CAMBRIDGE
International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/32**

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.

 [Turn over

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 2**) MP1 When the number of bits in the mantissa is raised, the precision / accuracy of the number represented increases // when the number of bits in the mantissa is lowered, the precision / accuracy of the number represented reduces. MP2 When the number of bits in the exponent is reduced, the range of numbers that can be represented is reduced // when the number of bits in the exponent is increased, the range of possible numbers that can be represented increases. MP3 When the range increases the accuracy decreases // When the range decreases the accuracy increases.	**2**
1(b)	**One** mark per mark point (**Max 3**) • number converted to binary e.g. 54.8125 = 00110110.1101 // Fractions method 1/2 + 1/4 + 1/16 + 1/32 + 1/128 + 1/256 + 1/1024 = 877/1024 // 32 + 16 + 4 + 2 + 0.5 + 0.25 + 0.0625 / (1/2 + 1/4 + 1/16) • exponent = 6 // Moving binary point the correct number of places • correct answer **Mantissa** **Exponent** 0 1 1 0 1 1 0 1 1 0 1 0 0 1 1 0	**3**

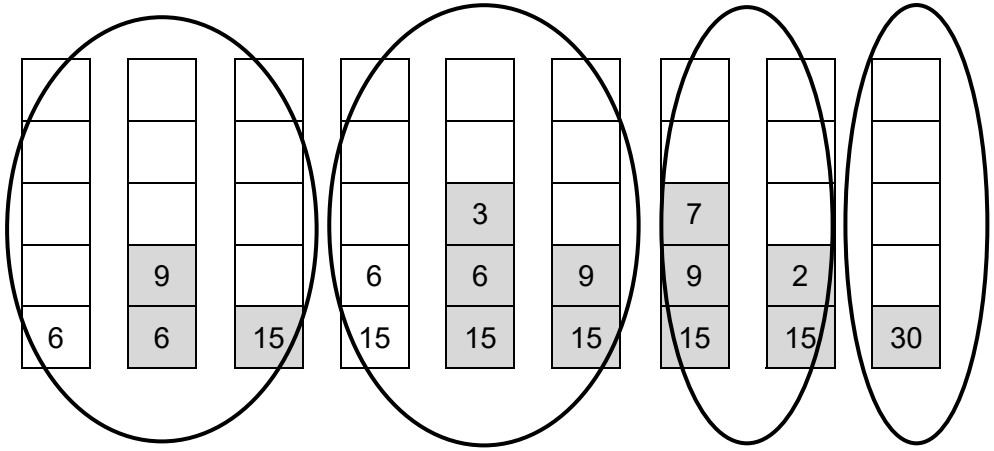
Question	Answer	Marks
2(a)	**One** mark per mark point (**Max 2**) MP1 Protocols provide a standard set of rules that enables **successful** data transfer between devices. MP2 Allows communication between devices on different platforms. MP3 Makes communications independent of software and hardware.	**2**
2(b)	**One** mark per mark point MP1 Sending - SMTP MP2 Receiving – POP3 // IMAP // Post Office Protocol 3	**2**
2(c)	**One** mark per mark point (**Max 3**) MP1 BitTorrent allows the sharing of files between thousands of users who are connected together over the **internet**. MP2 It allows **more** users to share files with each other than would be the case with a peer-to-peer network. MP3 Users share files **directly** with each other // the users' computers are acting as peers MP4 … no web server / central device is used // all users are of equal status.	**3**

Question	Answer	Marks
3(a)	**One** mark per mark point (**Max 3**) MP1 A data type that is defined without referencing another data type. MP2 It can be a primitive data type found in a programming language **or** a user-defined data type. MP3 Example – enumerated data type / pointer data type / allow a correct example of an enumerated or pointer data type declaration.	**3**

3(b)	**One** mark per mark point (**Max 4**)
	MP1 Type statement fully correct
	MP2 `DEFINE EvenNumbers`
	MP3 Correct list of values in brackets
	MP4 `: <set identifier>` from Type statement used
	
	Example answer
	`TYPE Numbers = SET OF INTEGER`
	`DEFINE EvenNumbers (2, 4, 6, 8, 10, 12): Numbers`
		4

Question	Answer	Marks
4	**One** mark per mark point (**Max 4**) MP1 Sheila's computer uses an algorithm to generate a matching pair of keys private and public MP2 Sheila's computer sends Fred's computer Sheila's public key // Fred's computer acquires Sheila's public key MP3 Fred's computer encrypts the document/plain text using Sheila's public key to create cipher text MP4 Fred's computer sends the **cipher text** to Sheila's computer The cipher text can only be decrypted using Sheila's private key // Sheila's computer uses Sheila's private key to decrypt the cipher text.	**4**

