Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/11**

Paper 1 Theory Fundamentals **October/November 2022**

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 2022 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

This document consists of **11** printed pages.

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.

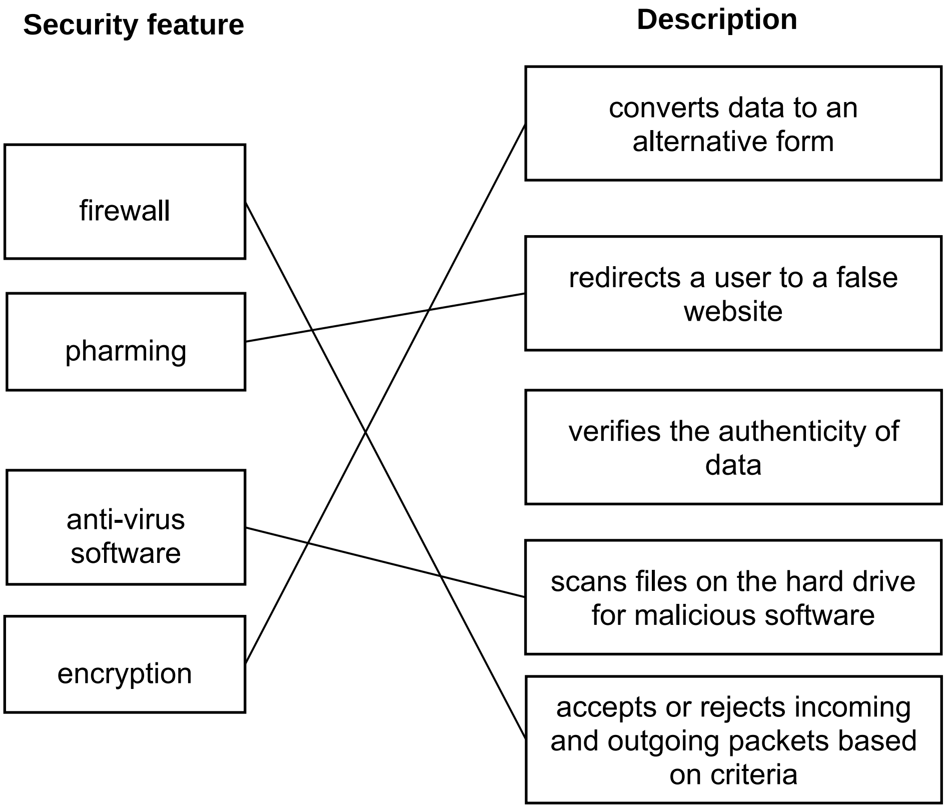
GENERIC MARKING PRINCIPLE 5:

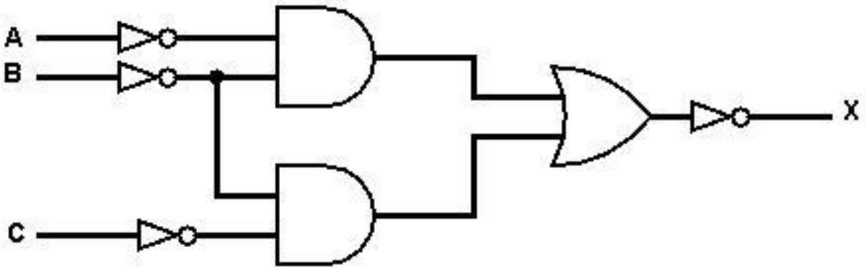
Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

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)(i)	39	**1**
1(a)(ii)	27	**1**
1(a)(iii)	–25	**1**
1(b)	**1 mark** for working, **1 mark** for answer (0011 1110) Working using borrowing: 1 10 10 1 0 ~~10~~ 0 0 ~~10~~ 10 ~~1~~ 0 ~~1~~ ~~1~~ 0 0 1 1 0 1 1 1 0 1 0 1 - 0 0 1 1 1 1 1 0 Working using two's complement: 1 0 1 1 0 0 1 1 Two's complement 1 0 0 0 1 0 1 1 + (1) 0 0 1 1 1 1 1 0 1 1	**2**
1(c)	**1 mark** for similarity, **2 marks** for differences Similarity (**max 1**): • both **can** use 8 bits • both represent each character using a **unique** code • Unicode will contain all the characters that ASCII contains // ASCII is a subset of Unicode Differences (**max 2**): • Unicode can go up to 32 bits per character whereas ASCII is 7 or 8 bits Unicode can represent a wider range of **characters** than ASCII • different **languages** are represented using Unicode, ASCII is only for one language	**3**
1(d)(i)	the number of samples taken **per unit time** / per second	**1**
1(d)(ii)	**1 mark** for each bullet point (**max 2**): • increases the number of bits per sample // larger range of values • which means that the file size increases • makes the sound file more accurate //digital waveform closer to original (analogue) waveform • smaller quantisation errors	**2**

