Cambridge International AS & A Level

Cambridge Assessment International Education
Cambridge International Advanced Subsidiary and Advanced Level

COMPUTER SCIENCE **9608/13**

Paper 1 Written Paper **October/November 2019**

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 2019 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)(i)</td><td>1 mark per input device to max 2

e.g.
• Barcode scanner / Infra-red scanner
• Pressure sensor
• RFID / chip reader
• Bank note scanner
• Pin / key pad
• Magnetic strip reader</td><td>2</td></tr>
<tr><td>1(a)(ii)</td><td>1 mark per output device to max 2

e.g.
• Speaker
• Printer
• LCD screen</td><td>2</td></tr>
<tr><td>1(a)(iii)</td><td>1 mark for at least two statements in the correct position, 2 marks for all four statements in correct position.

1 B …. ….. (The screen has a layer that stores an electrical charge)
2 When the user touches the screen
3 A …… (Charge is drawn to the point of contact)
4 C …… (There is a change in the electrostatic field)
5 The coordinates of the point of contact can be calculated
6 D ……. (These coordinates are sent to the touchscreen driver)</td><td>2</td></tr>
<tr><td>1(b)(i)</td><td>1 mark per bullet point to max 2

• To store the files needed to boot the system
• To store parts of the self- checkout machine operating system
• To store the self-checkout machine software
• To store the intermediate data / running total for items purchased</td><td>2</td></tr>
<tr><td>1(b)(ii)</td><td>1 mark for at least one correct row, 2 marks for all three correct rows

<table>
<tr><th>Statement</th><th>SRAM</th><th>DRAM</th></tr>
<tr><td>More expensive to make</td><td>✓</td><td></td></tr>
<tr><td>Requires refreshing (recharging)</td><td></td><td>✓</td></tr>
<tr><td>Made from flip-flops</td><td>✓</td><td></td></tr>
</table>
</td><td>2</td></tr>
</table>

Question	Answer	Marks
1(c)(i)	**1 mark** per bullet point to **max 3** plus **1 mark** for suitable example • When a barcode on an item is scanned • … the server performs any requested tasks // the server looks up the details of the product • The self-checkout machine is a client • … that send requests to the server // the self-checkout machine asks for, e.g. the price of the item • The server returns the results of the request // the server returns e.g. the item price • Self-checkout machine displays e.g. price to the user	**4**
1(c)(ii)	**1 mark** for each security method to **max 2**, **1 mark** for integrity **Security** • encryption • access rights • username and password // biometrics // user accounts • backup // disk mirroring • firewall • Physical methods (e.g. CCTV, locked rooms etc.) **Integrity** • checksum • parity • validation on input	**3**

Question	Answer	Marks
2(a)	**1 mark** per bullet point to **max 4** for each management task, **max 6** in total **Process Management** • Manages the **scheduling** of processes • … allows multi-tasking / multi-processing • … ensures fair access • … handles priorities • Manages which resources the processes require • Enables processes to share information • Prevents interference between processes // resolution of conflicts **Memory Management** • Allocates memory to processes • Ensures fair usage of memory • Organises memory / by example • Makes use of virtual memory • Keep processes separate • To release memory when a process stops	**6**

Question	Answer	Marks
2(b)(i)	**1 mark** per bullet point to **max 2** • The amplitude of the wave is measured • … at set, regular time intervals • The value is stored as a binary number	**2**
2(b)(ii)	**1 mark** per bullet point • Sampling resolution of 44100 Hz takes more samples per second, so the file size will be larger // Sampling resolution of 21000 Hz takes fewer samples per second, so the file size will be smaller • At a resolution of 44100 Hz, the sound recording is a closer / more accurate representation of Leonardo's voice // At a resolution of 21000 Hz, the sound recording is a less accurate representation of Leonardo's voice	**2**
2(b)(iii)	**1 mark** for naming a feature, **1 mark** for description, **max 2** marks for each feature e.g. • Amplify • ... Increase the volume of a section of sound • Change pitch • ... Increase/decrease frequency of section(s) • Change sampling resolution • … to change the accuracy of the sound / file size	**4**

Question	Answer	Marks

3(a)(i)	**1 mark** per table
	
	• CUSTOMER table has at least customer ID, customer name, address and contact details
	• ROOM has at least room number, room type,
	• BOOKING has at least booking ID, room number, customer ID, start date, number of nights
	
	`CUSTOMER (CustomerID, Name, Address, ContactDetails)`
	
	`ROOM (RoomNumber, RoomType)`
	
	`BOOKING (BookingID, RoomNumber, CustomerID, StartDate, NumberNights)`
		3

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(a)(ii)</td><td>1 mark for 1 or 2 correct Primary Keys, 2 marks for 3 correct Primary Keys

<code>CUSTOMER: CustomerID</code>

<code>ROOM: RoomNumber</code>

<code>BOOKING: BookingID</code></td><td>2</td></tr>
<tr><td>3(a)(iii)</td><td>1 mark for both table name and Foreign Key

