

---

**COMPUTER SCIENCE**

**9608/22**

Paper 2 Written Paper

**May/June 2017**

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 will not enter into discussions about these mark schemes.

Cambridge is publishing the mark schemes for the May/June 2017 series for most Cambridge IGCSE<sup>®</sup>, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

Question	Answer					Marks																									
1(a)	<table border="1" data-bbox="252 248 1382 546"> <thead> <tr> <th data-bbox="256 255 352 300">Item</th> <th data-bbox="357 255 1043 300">Statement</th> <th data-bbox="1048 255 1131 300">Input</th> <th data-bbox="1136 255 1259 300">Process</th> <th data-bbox="1264 255 1377 300">Output</th> </tr> </thead> <tbody> <tr> <td data-bbox="256 306 352 351">1</td> <td data-bbox="357 306 1043 351">SomeChars = "Hello World"</td> <td data-bbox="1048 306 1131 351"></td> <td data-bbox="1136 306 1259 351">✓</td> <td data-bbox="1264 306 1377 351"></td> </tr> <tr> <td data-bbox="256 358 352 403">2</td> <td data-bbox="357 358 1043 403">OUTPUT RIGHT(String1,5)</td> <td data-bbox="1048 358 1131 403"></td> <td data-bbox="1136 358 1259 403">✓</td> <td data-bbox="1264 358 1377 403">✓</td> </tr> <tr> <td data-bbox="256 409 352 454">3</td> <td data-bbox="357 409 1043 454">READFILE (MyFile, String2)</td> <td data-bbox="1048 409 1131 454">✓</td> <td data-bbox="1136 409 1259 454"></td> <td data-bbox="1264 409 1377 454"></td> </tr> <tr> <td data-bbox="256 461 352 506">4</td> <td data-bbox="357 461 1043 506">WRITEFILE (MyFile, "Data is " &amp; String2)</td> <td data-bbox="1048 461 1131 506"></td> <td data-bbox="1136 461 1259 506">✓</td> <td data-bbox="1264 461 1377 506">✓</td> </tr> </tbody> </table> <p data-bbox="252 577 464 613">Mark as follows:</p> <p data-bbox="252 645 472 680">Row 1 as shown</p> <p data-bbox="252 680 1094 716">Row 2 no marks if tick in Input column, otherwise 1 mark per tick</p> <p data-bbox="252 716 472 752">Row 3 as shown</p> <p data-bbox="252 752 1094 788">Row 4 no marks if tick in Input column, otherwise 1 mark per tick</p>					Item	Statement	Input	Process	Output	1	SomeChars = "Hello World"		✓		2	OUTPUT RIGHT(String1,5)		✓	✓	3	READFILE (MyFile, String2)	✓			4	WRITEFILE (MyFile, "Data is " & String2)		✓	✓	6
Item	Statement	Input	Process	Output																											
1	SomeChars = "Hello World"		✓																												
2	OUTPUT RIGHT(String1,5)		✓	✓																											
3	READFILE (MyFile, String2)	✓																													
4	WRITEFILE (MyFile, "Data is " & String2)		✓	✓																											
1(b)(i)	<ul data-bbox="252 815 1010 884" style="list-style-type: none"> <li>• Integer / Real / Single / Double / Floating Point / Float</li> <li>• Boolean</li> </ul>					2																									
1(b)(ii)	<table border="1" data-bbox="252 918 1091 1200"> <thead> <tr> <th data-bbox="256 925 807 969">Expression</th> <th data-bbox="812 925 1086 969">Evaluates to</th> </tr> </thead> <tbody> <tr> <td data-bbox="256 976 807 1043">(FlagA AND FlagB) OR FlagC</td> <td data-bbox="812 976 1086 1043"><b>TRUE</b></td> </tr> <tr> <td data-bbox="256 1050 807 1117">FlagA AND (FlagB OR FlagC)</td> <td data-bbox="812 1050 1086 1117"><b>TRUE</b></td> </tr> <tr> <td data-bbox="256 1124 807 1191">(NOT FlagA) OR (NOT FlagC)</td> <td data-bbox="812 1124 1086 1191"><b>FALSE</b></td> </tr> </tbody> </table> <p data-bbox="252 1232 501 1267">1 mark per answer</p>					Expression	Evaluates to	(FlagA AND FlagB) OR FlagC	<b>TRUE</b>	FlagA AND (FlagB OR FlagC)	<b>TRUE</b>	(NOT FlagA) OR (NOT FlagC)	<b>FALSE</b>	3																	
Expression	Evaluates to																														
(FlagA AND FlagB) OR FlagC	<b>TRUE</b>																														
FlagA AND (FlagB OR FlagC)	<b>TRUE</b>																														
(NOT FlagA) OR (NOT FlagC)	<b>FALSE</b>																														
1(c)	<pre data-bbox="252 1301 691 1507">MyCount ← 101  REPEAT   OUTPUT MyCount   MyCount ← MyCount + 2 UNTIL MyCount &gt; 199</pre> <p data-bbox="252 1538 671 1574">1 mark for each of the following:</p> <ul data-bbox="252 1606 994 1749" style="list-style-type: none"> <li>• Counter initialisation</li> <li>• Repeat ... Until loop</li> <li>• Method for choosing (correct range of) odd numbers</li> <li>• Output all odd numbers in the range</li> </ul> <p data-bbox="252 1780 887 1816">Note: Counter variable name must be consistent</p>					4																									

