CAMBRIDGE
International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/33**

Paper 3 Advanced Theory **October/November 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 October/November 2024 series for most Cambridge IGCSE, Cambridge International A and AS Level 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 1**) • correct answer • statement regarding number losing precision/rounding error **One** mark per mark point for working (**Max 2**) • number converted to binary 201.125 = 11001001.001 // 128 + 64 + 8 + 1 + 0.125 / $^1/_8$ seen • use of the exponent e.g. moving the binary point 8 places / $\times 2^8$. **Mantissa**: 0 1 1 0 0 1 0 0 1 0 **Exponent**: 0 0 1 0 0 0	**3**
1(b)	**One** mark per mark point (**Max 2**) • application of exponent to go from 1.010110011 to 101011.0011 // x 2^5 // movement of binary point 5 places seen • –32 + 8 + 2 + 1 + .125 + .0625 // –32 + 8 + 2 + 1 + $^1/_8$ + $^1/_{16}$ seen // –1 + ¼ + $^1/_{16}$ + $^1/_{32}$ + $^1/_{256}$ + $^1/_{512}$ // –1 + $^{179}/_{512}$ // –$^{333}/_{512}$ **One** mark for correct answer (**Max 1**) • –20.8125 // –$20^{13}/_{16}$	**3**

Question	Answer	Marks
2	**One** mark per mark point (**Max 4)** MP1 low number of instruction formats //low number of instruction sets MP2 uses single-clock cycle instructions MP3 uses fixed length instructions MP4 uses many general-purpose registers MP5 works well with pipelining MP6 hard-wired control unit MP7 makes extensive use of RAM MP8 uses a low number of addressing modes MP9 the design emphasis is on the software.	**4**

Question	Answer	Marks
3(a)	**One** mark per mark point (**Max 3**) MP1 A dedicated circuit / channel is required MP2 The circuit is established before the transmission begins MP3 The circuit lasts for the whole of the transmission // The circuit is closed at the end of the transmission MP4 Data travels in a continuous stream along the same route MP5 Transmission is usually bidirectional.	**3**

Question	Answer	Marks
3(b)	**One** mark for a benefit (**Max 1**) MP1 No need for data to be reassembled // data / frames arrive in the same order in which they were sent MP2 Suitable for real time transmission // fast data transfer rate MP3 The whole of the bandwidth is available **One** mark for a drawback (**Max 1**) MP4 No other transmission can use the same circuit when it is in use // Bandwidth can be wasted as it cannot be used by other messages MP5 Not secure // Can be intercepted as all data travelling along the same route MP6 If there is a problem with the route the transmission ends // No other route is available without first doing the setup MP7 The circuit is always there whether or not it's being used MP8 Can take time to set up before transmission starts.	**2**

Question	Answer	Marks
4	**One** mark per mark point (**Max 4**) MP1 Each layer can only accept input from the next higher layer or the next lower layer MP2 There is an interface between the adjacent layers which is the only interaction between layers MP3 Data is added to the headers as the frames/packets pass through the layers MP4 The interactions are carried out by installed software MP5 User interaction takes place at the highest/Application layer of the stack through protocols associated with that layer of the stack MP6 Direct access to hardware takes place at the lowest/Link layer of the stack.	**4**

Question	Answer	Marks
5(a)	**One** mark per mark point (**Max 3**) MP1 A hashing algorithm is used in direct access methods on random and sequential files MP2 It is a mathematical formula MP3 … used to perform a calculation applied to the key field of the record being searched / stored MP4 The result of the calculation gives the address where the record should be found / stored.	**3**
5(b)	**One** mark per mark point (**Max 2**) MP1 The record is stored in the next free memory space after the one identified by the hashing algorithm // Use linear progression MP2 An overflow area is set up and the record is stored in the next free memory space in the overflow area.	**2**

Question	Answer	Marks
6(a)	**One** mark per mark point (**Max 3**) MP1 A set user-defined data type is a composite data type MP2 … which includes a list of unordered elements MP3 Set theory operations, such as intersection and union, can be applied to these elements MP4 A set data type includes the type of data/data type it uses as part of its definition MP5 All the elements are of the same data type.	**3**

Question	Answer	Marks

6(b)	**One** mark for each mark point (**Max 4**)
	
	MP1 `TYPE SymbolSet/Operators =`
	MP2 `SET OF CHAR`
	MP3 `DEFINE Operators/SymbolSet`
	MP4 `('+', '-', '*', '/', '^')`
	MP5 `: SymbolSet/Operators`
	
	Example answers
	`TYPE SymbolSet = SET OF CHAR`
	`DEFINE Operators ('+', '-', '*', '/', '^') : SymbolSet`
	
	`TYPE Operators = SET OF CHAR`
	`DEFINE SymbolSet ('+', '-', '*', '/', '^') : Operators`
		4

