Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/03**

Paper 3 Advanced Theory **For examination from 2021**

MARK SCHEME

Maximum Mark: 75

Specimen

This document has **6** pages. Blank pages are indicated.

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)	= 100.1 (conversion to binary) [1] = 0.1001×2^3 (evidence of shifting binary point appropriately) [1] = 010010000000 0011 (stored as mantissa and exponent) [1]	**3**
1(b)	101101111111 (one's complement of 12 bit mantissa) [1] 101110000000 (two's complement of 12 bit mantissa) [1] 101110000000 0011 (stored as mantissa and exponent) [1]	**3**
1(c)	= 0.0011×2^5 (exponent is 5) = 110 or $(1/8 + 1/16) \times 32$ (calculation using mantissa and exponent) = 6 (denary value)	**3**
1(d)(i)	Not normalised	**1**
1(d)(ii)	First two bits of the mantissa should be different for normalised number // because the mantissa starts with 00	**1**
1(e)	For **each** effect **One** mark for effect and **one** mark for reason Reduction in precision … [1] … as the number of bits in the mantissa has decreased [1] Increase in range … [1] … as the number of bits in the exponent has increased [1]	**4**

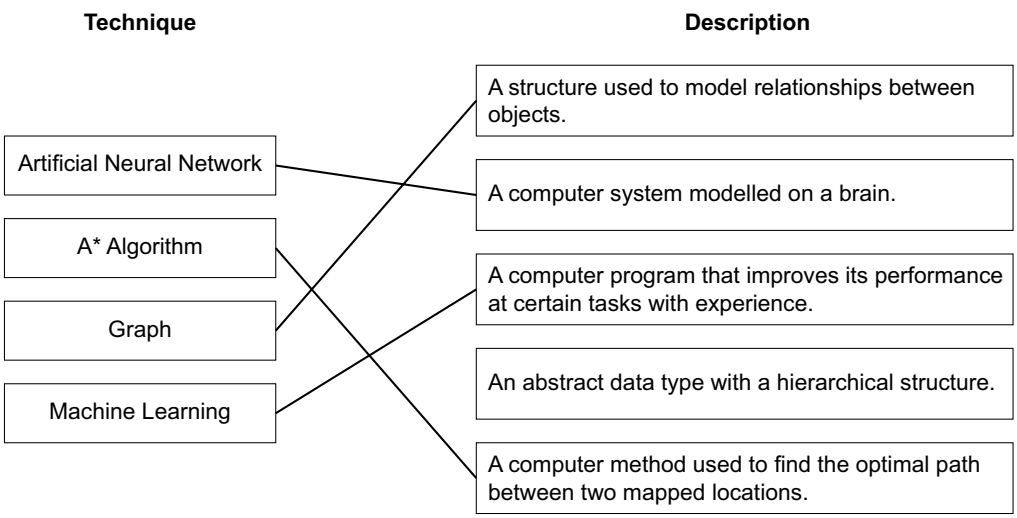
Question	Answer	Marks
2(a)	**One** mark for each correct layer in the correct row • Transport (layer) • Internet (layer) • Network (access layer)	**3**
2(b)(i)	Peer-to-peer	**1**
2(b)(ii)	File sharing	**1**
2(b)(iii)	Any **four** points from (max 4): BitTorrent client software made available A computer joins a swarm by using this to load a Torrent descriptor file A server called a tracker that keeps records of all the computers in the swarm … ... and shares their IP addresses allowing them to connect to each other One computer in the swarm must have a complete copy of the torrent to be shared Torrent is split into small pieces Pieces of the torrent are both downloaded and uploaded Once a computer has a piece it can become a seed and upload Leeches download much more than they upload	**4**

Question	Answer	Marks
2(c)	Any **two** protocols from: **One** mark for protocol and **one** mark for example (max 4) HTTP/HTTPS … [1] … Used for transfer of web pages from server to client [1] FTP … [1] … Used for interactive file transfer [1] SMTP … [1] … Used for sending email messages [1] POP3/IMAP … [1] … Used for retrieval of email messages [1]	**4**