Question	Answer	Marks
2(a)	<ul style="list-style-type: none"> <li>• to <u>increase the level of detail of an algorithm / design...</u> // breaking down a <u>problem / module / task</u> into smaller parts...</li> <li>• ...from which the task may be <u>programmed</u></li> </ul> <p>1 mark per underlined phrase or equivalent</p>	<b>2</b>
2(b)	<p>1 mark for first 3 data types – String 1 mark for last data type – Boolean</p> <p>1 mark for each description:</p> <p>FileUserID                      Stores (User) ID from <u>file</u> FilePreferredName            Stores (preferred) name from <u>file</u> IDFoundFlag                   True if (User) ID found in <u>file</u> // False if (User) ID not found in <u>file</u>    // If SearchUserID matches FileUserID</p>	<b>5</b>
2(c)	<ol style="list-style-type: none"> <li>1. LOOP through the file until EOF()...</li> <li>2. OR SearchUserId is found</li> <li>3. READ text line from UserNames.txt file <b>in a loop</b></li> <li>4. EXTRACT FileUserID <b>in a loop</b></li> <li>5. IF SearchUserId matches FileUserID THEN <b>in a loop</b></li> <li>6. SET FilePreferredName to the name from the file</li> <li>7. Check if User ID found <b>not in a loop</b></li> <li>8. OUTPUT appropriate message for both conditions</li> </ol> <p>1 mark per functional equivalent of each numbered statement.</p>	<b>Max 8</b>

Question	Answer	Marks
3	<pre> FUNCTION ExCamel (<u>InString</u>: STRING) RETURNS <u>STRING</u>   DECLARE <u>NextChar</u> : <u>CHAR</u>   DECLARE <u>OutString</u> : STRING   DECLARE n : INTEGER    <u>OutString</u> ← "" // initialise the return string   // loop through InString to produce OutString   FOR n ← 1 TO <u>LENGTH(InString)</u> // from first to last     <u>NextChar</u> ← <u>MID(InString, n, 1)</u> // get next character     IF <u>NextChar</u> &gt;= 'A' AND <u>NextChar</u> &lt;= 'Z' // check if upper   case       // <u>NextChar</u> = UCASE(<u>NextChar</u>)       THEN         IF n &gt; 1 // if not first character           THEN             <u>OutString</u> ← <u>OutString &amp; " "</u> // add space   to OutString           ENDIF         <u>NextChar</u> ← <u>LCASE(NextChar)</u> // make NextChar lower   case       ENDIF     <u>OutString</u> ← <u>OutString &amp; NextChar</u> // add Nextchar to   OutString   ENDFOR   <u>RETURN OutString</u> // return value ENDFUNCTION </pre> <p>1 mark per underlined word / expression</p>	Max 11

Question	Answer	Marks									
4(a)	<ul style="list-style-type: none"> <li>• Functions</li> <li>• Procedures</li> <li>• Global / Local variables</li> </ul> <p>1 mark per item</p>	<b>Max 2</b>									
4(b)	<table border="1" data-bbox="292 454 1342 779"> <thead> <tr> <th data-bbox="292 454 580 539">Name of parameter passing method</th> <th data-bbox="580 454 711 539">Value output</th> <th data-bbox="711 454 1342 539">Explanation</th> </tr> </thead> <tbody> <tr> <td data-bbox="292 539 580 658">(Call) by reference</td> <td data-bbox="580 539 711 658">5</td> <td data-bbox="711 539 1342 658"> <ul style="list-style-type: none"> <li>• The <u>address of the variable</u> is passed.</li> <li>• <u>Original value is changed</u> when parameter changed in called module.</li> </ul> </td> </tr> <tr> <td data-bbox="292 658 580 779">(Call) by value</td> <td data-bbox="580 658 711 779">4</td> <td data-bbox="711 658 1342 779"> <ul style="list-style-type: none"> <li>• A <u>copy of the variable</u> itself is passed.</li> <li>• <u>Original value not changed</u> when parameter changed in called module.</li> </ul> </td> </tr> </tbody> </table> <p>Mark as follows:</p> <ul style="list-style-type: none"> <li>• 1 mark for each name <b>and</b> value</li> <li>• 1 mark per bullet in explanation</li> </ul>	Name of parameter passing method	Value output	Explanation	(Call) by reference	5	<ul style="list-style-type: none"> <li>• The <u>address of the variable</u> is passed.</li> <li>• <u>Original value is changed</u> when parameter changed in called module.</li> </ul>	(Call) by value	4	<ul style="list-style-type: none"> <li>• A <u>copy of the variable</u> itself is passed.</li> <li>• <u>Original value not changed</u> when parameter changed in called module.</li> </ul>	<b>6</b>
Name of parameter passing method	Value output	Explanation									
(Call) by reference	5	<ul style="list-style-type: none"> <li>• The <u>address of the variable</u> is passed.</li> <li>• <u>Original value is changed</u> when parameter changed in called module.</li> </ul>									
(Call) by value	4	<ul style="list-style-type: none"> <li>• A <u>copy of the variable</u> itself is passed.</li> <li>• <u>Original value not changed</u> when parameter changed in called module.</li> </ul>									

