Cambridge International AS & A Level

Cambridge Assessment International Education
Cambridge International Advanced Subsidiary and Advanced Level

COMPUTER SCIENCE **9608/12**

Paper 1 Written Paper **October/November 2018**

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 2018 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.

Question	Answer	Marks
1(a)(i)	2	**1**
1(a)(ii)	**1 mark** per bullet point • Number of pixels: 6 × 6 // 36 • Number of bits: Number of pixels (36) × 2 … • … = 72 bits // 9 bytes	**3**
1(b)(i)	**1 mark** per bullet point • Number of pixels: 1000 × 1000 // 1 000 000 • Number of bytes: Number of pixels (1 000 000) × 2 // 2 000 000 // Number of bits: Number of pixels (1 000 000) × 16 // 16 000 000 • Conversion to megabytes • 2 (MB) // 1.91 (MB)	**4**
1(b)(ii)	**1 mark** per method correctly linked to its description **max 3** **1 mark** for each compression type correctly linked to its method(s). **max 2** **Description** / **Method** / **Compression type** Removes pixels Reduces number of pixels per inch Uses fewer bits per pixel Stores colour code and count of repetitions Crop the photograph Use run-length encoding Use fewer colours Lossy Lossless	**5**
1(c)	**1 mark** per bullet point. **Max 2** marks for each reason. • Smaller file size • Can be transferred quicker/downloaded quicker • Enlarges without pixilation • Needs to be used on different screens / devices / resolutions	**4**

Question	Answer	Marks
2(a)	**1 mark** for **1 correct** answer, **2 marks** for all **3 correct** answers 1 Gopal types into the web browser **2** **B** (Web browser sends URL to Domain name Service (DNS)) 3 DNS looks up URL in a table **4** **A** (DNS finds corresponding IP address) **5** **C** (DNS returns IP address to web browser)	**2**

Question	Answer	Marks
2(b)	**1 mark** per bullet point to **max 2** • Gives each device on a network an identifier // IP address used to locate a device on a network • Each address is unique within the network • Allows a device/gateway/node to send data to the correct destination / a specific device/gateway/node	**2**
2(c)(i)	**1 mark** per bullet point to **max 3** • Less interference in signal • Signal does not degrade as fast // Needs less signal boosting • More difficult to hack // more secure • Greater bandwidth // Faster transmission speeds possible	**3**
2(c)(ii)	**1 mark** per bullet point to **max 2** • (Initial) installation cost is higher // Cable / hardware is more expensive to buy (per metre) • Specialists / trained personnel needed to install / maintain • Difficult to terminate // Electronics at both ends are more complex • Fibre-optic cables can break when bent • Only transmits data in one direction • If a fibre-optic cable connection fails, many more services can be affected	**2**

Question	Answer	Marks
3(a)(i)	**1 mark** per bullet point Absolute addressing: • The operand is a numeric address // The numeric address is given // referring directly to a memory location Symbolic addressing: • The operand is a word/symbol // A word/symbol represents the memory location/address	**2**
3(a)(ii)	**1 mark** per example Absolute addressing: For example, `ADD 230` Symbolic addressing: For example, `ADD num1`	**2**
3(b)(i)	**1 mark** per bullet point Indexed addressing: • The address to be used is formed by: operand + the contents of the Index Register (IX) Immediate addressing: • The operand is not an address // the operand is the actual value to be loaded	**2**

Question	Answer	Marks
3(b)(ii)	**1 mark** per example Indexed: For example, `LDX 20` Immediate: For example, `ADD #20`	**2**
3(c)(i)	193	**1**
3(c)(ii)	C1	**1**
3(c)(iii)	–63	**1**
3(d)	**1 mark** per bullet point • Loading 2, comparing with 104 (instructions 40 and 41) • Loading 302 (instruction 43) • Comparing, and branching to 47 (instructions 44, 45) • Loading, decrementing accumulator and storing (instructions 47, 48 and 49) • Incrementing Index Register (instruction 50) • Loading 303, comparing and outputting + (instructions 43–46) • Loading, decrementing accumulator and storing, incrementing Index Register and end (instructions 47–51, 41, 42 and 54)	**7**