Question	Answer	Marks
5(a)	**One** mark `7 2 - 8 +` **One** mark `9 5 - /` **Complete answer** `7 2 - 8 + 9 5 - /`	**2**

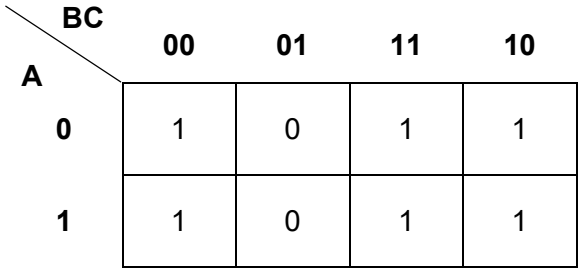
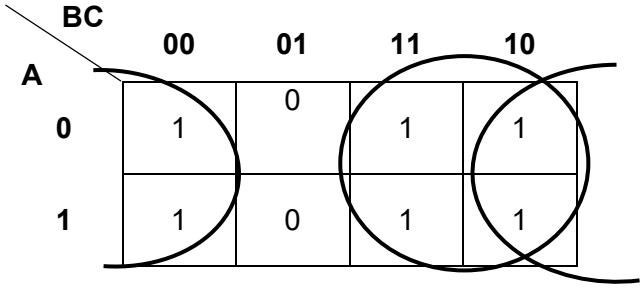
Question	Answer	Marks
5(b)	**One** mark per ring (**Max 4)** 	**4**

5(c)	**One** mark per correct term (**Max 3)**
	`(a – c + b)`
	`* (d + b)`
	`/c`
	
	Complete correct answer
	`(((a – c) + b) * (d + b))/c`
	
	Or
	
	`(a – c + b) * (d + b)/c`
		3

Question	Answer	Marks
6(a)	**One** mark for working, all five columns P, Q, R, S and T **One** mark for first four rows of column Z **One** mark for second four rows of column Z	**3**

			Working space					
A	**B**	**C**	**P**	**Q**	**R**	**S**	**T**	**Z**
0	0	0	1	1	1	1	1	0
0	0	1	1	1	0	1	0	1
0	1	0	1	0	1	1	0	1
0	1	1	1	0	0	1	0	1
1	0	0	0	1	1	0	0	0
1	0	1	0	1	0	1	0	1
1	1	0	0	0	1	1	0	1
1	1	1	0	0	0	1	0	1

6(b)	**Two** marks for all six correct terms only **One** mark for any three correct terms (Z =) $\overline{A}.\overline{B}.C + \overline{A}.B.\overline{C} + \overline{A}.B.C + A.\overline{B}.C + A.B.\overline{C} + A.B.C$	**2**

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

Question	Answer	Marks
7(a)	**One** mark per mark point (**Max 2**) MP1 Direct access allows a **record** to be found in a file without other records being read. MP2 **Records** are found by using the **key field** of the target record // the location of the **record** is found using a **hashing algorithm**.	**2**
7(b)(i)	**One** mark per mark point (**Max 2**) MP1 In sequential files, an index of all key fields is kept MP2 The index is **searched** for the address of the file location where the target record is stored.	**2**

Question	Answer	Marks
7(b)(ii)	**One** mark per mark point (**Max 2**) MP1 A **hashing algorithm** is used on the **key** field of the record MP2 … to calculate the address of the memory **location** where the target record is expected to be stored. MP3 Method to find a record if it is not at the expected location e.g. linear probing, search overflow area etc.	**2**

Question	Answer	Marks

8(a)	**One** mark for each correctly completed line (**Max 4**)
	`DECLARE Widgets : ARRAY[1:50000] OF STRING`
	`DECLARE TopOfList : INTEGER`
	`DECLARE EndOfList : INTEGER`
	`DECLARE Count : INTEGER`
	`DECLARE ToFind : STRING`
	`DECLARE Found : BOOLEAN`
	`DECLARE NotInList : BOOLEAN`
	`TopOfList ← 1`
	`EndOfList ← 50000`
	`OUTPUT "Enter the name of the item you wish to find "`
	`INPUT ToFind`
	`Found ← FALSE`
	`NotInList ← FALSE`
	`Count ← TopOfList`
	`WHILE` **`Found = FALSE`** `AND` **`NotInList = FALSE // Count <= EndOfList`**
	`  IF` **`ToFind = Widgets[Count] // Widgets[Count] = ToFind`** `THEN`
	`  Found ← TRUE`
	`  ENDIF`
	`  Count ← Count + 1`
	`  IF` **`Found = FALSE AND Count > EndOfList`** `THEN`
	`  NotInList ← TRUE`
	`  ENDIF`
	`ENDWHILE`
	`IF Found = TRUE THEN`
	`  OUTPUT "Item found at position ", Count - 1, " in array"`
	`ELSE`
	`  OUTPUT "Item not in array"`
	`ENDIF`
		4