Table: <code>BOOKING</code>
Foreign Key: <code>CustomerID / RoomNumber</code></td><td>1</td></tr>
<tr><td>3(b)</td><td>1 mark per bullet point to max 2 plus 1 mark for suitable example for each DBMS tool

Developer Interface
• To create user friendly features e.g. forms to enter new bookings
• To create outputs e.g. report of bookings on a given date
• To create interactive features e.g. buttons and menus

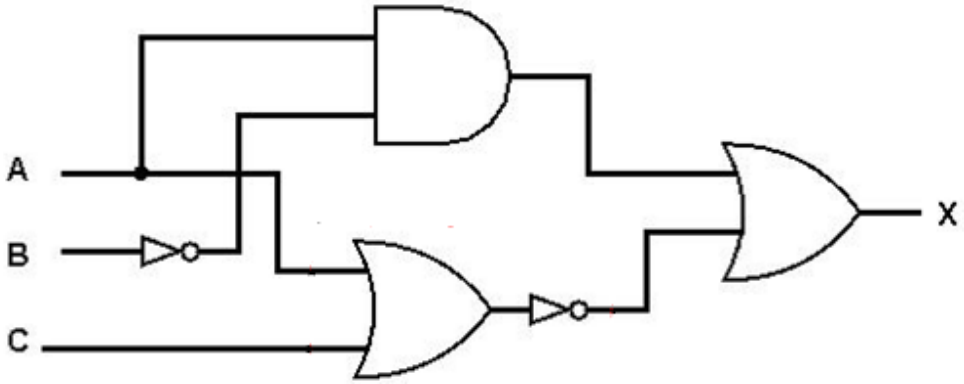
Query processor
• To create SQL/QBE queries
• To search for data that meets set criteria, e.g. all bookings for next week
• To perform calculations on extracted data, e.g. number of empty rooms tomorrow</td><td>5</td></tr>
<tr><td>3c</td><td>1 mark for at least two correct rows, 2 marks for all four correct rows

<table>
<tr><th>Script</th><th>DDL</th><th>DML</th></tr>
<tr><td><code>CREATE TABLE FILMS</code></td><td>✓</td><td></td></tr>
<tr><td><code>SELECT FilmID FROM FILMS</code></td><td></td><td>✓</td></tr>
<tr><td><code>ALTER TABLE FILMS ADD PRIMARY KEY (FilmID)</code></td><td>✓</td><td></td></tr>
<tr><td><code>CREATE DATABASE MYDATA</code></td><td>✓</td><td></td></tr>
</table></td><td>2</td></tr>
</table>

Question	Answer	Marks
4(a)	1532	**1**
4(b)	1111 0001 0001	**1**
4(c)	101	**1**
4(d)	65	**1**
4(e)	DE	**1**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>5(a)</td><td>1 mark for each correct gate

• A OR C
• NOT(A OR C)
• NOT B
• A AND NOT B
• Final OR

 
</td><td>5</td></tr>
<tr><td>5(b)</td><td>1 mark for each pair of rows

<table>
<tr><th>A</th><th>B</th><th>C</th><th>Working space</th><th>X</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td></td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td></td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td></td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td></td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td></td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td></td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td></td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td></td><td>0</td></tr>
</table></td><td>4</td></tr>
</table>

Question	Answer	Marks
6(a)(i)	**1 mark** per bullet point to **max 3** • `checkAnswer` • `Math.floor // floor` • `Math.random // random` • `getElementById` • `toString` • `alert`	**3**
6(a)(ii)	**1 mark** per bullet point • 12 • 17 and 19	**2**
6(a)(iii)	14	**1**
6(a)(iv)	**1 mark** per bullet point to **max 2** • Converts the number in the variable `answer` to a string • Compares the value in `userAnswer` with the string value of answer • The `if` determines which line is executed next	**2**
6(b)	**1 mark** per bullet point to **max 4** • Using program libraries saves time • … as she does not have to write some routines • Program library routines should already be thoroughly tested • … so, Willow can use them without having to spend time checking they work / correcting errors • Program library routines may perform complex tasks • ... so, Willow won't have to work out how to code them	**4**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>7</td><td>1 mark for each correct addressing mode
<table>
<tr><th>Addressing mode</th><th>Description</th></tr>
<tr><td>Relative</td><td>Form the address by adding the given number to a base address. Load the contents of the calculated address to the Accumulator (ACC).</td></tr>
<tr><td>Indirect</td><td>Load the contents of the address held at the given address to ACC.</td></tr>
<tr><td>Direct</td><td>Load the contents of the given address to ACC.</td></tr>
<tr><td>Indexed</td><td>Form the address from the given address + the contents of the Index Register. Load the contents of the calculated address to ACC.</td></tr>
<tr><td>Immediate</td><td>Load the given value directly to ACC.</td></tr>
</table></td><td>5</td></tr>
</table>