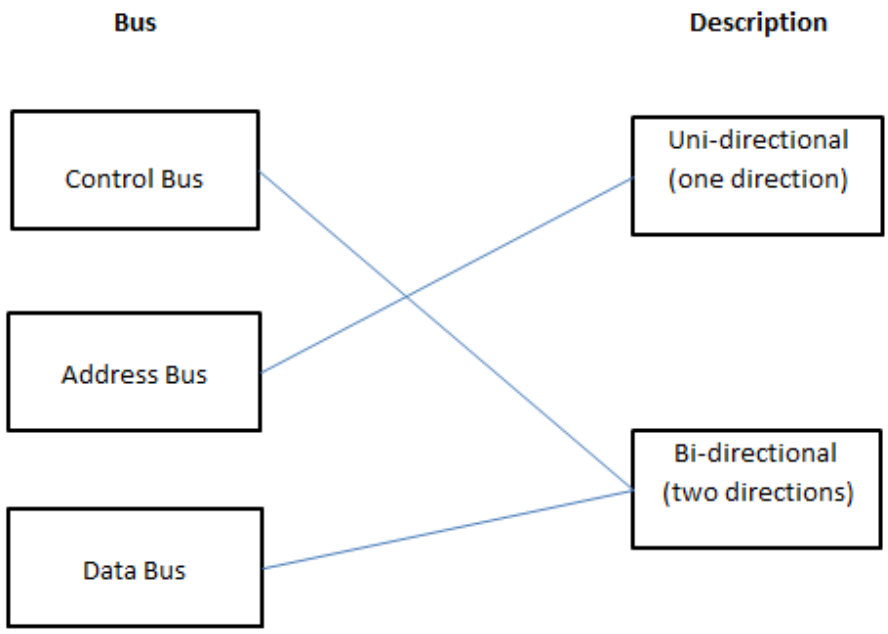
Question	Answer	Marks

Instruction address	ACC	Memory address						IX	OUTPUT
		100	101	102	103	104	105		
		2	302	303	303	0	303	1	
40	2								
41									
43	302								
44									
45									
47	2								
48	1								
49		1							
50								2	
51									
41									
42									
43	303								
44									
45									
46									+
47	1								
48	0								
49		0							
50								3	
51									
41									
54									

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>4(a)</td><td>1 mark per correct line, max 3
<table>
<tr><th>Line number of error</th><th>Correct notation</th></tr>
<tr><td>1</td><td>MAR ← [PC]</td></tr>
<tr><td>3</td><td>MDR ← [[MAR]]</td></tr>
<tr><td>4</td><td>CIR ← [MDR]</td></tr>
<tr><td>2</td><td>PC ← [PC] + 1</td></tr>
</table></td><td>3</td></tr>
<tr><td>4(b)(i)</td><td>1 mark for each event to max 3

For example:
• Hardware fault // Example of hardware fault
• I/O request // Example of I/O request
• Program/software error // Example of software error
• End of a time-slice</td><td>3</td></tr>
<tr><td>4(b)(ii)</td><td>1 mark per bullet point to max 5

• At the <u>end</u> of each fetch–execute cycle the processor checks for interrupt(s)
• Check if an interrupt flag is set // Check if bit set in interrupt register
• Processor identifies source of interrupt
• Processor checks priority of interrupt
• If interrupt priority is high enough // Lower priority interrupts are disabled
• Processor saves current contents of registers
• Processor calls interrupt handler / Interrupt Service Routine (ISR)
• Address of ISR is loaded into Program Counter (PC)
• When servicing of interrupt complete, processor restores registers
• Lower priority interrupts are re-enabled
• Processor continues with next F–E cycle</td><td>5</td></tr>
</table>

Question	Answer	Marks
4(c)	**1 mark** for **1 correct** connection **2 marks** for all **3 correct** connections 	**2**

Question	Answer	Marks
5(a)	**1 mark** per bullet point for each justification, to **max 2** **Either** Unethical • Noah's work may be confidential • Wendy shouldn't claim someone else's ideas / work as her own • She is bringing the profession into disrepute • Reference to IEEE standards in context **Or** Ethical • The code could be open source • Wendy may have permission from Noah • Wendy isn't copying the code, just getting ideas • Reference to IEEE standards in context	**2**

Question	Answer	Marks
5(b)	**1 mark** per bullet point for each justification, to **max 2** **Either** Unethical • Amit has a responsibility to his company • He should have taken it to the police rather than putting it on the Internet • He has a signed agreement to say he will not give anything away • Reference to IEEE standards in context **Or** Ethical • Amit is acting in the public interest • Amit may not have actually signed the confidentiality agreement • If acting illegally, the multinational company should be brought to justice • Reference to IEEE standards in context	**2**
5(c)	**1 mark** per bullet point for each justification, to **max 2** **Either** Ethical • It might save people's jobs • Farah is acting in the best interest of her company • Reference to IEEE standards in context **Or** Unethical • Farah has a responsibility to act in the best interest of her client • It could give her company a bad reputation • Reference to IEEE standards in context	**2**

Question	Answer	Marks
6(a)	**1 mark** for each method to **max 2** • Biometric authentication // by example • Two-step authentication // by example • Firewall / proxy • Encryption • Different access rights for different users • Password protect file (using a different password) • Anti-malware	**2**
6(b)(i)	**1 mark** per bullet to **max 2** • Pre-existing / pre-compiled / pre-written modules / code … • … can be linked into her program (without amendment) • To perform common / complex tasks	**2**

Question	Answer	Marks
6(b)(ii)	**1 mark** per bullet point. **Max 2** for <u>one benefit</u>, **max 2** for <u>one drawback</u> Benefit: • Less code needs to be written • … saves time / saves re-inventing the wheel • Pre-tested // Used by many people • ... reduces time testing // can be fairly sure that the function will perform as it should • Can be written in a different programming language • … making use of special features of that language • Can be complex algorithms (e.g. mathematical/graphics functions) • ... she does not need to work out how to write it // …that she may not know how to code • Simplifies the program • … since just the name of the function included in the source code Drawback: • Compatibility issues • … may not work with the other code/may require changing program for it to work • Not guaranteed thorough testing • … may be unknown or unexpected bugs / virus • Library routine may not meet exact needs • … may give unexpected results // … may need editing • If library routine is changed • … there may be unexpected results / errors	**4**

Question	Answer	Marks
6(c)	**1 mark** per bullet point. **Max 3** marks for interpreter, **max 3** marks for compiler Interpreter: • Used during development • Debugging is easier • … Because errors are reported as they are found // No need to wait until the end of the process for the error report • … Because errors can be corrected as they are found Compiler: • Compiler used when development complete // compiler used when program ready for distribution • Produces an executable file (.exe) • … After compilation the compiler does not need to be present for the program to run • … The program can be given to others without access to (source) code • … Final program does not need to be re-compiled each time it is run • Cross-compilation, the program can be compiled to run on different platforms	**4**