Question	Answer	Marks
8(b)	**Max 4** **One** mark per mark point (**Max 3**) MP1 Linear search sequentially checks each element of the array / list. MP2 … until the matching element is found, or the end of the array / list is reached. MP3 Binary search finds the mid-point of an array/list **and** determines which side contains the item to be found MP4 … it discards the half of the array/list not containing the search item // … it finds the position of a target value within an array / list by repeatedly halving the target search field. MP5 The binary search requires the elements to be sorted // The linear search does not require the elements to be sorted. MP6 The binary search will usually do many fewer comparisons of records/iterations against the target than a linear search before it finds its target. MP7 Linear search starts at the beginning of the array/list **and** binary search starts in the middle of the array/list. **One** mark per mark point (**Max 2**) MP8 Big O for binary search is O (Log_2 n) MP9 Big O for linear search is O (n) MP10 Big O notation is used to indicate the time / space complexity of an algorithm	**4**

Question	Answer	Marks
9(a)	**One** mark per mark point for up to **two** benefits (**Max 2**) MP1 COMPATIBILTY e.g. Applications that aren't compatible with the host computer can be run on the virtual machine // It is possible to emulate old software on a new system by running a compatible guest operating system as a virtual machine // Software can be tried on different OS on the same hardware. MP2 PROTECTION e.g. The guest operating system has no effect on anything outside the virtual machine other virtual machines or the host computer//Virtual machines are useful for testing as they will not crash the host computer if something goes wrong // Easier to recover if software causes a system crash as virtual machine software protects the host system. MP3 COST e.g. No need to buy extra computers / hardware as multiple virtual machines can be implemented on the same hardware. **One** mark per mark point for up to **two** limitations (**Max 2**) MP4 PERFORMANCE e.g. The performance of the guest operating system will not be as good on a virtual machine as it would be on its own compatible machine because of the extra code / using more RAM/memory space // The performance of the VM is dependent on the capabilities of the host computer // Response times cannot be accurately measured using a virtual machine. MP5 COMPLEXITY e.g. Building an in-house virtual machine can be expensive, time consuming and complex to maintain / set-up. MP6 HARDWARE/SOFTWARE ISSUES e.g. Some hardware/software can't be emulated with a virtual machine // Some of the host machine's hardware can't be directly accessed by the virtual machine.	**4**
9(b)	**One** mark per mark point – host operating system (**Max 2**) MP1 The host operating system is the normal operating system for the host computer / machine. MP2 It has control of all the resources of the host computer / machine. // It can access the physical resources of the host computer / machine. MP3 It provides a user interface to operate the virtual machine software. MP4 It also runs the virtual machine software. **One** mark per mark point – guest operating system (**Max 2**) MP5 The guest operating system **runs** within the virtual machine. MP6 … it controls the virtual hardware/software during the emulation. // It accesses the actual hardware through the virtual machine and host operating system. MP7 It provides a virtual user interface for the emulated hardware/software. MP8 The guest operating system runs under the control of the host operating system.	**3**

Question	Answer	Marks

10(a)	**One** mark for each correctly completed clause (**Max 4**)
	
	Example answer
	`(25) client(jane).`
	`(26) activity(surfing).`
	`(27) choice(jane, surfing).`
	`(28) done(jane, sailing).`
		4

10(b)	`(List =) frankie, erik, henry`	**1**

10(c)	**One** mark per mark point (**Max 4**)
	MP1 `client(C)`
	MP2 `activity(A)`
	MP3 `done(C, A)`
	MP4 all correct Boolean operators and punctuation (allow , for AND). There must be the correct number of terms and no additional lines of code
	
	Example answers
	
	`may_choose_activity(C, A)`
	`IF`
	
	`client(C) AND activity(A) AND NOT done(C, A).`
	
	`client(C), activity(A), NOT (done(C, A)).`
		4

Question	Answer	Marks
11	**One** mark per mark point (**Max 3**) MP1 Reinforcement learning is a machine learning technique based on feedback / rewards / punishment. MP2 … in which an agent learns to behave in an environment by performing the actions and seeing the results of the actions. MP3 … for each good action, the agent gets positive feedback / reward and each bad action receives negative feedback / punishment. MP4 The agent learns automatically using feedback without any labelled data / specific instructions. MP5 Adjust node weightings to achieve the correct outcome. // Using feedback to improve its performance at accomplishing similar tasks.	**3**