is $O(Log_2\ n)$. MP2 Big O notation is used to indicate the time/space complexity of an algorithm. **One** mark from MP3 The time taken to complete the search increases logarithmically as the number of search items increases linearly MP4 The time taken to complete the search increases linearly as the number of search items increases exponentially MP5 As the search field is repeatedly getting smaller, the number of comparisons made before the item is found, or the number of items runs out, is relatively small.	**2**

Question	Answer	Marks
11(a)	**One** mark per mark point (**Max 2**) MP1 Uses **hard-wired code/control units** MP2 Uses relatively few instructions / simple instructions MP3 Uses relatively few addressing modes MP4 Makes use of a single-cycle for each instruction MP5 Makes use of fixed length / fixed format instructions MP6 Makes use of general-purpose registers MP7 Pipelining is **straightforward to apply** MP8 The design emphasis is on the **software** MP9 Processor chips require few transistors.	**2**
11(b)	**One** mark per mark point (**Max 3**) MP1 Once the processor detects an interrupt at the start/end of the fetch-execute cycle MP2 … the current program is temporarily stopped and the **status of each register** stored on the stack. MP3 After the interrupt has been serviced/the Interrupt Service Routine (ISR) has been executed … MP4 … the **registers** can be restored to its original status before the interrupt was detected // … the data can be restored **from the stack**.	**3**
11(c)	**One** mark per mark point (**Max 3**) MP1 Pipelining adds an additional complexity // there could be a number of instructions still in the pipeline when the interrupt is received MP2 All the instructions currently in operation are usually discarded except for the last one/the one at write back MP3 … the interrupt handler routine is applied to the remaining instruction. MP4 Once the interrupt has been serviced the processor can restart with the next instruction in the sequence.	**3**