Question	Answer	Marks
5(a)(i)	<ul style="list-style-type: none"> <li>• Any character <u>except</u> colon, space or any alpha-numeric</li> <li>• Reason: character is not in the login information strings</li> </ul>	<b>2</b>
5(a)(ii)	<p>DECLARE <u>LogArray</u> : ARRAY[1 : 20] OF <u>STRING</u></p> <p>1 mark per underline</p>	<b>2</b>

Question	Answer	Marks
5(b)	<p>Pseudocode solution included here for development and clarification of mark scheme. Programming language example solutions appear in the <b>Appendix</b>.</p> <pre> PROCEDURE LogEvents()   DECLARE FileData : STRING   DECLARE ArrayIndex : INTEGER   OPENFILE "LoginFile.txt" FOR APPEND   FOR ArrayIndex ← 1 TO 20 //     IF LogArray[ArrayIndex] &lt;&gt; "****"       THEN         FileData ← LogArray[ArrayIndex]         WRITEFILE ("LoginFile.txt", FileData)       ENDIF     ENDFOR   CLOSEFILE ("LoginFile.txt") ENDPROCEDURE </pre> <p>1 mark for each of the following:</p> <ol style="list-style-type: none"> <li>1. Procedure heading and ending</li> <li>2. Declare <code>ArrayIndex</code> as integer // commented in python</li> <li>3. Open file 'LoginFile' for append</li> <li>4. Correct loop</li> <li>5. extract data from array <b>in a loop</b></li> <li>6. check for unused element <b>in a loop</b></li> <li>7. write data to file <b>in a loop</b></li> <li>8. Close the file <b>outside the loop</b></li> </ol>	8

Question	Answer	Marks
6(a)	<p>Pseudocode solution included here for development and clarification of mark scheme. Programming language example solutions appear in the Appendix.</p> <pre> FUNCTION ValidateRegistration(Registration : STRING) RETURNS                                 BOOLEAN     DECLARE UCaseChar, NumChar : INTEGER     DECLARE NextChar : CHAR     DECLARE ReturnFlag : BOOLEAN     DECLARE n : INTEGER      ReturnFlag ← TRUE     ValidateRegistration ← True      IF LEN(Registration) &lt; 6 OR LEN(Registration) &gt; 9 //check   length     THEN         ReturnFlag ← False     ELSE         FOR n ← 1 TO 3 //check for 3 upper case alpha             NextChar ← MID(Registration, n, 1)             IF NextChar &lt; 'A' AND NextChar &gt; 'Z'             THEN                 ReturnFlag ← False             ENDIF         ENDFOR          FOR n ← 4 TO 5 //check for 2 numeric             NextChar ← MID(Registration, n, 1)             IF NextChar &lt; '0' AND NextChar &gt; '9'             THEN                 ReturnFlag ← False             ENDIF         ENDFOR          FOR n ← 6 TO LEN(Registration) //check remaining   characters             NextChar ← MID(Registration, n, 1)             IF NextChar &lt; 'A' AND NextChar &gt; 'Z'             THEN                 ReturnFlag ← False             ENDIF         ENDFOR     ENDIF     RETURN (ReturnFlag) ENDFUNCTION </pre>	Max 9

Question	Answer	Marks
6(a)	<p>1 mark for each of the following:</p> <ol style="list-style-type: none"> <li>1. Correct Function heading and ending</li> <li>2. Check for correct length</li> <li>3. Extract first three characters</li> <li>4. Check first three characters are capitals</li> <li>5. Extract characters four and five</li> <li>6. Check characters four and five are numeric</li> <li>7. Extract remaining characters</li> <li>8. Check remaining characters are capitals</li> <li>9. Combine all four tests results into a single Boolean value</li> <li>10. Return a Boolean value</li> </ol>	
6(b)	<p><b>String1:</b> (for example, "ABC12XYZ")</p> <p>One mark for a valid string having:</p> <ul style="list-style-type: none"> <li>• Correct length (between 6 and 9 characters)</li> <li>• 3 capital letters followed by...</li> <li>• 2 numeric characters followed by...</li> <li>• between 1 and 4 capital letters</li> </ul> <p><b>String2 to String5:</b></p> <p>1 mark for each string <b>and</b> explanation (testing different rules of the function)</p> <p>Test strings breaking <b>one different</b> rules:</p> <ul style="list-style-type: none"> <li>• Incorrect length</li> <li>• With incorrect number of capital letters at the start</li> <li>• With non-numeric characters in positions 4 and 5</li> <li>• With incorrect number of capital letters at the end</li> <li>• Containing an invalid character (not alpha-numeric)</li> </ul>	<b>5</b>