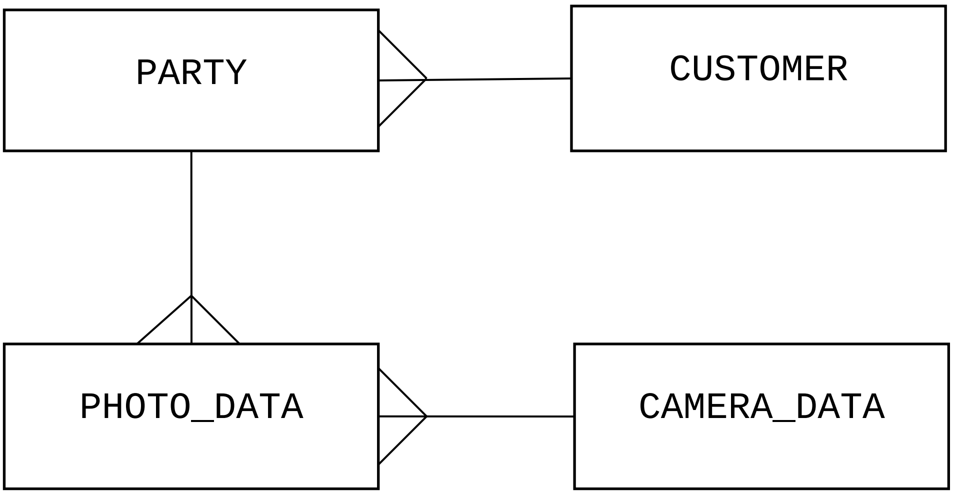
Question	Answer	Marks
2	**1 mark** for each correct line.  Security feature Description firewall pharming anti-virus software encryption converts data to an alternative form redirects a user to a false website verifies the authenticity of data scans files on the hard drive for malicious software accepts or rejects incoming and outgoing packets based on criteria	**4**

Question	Answer	Marks
3(a)	**1 mark** for each bullet point: • NOT A AND NOT B **and** NOT B AND NOT C // A NOR B **and** B NOR C • final OR and NOT gates (with correct inputs) // NOR gate (with correct inputs)  	**2**

Question	Answer	Marks
3(b)	**1 mark** for each set of rows as highlighted:	2

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

Question	Answer	Marks
4(a)	**1 mark** for each correct relationship: • 1:M between CUSTOMER and PARTY • 1:M between PARTY and PHOTO_DATA • 1:M between CAMERA_DATA and PHOTO_DATA	3



Question	Answer	Marks
4(b)	**1 mark** for each bullet point (**max 3**): • no repeating groups of attributes // data is atomic • no partial key dependencies • no non-key dependencies // no transitive dependencies	3

Question	Answer	Marks
4(c)(i)	**1 mark** for the definition, **1 mark** for the example: • definition: a single row in a table • example: from the PHOTO_DATA table	**2**

4(c)(ii)	**1 mark** for each correctly completed empty space:
	• COUNT
	• PhotoID
	• PHOTO_DATA
	• 'CAN*' // 'CAN%'
	`SELECT COUNT(PhotoID)`
	`FROM PHOTO_DATA`
	`WHERE CameraID LIKE 'CAN*'; // WHERE CameraID LIKE`
	`'CAN%';`
		4

4(d)	**1 mark** for each bullet point:
	• ALTER TABLE CAMERA_DATA
	• ADD NumberStored INTEGER
	• , LastUsed DATE;
	`ALTER TABLE CAMERA_DATA`
	`ADD NumberStored INTEGER, LastUsed DATE;`
		3

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>Instructions and data are stored in **the same** memory space / in main memory.</td><td>**1**</td></tr>
<tr><td>5(b)(i)</td><td>**1 mark** for each special purpose register:
Program Counter (PC):
• to **store** the address / location / memory location of the next instruction to be fetched
Index Register (IX):
• to **store** a value that is added to an address to give another address
Status Register (SR):
• to **store flags** which are set by events // from the results of arithmetic and logic operations and interrupt flags</td><td>**3**</td></tr>
<tr><td>5(b)(ii)</td><td>**1 mark** for both rows:
<table>
<tr><th>CPU component</th><th>Data bus</th><th>Address bus</th><th>Control bus</th></tr>
<tr><td>System clock</td><td></td><td></td><td>✓</td></tr>
<tr><td>Memory Address Register (MAR)</td><td></td><td>✓</td><td></td></tr>
</table></td><td>**1**</td></tr>
</table>

