Cambridge International AS & A Level

COMPUTER SCIENCE **9618/32**

Paper 3 Advanced Theory **October/November 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 October/November 2023 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 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.

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>1(a)</td><td>One mark per mark point (Max 1)
• conversion of −96.75 to binary e.g., positive 96.75, flip the bits + 1 to give 10011111.01
// –128 + 16 + 8 + 4 + 2 + 1 + 0.25 / ¼ seen

One mark per mark point (Max 2)
• correct mantissa
• correct exponent

Mantissa<table><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>Exponent<table><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr></table></td><td>3</td></tr>
<tr><td>1(b)</td><td>One mark per mark point (Max 3)
MP1 Real numbers (can) have a fractional part (such as $^{1}/3$ and ½) / (such as 0.4 and 0.25)
MP2 The fixed length of the storage means that you can’t store very large / very small numbers
MP3 Binary numbers represent numbers based on powers of 2, with limited fractional representations such as 1/2, 1/4, 1/8, 1/16, etc.
MP4 It isn’t possible to store all fractions with the level of precision provided by this system
MP5 …the fractional part of the number is as close as possible within these constraints.</td><td>3</td></tr>
</table>

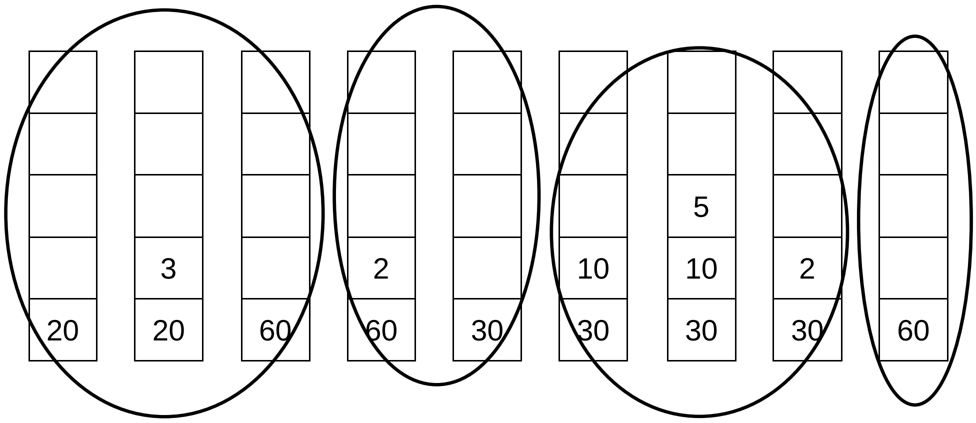
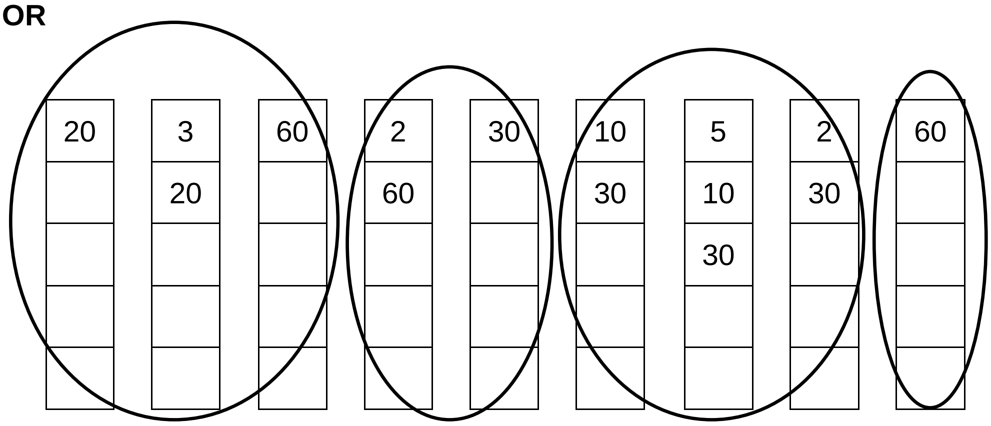
Question	Answer	Marks
2	**One** mark per mark point – composite (**Max 2**) MP1 A (user defined) data type that is a collection of data that can consist of multiple elements MP2 …of different or the same **data types** MP3 …grouped under a single identifier. **One** mark per mark point – non-composite (**Max 2**) MP4 It can be defined without referencing another data type. MP5 It can be a primitive type available in a programming language, **or** a user- defined type.	**4**

Question	Answer	Marks
3(a)	**One** mark per mark point (**Max 2**) MP1 A collision is when the two values / data items in the key field for two records (pass through a hashing algorithm and) result in the same hash value MP2 …so the location identified (by the hashing algorithm) may already be in use // two records cannot occupy the same address.	**2**

Question	Answer	Marks
3(b)	**One** mark per mark point (**Max 3**) MP1 A process of collision resolution is used MP2 Start at the original hashed storage space MP3 …go through the following spaces in a linear fashion MP4 …and **store the data item** in the first available slot. **OR** MP5 Search the overflow area MP6 …go through the following spaces in a linear fashion MP7 …and **store the data item** in the first available slot. **OR** MP8 Each storage space holds a reference to a collection / chain of items MP9 …which can be searched individually. MP10 The **data item is stored** in the first available space in this chain.	**3**

Question	Answer	Marks
4	**One** mark for each correct word (**Max 3**) The protocols in a **stack** determine the interconnectivity rules for a **layered** network model such as the **TCP/IP** model.	**3**

Question	Answer	Marks
5(a)	**One** mark per mark point (**Max 2**) MP1 Virtual memory is used when RAM is running low MP2 …such as when a computer is running many processes at once. MP3 Virtual memory may be used for efficient use of RAM / the processor MP4 …such as if data / programs are not immediately needed, they can be moved from RAM to virtual memory	**2**
5(b)	**One** mark per mark point (**Max 3**) MP1 Disk thrashing is a problem that may occur when frequent transfers between main memory and secondary memory take place // Disk thrashing is a problem that may occur when virtual memory is being used MP2 As main memory fills up, more pages need to be swapped in and out of secondary/virtual memory MP3 This swapping leads to a very high rate of hard disk head movements MP4 Eventually, more time is spent swapping the pages/data than processing the data.	**3**

Question	Answer	Marks
6(a)	**One** mark per ring (**Max 4**).  **OR** 	**4**
6(b)	**One** mark per mark point (**Max 3**) MP1 The (RPN) expression is read from left to right, one item at a time MP2 Each element is checked to see if it as operator or a value MP3 Values are pushed onto a stack until an operator is found MP4 The operator is applied to the last two values on the stack **and** the result is pushed back onto the stack MP5 This repeats until a single value remains, which is the solution.	**3**

Question	Answer	Marks
7(a)	see table below	**1**
7(b)	X= $\overline{A}.\overline{B}.\overline{C}$	**1**

7(a):

A	**B**	**C**	**X**
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Question	Answer	Marks
7(c)	**(Max 2)** for correct working from points shown (T =) $X.Y.Z + X.\overline{Y}.Z + \overline{X}$ Distributive law (T=) $X.Z.(Y + \overline{Y}) + \overline{X}$ Complement law (T=) $X.Z.(1) + \overline{X}$ Identity law (T=) $X.Z + \overline{X}$ Redundancy law (to get final answer) **One** mark for correct answer (T=) $\overline{X} + Z$	**3**

Question	Answer	Marks
8	**One** mark per mark point - working (**Max 3**) May be seen on diagram or in working section MP1 Initialisation – setting Start to 0 MP2 …and the rest of the towns to MP3 Evidence to show values at nodes being updated MP4 Evidence to show 'visited node(s)' MP5 Evidence to **show a correct calculation** of at least one route MP6 Evidence to show more than one route has been calculated for at least one town Correct Answers (**Max 2**) **Two** marks for all six correct values **One** mark for four or five correct values.	**5**

A	B	C	D	E	F
8	12	19	14	16	10

Question	Answer	Marks
9(a)(i)	**Two** marks for all five empty boxes correct **One** mark for any three or four empty boxes correct	**2**

Identifier	Data type	Description
`BasePointer`	`INTEGER`	Points to the bottom of the stack
`TopPointer`	`INTEGER`	Points to the top of the stack
`Stack`	`REAL`	List of decimal numbers stored in the stack

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>9(a)(ii)</td><td>One mark for each correctly completed line (Max 5)
<pre>
CONSTANT MaxSize = 40
DECLARE BasePointer : INTEGER
DECLARE TopPointer : INTEGER
DECLARE Stack : ARRAY[1:40] OF REAL

// initialisation of stack
PROCEDURE Initialise()
   BasePointer ← 1
   TopPointer ← 0
ENDPROCEDURE

// adding an item to the stack
PROCEDURE Push(NewItem)
   IF TopPointer < MaxSize THEN
      TopPointer ← TopPointer + 1
      Stack[TopPointer] ← NewItem
   ENDIF
ENDPROCEDURE
</pre></td><td>5</td></tr>
<tr><td>9(b)</td><td>One mark for linked list and one mark for array (Max 2)

Linked list
MP1 A linked list is a dynamic data structure / not restricted in size
MP2 Has greater freedom to expand or contract by adding or removing nodes as necessary
MP3 Allows more efficient editing using pointers (instead of moving the data).

Array
MP4 An array is a static data structure1 generally fixed in size
MP5 When the array is full, the stack cannot be extended any further.</td><td>2</td></tr>
<tr><td>9(c)</td><td>One mark per mark point (Max 1)
MP1 The compiler must produce object code to

One mark per mark point (Max 3)
MP2 …push return addresses / values of local variables onto a stack
MP3 …with each recursive call // … to set up winding
MP4 …pop return addresses / values of local variables off the stack …
MP5 …after the base case is reached // … to implement unwinding.</td><td>4</td></tr>
</table>

Question	Answer	Marks

10	**One** mark per mark point – SIMD (**Max 2**)
	MP1 Single Instruction, Multiple Data (architecture) // Performs the same operation on multiple different data streams simultaneously.
	MP2 The instructions can be performed sequentially, taking advantage of **pipelining**.
	MP3 Parallel computers **with** multiple processors.
	
	One mark per mark point – MISD (**Max 2**)
	MP4 Multiple Instruction, Single Data (architecture) // Performs different operations on the same data stream.
	MP5 Each processor works on the same data stream independently.
	MP6 Parallel computers **with** multiple processors.
		4

Question	Answer	Marks

11(a)	**One** mark for each correctly completed clause (**Max 4**)
	
	`20 person(carlos).`
	`21 hobby(cycling).`
	`22 enjoys(carlos, cycling).`
	`23 dislikes(carlos, music).`
		4

11(b)	`P = toby, nina`	**1**

11(c)	**One** mark per mark point (**Max 4**)
	• `person(N)`
	• `hobby(H)`
	• `dislikes(N, H)`
	• **all** logical operators correct with no additional code (see example answers)
	
	Example answers:
	
	`might_enjoy(N, H)`
	`IF` **`person(N) AND hobby(H) AND NOT dislikes(N, H`**`)`
	
	`might_enjoy(N, H)`
	`IF` **`NOT dislikes(N, H), person(N), hobby(H)`**
		4

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

One mark for description, for example:

An exception is an event that occurs during the execution of a program that disrupts the normal flow of instructions / causes the program to halt execution

One mark for example:

- Hardware failure // hard disk crash
- Programming error // trying to access out-of-bounds array element // divide by zero error // runtime error
- User error // typing incorrect filename / data type

</td><td>2</td></tr>
<tr><td>12(b)</td><td>

One mark for each correctly completed blank (**Max 7**)

```
DECLARE Customer : TAccount
DECLARE Location : INTEGER
DECLARE MaxSize : INTEGER
DECLARE: FoundFlag : BOOLEAN
DECLARE SearchCustomer : STRING
MaxSize ← 1000
OPENFILE "AccountRecord.dat" FOR RANDOM
Location ← 1
FoundFlag ← FALSE
OUTPUT "Enter the customer's name"
INPUT SearchCustomer
WHILE NOT FoundFlag AND Location <= MaxSize
   SEEK "AccountRecord.dat", Location
   GETRECORD "AccountRecord.dat", Customer
   IF SearchCustomer = Customer.Name THEN
      OUTPUT "Customer found: "
      OUTPUT Customer
      FoundFlag ← TRUE
   ENDIF
   Location ← Location + 1
ENDWHILE
IF NOT FoundFlag THEN
   OUTPUT "Customer does not exist."
ENDIF
```

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