Question	Answer	Marks
3(a)	+ A . B . $\overline{C}$ [1] + A . B . C [1]	**2**
3(b)(i)	**AB** **C** / **00** 0, **01** 0, **11** 1, **10** 0 **1** 0, 1, 1, 0	**1**
3(b)(ii)	**One** mark for each correct loop **AB** **C** / **00** 0, **01** 0, **11** 1, **10** 0 **1** 0, 1, 1, 0	**2**
3(b)(iii)	A . B [1] + B . C [1]	**2**

Question	Answer	Marks
4(a)(i)	Syntax analysis	**1**
4(a)(ii)	Any **two** tasks from (max 2): Parsing/construction of a parse tree [1] Checking that the rules of grammar/syntax have been obeyed [1] Production of an error report [1]	**2**
4(b)	Minimise the execution time	**1**
4(c)	Replace 2 * 6 with the value 12	**1**
4(d)	Remove the second instance of `LDD 436` / line `04` [1] And remove the second instance of `ADD 437`/ line `05` [1] The value required is already stored in the accumulator [1]	**3**

Question	Answer	Marks
5(a)(i)	A single key is used … [1] … for both encryption and decryption [1]	**2**
5(a)(ii)	Any **two** from (max 2): Key has to be exchanged securely [1] Once compromised the key can be used to decrypt both sent and received messages [1] Cannot ensure non-repudiation (proof of integrity and origin of data) [1]	**2**
5(b)	Any **two** from (max 2): Any eavesdropping can be identified [1] Integrity of the key once transferred can be guaranteed (cannot be copied and decrypted at a later date) [1] Longer/more secure keys can be exchanged [1]	**2**

Question	Answer	Marks
6(a)	One mark for each correct line from technique to description  	**4**
6(b)	Any **two** categories from (max 4): **One** mark for category and **one** mark for description Supervised learning … [1] … using known tasks with given outcomes to enable a computer program to improve its performance in accomplishing similar tasks [1] Unsupervised learning … [1] … using a large number of tasks with unknown outcomes to enable a computer program to improve its performance in accomplishing similar tasks [1] Reinforcement learning … [1] … using a large number of tasks with unknown outcomes and the use of feedback to enable a computer program to improve its performance in accomplishing similar tasks [1]	**4**

Question	Answer	Marks

7(a)	`PROCEDURE AddToTree(ByVal NewDataItem : STRING)`
	`// if no free node report an error`
	`  IF FreePointer` **`= -1`** [1]
	`  THEN`
	`  ERROR("No free space left")`
	`  ELSE // add new data item to first node in the free list`
	`  NewNodePointer ← FreePointer`
	`Tree[NewNodePointer].Data ← NewDataItem` [1]
	`  // adjust free pointer`
	`  FreePointer ←` **`Tree[FreePointer].LeftPointer`** [1]
	`  // clear left pointer`
	`  Tree[NewNodePointer].LeftPointer ←` **`-1`** [1]
	`  // is tree currently empty?`
	`  IF` **`RootPointer = -1`** [1]
	`  THEN // make new node the root node`
	`RootPointer ← NewNodePointer` [1]
	`  ELSE // find position where new node is to be added`
	`  Index ← RootPointer`
	`  CALL FindInsertionPoint(NewDataItem, Index, Direction)`
	`  IF Direction = "Left"`
	`  THEN // add new node on left`
	`Tree[Index].LeftPointer ← NewNodePointer` [1]
	`  ELSE // add new node on right`
	`Tree[Index].RightPointer ← NewNodePointer` [1]
	`  ENDIF`
	`  ENDIF`
	`  ENDIF`
	`ENDPROCEDURE`
		8

7(b)	test for base case [1]
	recursive call for left pointer [1]
	output data [1]
	recursive call for right pointer [1]
	order: visit left, output, visit right [1]
	`  IF Pointer <> -1 THEN`
	`  TraverseTree(Tree[Pointer].LeftPointer)`
	`  OUTPUT Tree[Pointer].Data`
	`  TraverseTree(Tree[Pointer].RightPointer)`
	`  ENDIF`
		5

Question	Answer	Marks
8(a)	`LDM` immediate (addressing) [1] `LDD` direct (addressing) [1]	**2**
8(b)	`LDM #12` [1] `ADD 500` [1] `STO 509` [1]	**3**