Question	Answer	Marks
5(b)(iii)	**1 mark** for each bullet point (**max 2**): • to coordinate / synchronise the actions of other components in the CPU • to send / receive control signals along the control bus • to manage the execution of instructions (in sequence) • to control the communication between the components of the CPU	**2**
5(c)	**1 mark** for each bullet point: • to **send a signal** from a device or process • ... seeking the attention of the processor	**2**
5(d)	**1 mark** for each bullet point (**max 2**). For example: • division by zero // runtime error in a program • attempt to access an invalid memory location • array index out of bounds • stack overflow	**2**

Question	Answer	Marks
6(a)(i)	**1 mark** for each set of highlighted rows.	4

Instruction address	ACC	IX	Memory address					Output
			100	101	110	111	112	
			0	0	66	65	35	
77		0						
78	66							
79								
80								
81								
82				66				
83	1							
84								
85			1					
86		1						
87	65							
88								
89								
81	66							
82								
83	1							
84	2							
85			2					
86		2						
87	35							
88								
89								
90	2							
91	50							
92								2
93								

Question	Answer	Marks
6(a)(ii)	swaps the contents of memory address 100 and 101	1

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(b)(i)</td><td><code>1000 1100</code></td><td>1</td></tr>
<tr><td>6(b)(ii)</td><td><code>1001 0000</code></td><td>1</td></tr>
<tr><td>6(b)(iii)</td><td><code>1101 1111</code></td><td>1</td></tr>
<tr><td>6(b)(iv)</td><td><code>0010 0100</code></td><td>1</td></tr>
<tr><td>6(c)</td><td><b>1 mark</b> for each pair of highlighted rows
<table>
<tr><th>Task</th><th>First pass</th><th>Second pass</th></tr>
<tr><td>Remove comments.</td><td>✓</td><td></td></tr>
<tr><td>Read the assembly language program one line at a time.</td><td>✓</td><td>✓</td></tr>
<tr><td>Generate the object code.</td><td></td><td>✓</td></tr>
<tr><td>Check the opcode is in the instruction set.</td><td>✓</td><td></td></tr>
</table></td><td>2</td></tr>
</table>

Question	Answer	Marks
7(a)	**1 mark** for each benefit (**max 2**): • (main) memory requirements for program are reduced as dynamic link library is loaded only once / when required • the executable file size is smaller because the executable does not contain all the library routines • maintenance not needed to be done by the programmer because the DLL is separate from program • no need to recompile the main program when changes are made to DLL because changes / improvements/ error correction to the DLL file code are done independently of the main program	2
7(b)	**1 mark** for each bullet point (**max 2**): • RAM is assigned into blocks • dynamic allocation of RAM to programs / processes • reclaims unused blocks of RAM • prevents two programs / processes occupying the same area of RAM at the same time • moves data from secondary storage when needed // manages paging, segmentation and virtual memory	2

Question	Answer	Marks
7(c)	**1 mark** for each bullet point (**max 3**): • lossless compression • Run Length Encoding • repeated **sequences** of characters are replaced by • ... a single copy of the character • ... and a counter of the number of characters	**3**
7(d)	**1 mark** for each bullet point (**max 2**): • cache is fast access memory (close to the CPU) • cache stores frequently used instructions / data • ... more cache means more instructions / data can be transferred faster • ... less swapping between RAM and cache • prevents the CPU idling while waiting for data	**2**
7(e)	**1 mark** for each device. • **3D printer**: USB port / COM port • **Monitor**: HDMI / VGA / USB / DisplayPort	**2**

Question	Answer	Marks
8	**1 mark** for bullet point (**max 4**): • CSMA/CD is a protocol used to detect and prevent collisions in a bus topology • before transmitting, a **device** checks if the channel is busy • If the channel is busy the device waits // if the channel is free the data is sent • because there is more than one computer connected to the same transmission medium • ... two workstations can start to transmit at the same time, causing a collision • If a collision is detected by the device, transmission is aborted / a jamming signal is transmitted • both devices wait a (different) **random** time and then try again	**4**

Question	Answer	Marks
9(a)	**1 mark** for each bullet point (**max 2**): • the embedded system is **built into** / integrated into the TV • combination of hardware and software designed for a **specific function** • The system is **not easily changed**/updated by the TV owner	**2**

Question	Answer	Marks
9(b)	**1 mark** for each benefit and **1 mark** for corresponding expansion (**max 2**). For example: • no additional equipment is needed to change • ... enables firmware updates by non-technical users • can be erased and reprogrammed several times • ... so firmware can be updated • ... can erase a particular byte or the whole EEPROM • possible to reprogram / update • ... without removing it from the device	**2**