Question	Answer	Marks
7(a)	**One** mark for every two correct products (**Max 3**) $(\mathrm{T} =) \overline{\mathrm{A}}.\overline{\mathrm{B}}.\overline{\mathrm{C}}.\mathrm{D} + \overline{\mathrm{A}}.\overline{\mathrm{B}}.\mathrm{C}.\mathrm{D} + \mathrm{A}.\overline{\mathrm{B}}.\overline{\mathrm{C}}.\mathrm{D} + \mathrm{A}.\overline{\mathrm{B}}.\mathrm{C}.\mathrm{D} + \mathrm{A}.\mathrm{B}.\overline{\mathrm{C}}.\mathrm{D} + \mathrm{A}.\mathrm{B}.\mathrm{C}.\mathrm{D}$	**3**

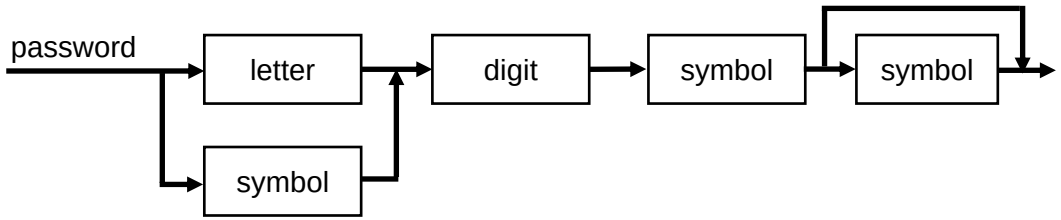
Question	Answer	Marks
7(b)	**Two** marks if no errors present **One** mark if one error present **AB** **CD** | **00** | **01** | **11** | **10** **00** | 0 | 0 | 0 | 0 **01** | 1 | 0 | 1 | 1 **11** | 1 | 0 | 1 | 1 **10** | 0 | 0 | 0 | 0	**2**
7(c)	**One** mark for each correct loop **(Max 2)** **AB** **CD** | **00** | **01** | **11** | **10** **00** | 0 | 0 | 0 | 0 **01** | 1 | 0 | 1 | 1 **11** | 1 | 0 | 1 | 1 **10** | 0 | 0 | 0 | 0	**2**

Question	Answer	Marks
7(d)(i)	**One** mark for each mark point (**Max 2**) • Any correct Boolean term • Boolean terms and operator correct and no other terms present $(\mathrm{T} =) \mathrm{A}.\mathrm{D} + \overline{\mathrm{B}}.\mathrm{D}$ // $\overline{\mathrm{B}}.\mathrm{D}. + \mathrm{A}.\mathrm{D}$	**2**
7(d)(ii)	**One** mark for simplest form (**Max 1**) $(\mathrm{T} =) \mathrm{D}.(\mathrm{A}.\overline{\mathrm{B}})$	**1**

Question	Answer	Marks
8(a)	**One** mark per mark point (**Max 4**) MP1 In segmented memory, the logical / virtual address space is broken into varying sized blocks called segments / sections. MP2 Each segment has a name and size. MP3 During execution segments from logical / virtual memory are loaded into physical memory. MP4 The address is specified by the user MP5 … it contains the segment name and offset value. MP6 Segments are numbered MP7 … and this number is used as an index in the segment map table. MP8 The offset value determines the size of the segment. MP9 A segment map table maps logical / virtual addresses to physical addresses / contains the segment number and offset.	**4**

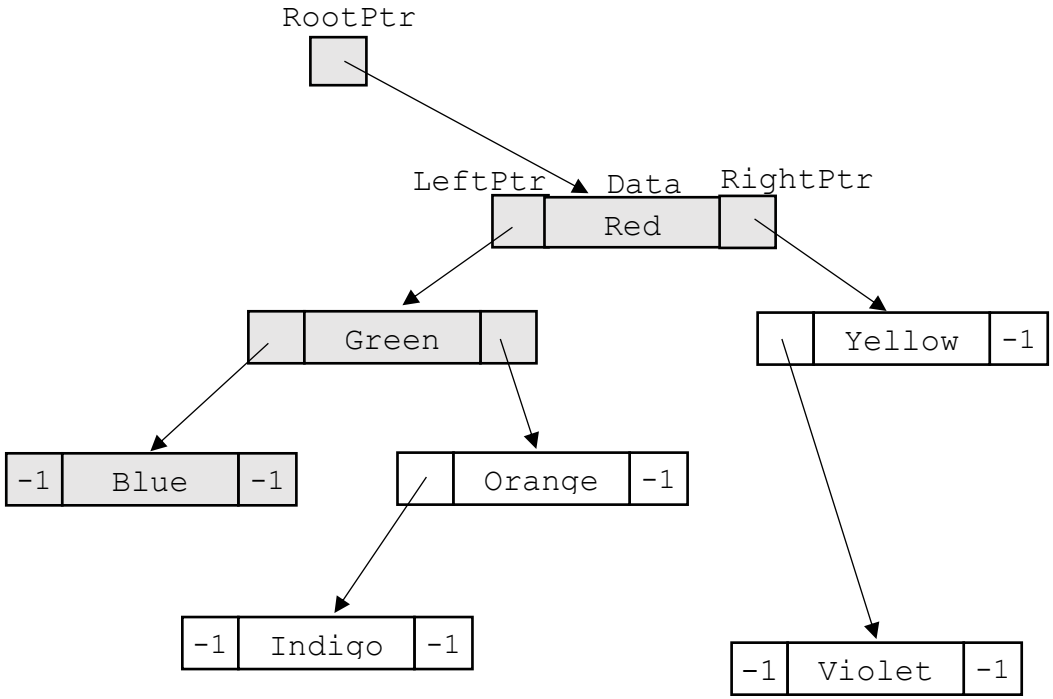
Question	Answer	Marks
8(b)	**One** mark per mark point (**Max 3**) MP1 Disk thrashing is a problem that may occur when virtual memory is being used. MP2 As the main memory fills up, more and more pages need to be swapped in and out of virtual memory. MP3 This swapping leads to a very high rate of hard disk access / excessive disk head movements. MP4 Moving a hard disk read/write head takes a relatively long time / long latency time. MP5 Eventually, more time is spent swapping pages than processing data thrash point, which can cause the program to freeze or not run.	**3**

