

Revision Sheet

IGCSE Computer Science · Unit 8 Programming

12 vocabulary items · 40 marks of exam questions (6 questions)

Part 1 · Vocabulary review answer key. Section A wants the exact English spelling.

Question	Answer	Marks
1	variable	1
2	constant	1
3	data type	1
4	string	1
5	iteration	1
6	procedure	1
7	function	1
8	parameter	1
9	array	1
10	one-dimensional (1D) array	1
11	two-dimensional (2D) array	1
12	file	1

13(a)	<p>One mark per mark point, max three</p> <ul style="list-style-type: none"> • Storing string in <code>Phrase</code> • Correct use of <code>LENGTH</code> function • Correct use of <code>UCASE</code> function • Correct outputs of <code>LENGTH</code> and <code>UCASE</code> <p>For example:</p> <pre>Phrase ← "The beginning is the most important part" OUTPUT LENGTH(Phrase) OUTPUT UCASE(Phrase)</pre>
13(b)	<p>One mark for each correct line, max two</p> <pre>40 THE BEGINNING IS THE MOST IMPORTANT PART</pre>

14	<p>One mark per mark point, max five</p> <p>MP1 storing string in <code>Quote</code></p> <p>MP2 correct assignment for <code>Start</code> // sending correct start value</p> <p>MP3 correct assignment for <code>Number</code> // sending correct number of characters</p> <p>MP4 use of <code>SUBSTRING</code> function with first parameter as <code>Quote</code>, or equivalent, plus two other parameters</p> <p>MP5 correct use of <code>LCASE</code> function</p> <p>MP6 two correct outputs</p> <p>For example:</p> <pre>Quote ← "Learning Never Exhausts The Mind" Start ← 25 Number ← 8 OUTPUT SUBSTRING(Quote, