

Cambridge IGCSE™

COMPUTER SCIENCE**0478/12**

Paper 1 Computer Systems

February/March 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 February/March 2024 series for most Cambridge IGCSE, Cambridge International A and AS Level components, and some Cambridge O Level components.

This document consists of **13** printed pages.

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.

Mark scheme abbreviations

/ separates alternative words / phrases within a marking point

// separates alternative answers within a marking point

underline actual word given must be used by candidate (grammatical variants accepted)

max indicates the maximum number of marks that can be awarded

() the word / phrase in brackets is not required, but sets the context

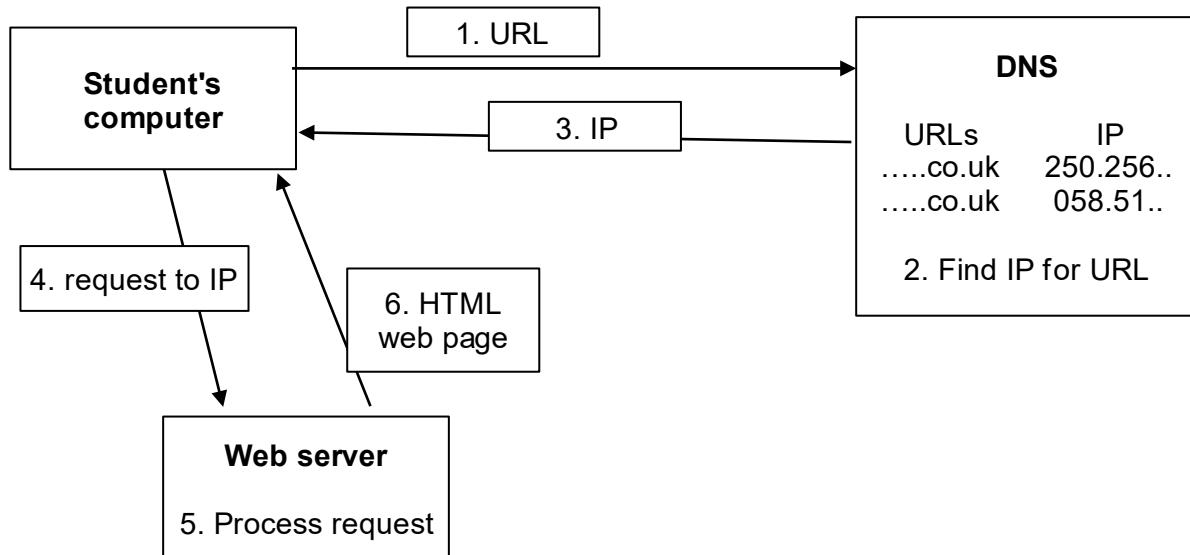
Note: No marks are awarded for using brand names of software packages or hardware.

Question	Answer	Marks
1(a)	B	1
1(b)(i)	A	1
1(b)(ii)	01001110	1
1(b)(iii)	<ul style="list-style-type: none"> • Unique binary/denary number given/stored for each character • The code for R is stored, then the code for E then D in sequence 	2
1(c)(i)	Any two from: <ul style="list-style-type: none"> • More bits allocated to each amplitude • Amplitudes can be more precise • A wider range of amplitudes can be recorded 	2
1(c)(ii)	Increase the sample rate	1

Question	Answer	Marks
2(a)	<p>One mark for letter and One mark for matching correction.</p> <ul style="list-style-type: none"> • Statement B ... • ...MAR stores addresses and not instructions • Statement C ... • ...Data is from bus not PC // Data is from address in MAR not PC 	4
2(b)(i)	It can run 3.5 billion FE cycles each second // it can execute 3.5 billion instructions each second	1
2(b)(ii)	<p>Any two from:</p> <ul style="list-style-type: none"> • More cores increases/improves the performance • More cores mean more FDE cycles/instructions are executed each second • ...because each core runs an FE cycle/instruction simultaneously 	2
2(b)(iii)	<p>Any two from:</p> <ul style="list-style-type: none"> • More cache improves performance • ...because more cache means the processor can access more frequently used data/instructions faster • ...instead of having to access the data from the slower-access RAM 	2
2(c)(i)	<p>Any two from:</p> <ul style="list-style-type: none"> • Volatile storage • Stores data for the processor to access directly/quickly // directly accessed by the CPU • Stores currently running data/instructions 	2
2(c)(ii)	<p>Any one from: e.g.</p> <ul style="list-style-type: none"> • BIOS // bootstrap/loader • Firmware • Parts of OS 	1
2(c)(iii)	<p>Any one from: e.g.</p> <ul style="list-style-type: none"> • To run programs when there is insufficient RAM to run them • To allow RAM to store more data when required 	1

Question	Answer		Marks							
3(a)	<p>One mark each:</p> <table border="1" data-bbox="332 269 1544 746"> <thead> <tr> <th data-bbox="332 269 669 333">Function name</th><th data-bbox="669 269 1544 333">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="332 333 669 476">managing memory</td><td data-bbox="669 333 1544 476"> Examples: • allocates memory to processes • prevents two processes accessing the same memory </td></tr> <tr> <td data-bbox="332 476 669 555">platform for running applications</td><td data-bbox="669 476 1544 555">allows application software to run on the computer</td></tr> <tr> <td data-bbox="332 555 669 746">managing peripherals</td><td data-bbox="669 555 1544 746"> Examples: • allocates data to buffers • transmits data to hardware • receives data from hardware </td></tr> </tbody> </table>	Function name	Description	managing memory	Examples: • allocates memory to processes • prevents two processes accessing the same memory	platform for running applications	allows application software to run on the computer	managing peripherals	Examples: • allocates data to buffers • transmits data to hardware • receives data from hardware	3
Function name	Description									
managing memory	Examples: • allocates memory to processes • prevents two processes accessing the same memory									
platform for running applications	allows application software to run on the computer									
managing peripherals	Examples: • allocates data to buffers • transmits data to hardware • receives data from hardware									
3(b)(i)	To indicate that something requires the attention of the processor/OS/CPU	1								
3(b)(ii)	<p>One mark for input device and matching interrupt: e.g.</p> <ul style="list-style-type: none"> <li data-bbox="332 905 669 952">• Keyboard Key pressed <li data-bbox="332 952 669 1032">• Mouse Mouse moved//button clicked 	1								

Question	Answer	Marks
3(b)(iii)	<p>Any five from:</p> <ul style="list-style-type: none"> • Interrupt is given priority • ...and placed in interrupt queue • Processor finishes current FE cycle for program • Processor checks interrupt priority queue // processor checks for higher priority interrupt than program/process • If lower priority processor runs next FDE cycle for program/process // if lower priority processor continues with program/process • (if higher priority) processor stores current process/registers on stack • Checks source of interrupt • ... and calls the appropriate ISR • ISR handles/resolves interrupt • If there is another higher priority interrupt (than process) then repeat • ... (otherwise) processor retrieves content of stack/registers/previous process (to continue with process from program) 	5

Question	Answer	Marks
4	<p>Any five marks for each part of diagram:</p> <ul style="list-style-type: none"> • URL from computer/browser to DNS • DNS storing table/database of URL and IPs // DNS finding IP for URL • DNS sending IP to computer / browser • DNS sending to higher DNS if not found • Web browser/computer sending request to IP of web server • Web server processing request • Web page data sent from server to computer / web browser  <pre> graph LR A[Student's computer] -- "1. URL" --> B["DNS
URLs
.....co.uk 250.256..
.....co.uk 058.51..
2. Find IP for URL"] B -- "3. IP" --> C[Web server
5. Process request] C -- "4. request to IP" --> D[6. HTML web page] </pre> <p>The diagram illustrates the process of retrieving a web page. It starts with a 'Student's computer' sending a 'URL' (step 1) to a 'DNS' server. The 'DNS' server contains a table of URLs and their corresponding IP addresses. It finds the IP address for the given URL (step 2) and returns it to the 'Student's computer' (step 3). The 'Student's computer' then sends a 'request to IP' (step 4) to a 'Web server'. The 'Web server' processes the request (step 5) and returns an 'HTML web page' (step 6) to the 'Student's computer'.</p>	5

Question	Answer	Marks
5	<p>One mark for each term in correct place:</p> <ul style="list-style-type: none"> • physically • blockchains • time-stamp • traced <p>A digital currency does not exist physically, it can only be accessed electronically.</p> <p>Some digital currencies have digital ledgers called a blockchains. These are decentralised databases where each transaction is stored as a new set of data with a time-stamp and is linked to the previous set of data. This means that transactions cannot be altered, only new transactions added, which allows the location of the data to be traced.</p>	4

Question	Answer	Marks
6(a)	<p>Any one from:</p> <ul style="list-style-type: none"> • It has electrical components // by example • It is programmable 	1
6(b)(i)	<p>Any five from:</p> <ul style="list-style-type: none"> • Sensor continuously sends the digitised value / reading / distance to the microprocessor • Microprocessor compares the data / signal to the stored data of a person and distance of 3m • If the data / signal is less than (or equal to) a person and within 3m • ...a signal is sent to actuator to make the tractor stop / apply the brakes • If the data/signal is greater than 3m no action is taken • If stopped and data/signal is not a person and/or more than 3m a signal is sent to actuator to make the tractor start • The whole process repeats until turned off 	5

Question	Answer	Marks
6(b)(ii)	<p>One mark for sensor and One mark for matching use: e.g.</p> <ul style="list-style-type: none"> • Accelerometer ... • ...to adjust for uneven ground // to detect if the tractor crashes • Proximity ... • ...to detect if near the end of the field // to detect other obstacles • Light ... • ... to identify when to turn the headlights on 	2
6(c)	<p>Any three from: e.g.</p> <ul style="list-style-type: none"> • Set-up cost may be high • Maintenance cost may be high • ... needs skilled workers/expert to fix • Farmer may need reskilling in how to use it ... • ... which could be costly • Leads to deskilling of farmers/workers • Farmer may need fewer employees • ... leading to unemployment • Can malfunction • ... and not recognise a person and fails to stop • Changing function can be expensive 	3
6(d)	<p>Any three from</p> <ul style="list-style-type: none"> • Knowledge base • Rule base • Inference engine • Interface 	3

Question	Answer	Marks																																																																								
6(e)(i)	<p>Any three from:</p> <ul style="list-style-type: none"> • The same data is transmitted back (from computer to tractor) • Tractor compares both sets of data • ... if they are identical there is no error // reverse • Tractor sends confirmation of accuracy if the same // Tractor resends the data if they are different // Tractor transmits error if they are different 	3																																																																								
6(e)(ii)	<p>One mark for two correct Two marks for five correct Three marks for all eight correct</p> <table border="1" data-bbox="332 589 1477 1240"> <thead> <tr> <th data-bbox="332 589 466 695">parity bit</th><th data-bbox="466 589 624 695">bit 7</th><th data-bbox="624 589 781 695">bit 6</th><th data-bbox="781 589 938 695">bit 5</th><th data-bbox="938 589 1095 695">bit 4</th><th data-bbox="1095 589 1252 695">bit 3</th><th data-bbox="1252 589 1409 695">bit 2</th><th data-bbox="1409 589 1477 695">bit 1</th></tr> </thead> <tbody> <tr> <td data-bbox="332 695 466 747">byte 1</td><td data-bbox="466 695 624 747">1</td><td data-bbox="624 695 781 747">1</td><td data-bbox="781 695 938 747">0</td><td data-bbox="938 695 1095 747">0</td><td data-bbox="1095 695 1252 747">1</td><td data-bbox="1252 695 1409 747">1</td><td data-bbox="1409 695 1477 747">0</td></tr> <tr> <td data-bbox="332 747 466 800">byte 2</td><td data-bbox="466 747 624 800">1</td><td data-bbox="624 747 781 800">0</td><td data-bbox="781 747 938 800">0</td><td data-bbox="938 747 1095 800">0</td><td data-bbox="1095 747 1252 800">0</td><td data-bbox="1252 747 1409 800">1</td><td data-bbox="1409 747 1477 800">1</td></tr> <tr> <td data-bbox="332 800 466 852">byte 3</td><td data-bbox="466 800 624 852">0</td><td data-bbox="624 800 781 852">1</td><td data-bbox="781 800 938 852">0</td><td data-bbox="938 800 1095 852">0</td><td data-bbox="1095 800 1252 852">0</td><td data-bbox="1252 800 1409 852">0</td><td data-bbox="1409 800 1477 852">0</td></tr> <tr> <td data-bbox="332 852 466 905">byte 4</td><td data-bbox="466 852 624 905">0</td><td data-bbox="624 852 781 905">1</td><td data-bbox="781 852 938 905">0</td><td data-bbox="938 852 1095 905">0</td><td data-bbox="1095 852 1252 905">1</td><td data-bbox="1252 852 1409 905">1</td><td data-bbox="1409 852 1477 905">1</td></tr> <tr> <td data-bbox="332 905 466 957">byte 5</td><td data-bbox="466 905 624 957">1</td><td data-bbox="624 905 781 957">0</td><td data-bbox="781 905 938 957">0</td><td data-bbox="938 905 1095 957">0</td><td data-bbox="1095 905 1252 957">0</td><td data-bbox="1252 905 1409 957">0</td><td data-bbox="1409 905 1477 957">0</td></tr> <tr> <td data-bbox="332 957 466 1009">byte 6</td><td data-bbox="466 957 624 1009">0</td><td data-bbox="624 957 781 1009">1</td><td data-bbox="781 957 938 1009">1</td><td data-bbox="938 957 1095 1009">1</td><td data-bbox="1095 957 1252 1009">1</td><td data-bbox="1252 957 1409 1009">1</td><td data-bbox="1409 957 1477 1009">1</td></tr> <tr> <td data-bbox="332 1009 466 1062">byte 7</td><td data-bbox="466 1009 624 1062">1</td><td data-bbox="624 1009 781 1062">1</td><td data-bbox="781 1009 938 1062">0</td><td data-bbox="938 1009 1095 1062">0</td><td data-bbox="1095 1009 1252 1062">1</td><td data-bbox="1252 1009 1409 1062">1</td><td data-bbox="1409 1009 1477 1062">0</td></tr> <tr> <td data-bbox="332 1062 466 1240">parity byte</td><td data-bbox="466 1062 624 1240">1</td><td data-bbox="624 1062 781 1240">0</td><td data-bbox="781 1062 938 1240">0</td><td data-bbox="938 1062 1095 1240">0</td><td data-bbox="1095 1062 1252 1240">1</td><td data-bbox="1252 1062 1409 1240">0</td><td data-bbox="1409 1062 1477 1240">1</td></tr> </tbody> </table>	parity bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	byte 1	1	1	0	0	1	1	0	byte 2	1	0	0	0	0	1	1	byte 3	0	1	0	0	0	0	0	byte 4	0	1	0	0	1	1	1	byte 5	1	0	0	0	0	0	0	byte 6	0	1	1	1	1	1	1	byte 7	1	1	0	0	1	1	0	parity byte	1	0	0	0	1	0	1	3
parity bit	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1																																																																			
byte 1	1	1	0	0	1	1	0																																																																			
byte 2	1	0	0	0	0	1	1																																																																			
byte 3	0	1	0	0	0	0	0																																																																			
byte 4	0	1	0	0	1	1	1																																																																			
byte 5	1	0	0	0	0	0	0																																																																			
byte 6	0	1	1	1	1	1	1																																																																			
byte 7	1	1	0	0	1	1	0																																																																			
parity byte	1	0	0	0	1	0	1																																																																			

Question	Answer	Marks
7(a)(i)	If the data is intercepted, it cannot be understood	1
7(a)(ii)	Four from: <ul style="list-style-type: none"> • Symmetric has a shared key... • ... to encrypt and decrypt • Both the sender and receiver know the key • Asymmetric has different keys // a public key and a private key • ...public to encrypt the data and private to decrypt • ...anyone can know the public key but only those intended know the private key 	4
7(b)(i)	Any two from: e.g. <ul style="list-style-type: none"> • Destination address/IP • Sender address/IP • Packet number 	2
7(b)(ii)	Any one from: <ul style="list-style-type: none"> • Control the route the packet takes • Send each packet towards its destination • Choose more efficient route 	1

Question	Answer	Marks
8(a)(i)	EC	1
8(a)(ii)	Any one from: <ul style="list-style-type: none"> • Easier for humans to read/remember • Shorter for humans to enter • Less likely for humans to make mistakes • Easier for humans to spot errors/debug • Takes up less space onscreen 	1

Question	Answer	Marks
8(b)(i)	One mark for working, one mark for answer <ul style="list-style-type: none">• e.g. showing flip and add 1 $10110111 = 01001001$• -73	2
8(b)(ii)	00101101	1
8(c)	One mark each <ul style="list-style-type: none">• Divide• ...by $16 / 2^4$	2