Question	Answer	Marks
9(a)	To ensure that the attributes are only accessible using the class's own methods/within the class.	**1**
9(b)	**One** mark per mark point (**Max 5**) MP1 Two correct attributes with sensible names and correct data types. MP2 Constructor present. MP3 Two correct setters with exact names and appropriate parameters and data types. MP4 Two correct getters with appropriate names. MP5 Name assigned to pet name getter matches the attribute. **Pet** PetID : STRING PetType : STRING OwnerTelephone : STRING DateRegistered : DATE **PetName : STRING** **OwnerName : STRING** **Constructor()** **SetPetID(APetID : STRING)** **SetDateRegistered(RegDate : DATE)** **GetPetName()** **GetOwnerTelephone()**	**5**

Question	Answer	Marks
10(a)	**One** mark per mark point (**Max 4**) • <operator> ::= + | − | * | / • <label> ::= <letter><digit>|<letter><digit><digit> • <equation> ::= <label> = • <label><operator><label>	**4**
10(b)(i)	**One** mark per mark point (**Max 3)** MP1 begin with either a letter or a symbol MP2 end with either one or two symbols MP3 digit and all other connections and label correct. 	**3**

Question	Answer	Marks

10(b)(ii)	**One** mark per mark point (**Max 2)**
	
	• `<password> ::= <letter><digit><symbol>|`
	• `<letter><digit><symbol><symbol>|<symbol><digit><symbol>|`
	`<symbol><digit><symbol><symbol>`
	
	`<password> ::= <letter><digit><symbol>|`
	`<letter><digit><symbol><symbol>|<symbol><digit><symbol>|`
	`<symbol><digit><symbol><symbol>`
	
	Alternative Answer
	
	One mark per mark point (**Max 2)**
	• All three lines correct
	• Any two lines correct
	
	`<first> ::= <letter>|<symbol>`
	`<last> ::= <symbol>|<symbol><symbol>`
	`<password> ::= <first><digit><last>`
		2

Question	Answer	Marks
11(a)	**One** mark per mark point (**Max 4)** MP1 Four additional nodes with correct data values MP2 Correct null pointers in all added nodes (6) with no extra null pointers where the arrow points to the next node MP3 Correct arrows to represent pointers joining parent nodes to child nodes MP4 All nodes in correct order and no extra data added to pointers.  	**4**

Question	Answer	Marks
11(b)	**One** mark per mark point (**Max 4)** MP1 Correct `Red` and `Green` rows MP2 Correct `Yellow` and `Blue` rows MP3 Correct `Orange`, `Indigo` and `Violet` rows MP4 Correct `FreePtr` with blank row 7	**4**

RootPtr
0

FreePtr
7

Index	LeftPtr	Data	RightPtr
0	1	Red	2
1	3	Green	4
2	6	Yellow	-1
3	-1	Blue	-1
4	5	Orange	-1
5	-1	Indigo	-1
6	-1	Violet	-1
7			

Question	Answer	Marks

11(c)	**One** mark for any correct row (**Max 4**)
	
	`FUNCTION SearchTree(Item : STRING) RETURNS INTEGER`
	`  NowPtr ← RootPtr`
	`  WHILE NowPtr <> -1`
	`  IF BinTree[NowPtr].Data > Item THEN`
	`  NowPtr ← BinTree[NowPtr].LeftPtr`
	`  ELSE`
	`  IF BinTree[NowPtr].Data < Item THEN`
	`  NowPtr ← BinTree[NowPtr].RightPtr`
	`  ELSE`
	`  RETURN NowPtr`
	`  ENDIF`
	`  ENDIF`
	`  ENDWHILE`
	`  RETURN NowPtr`
	`ENDFUNCTION`
		4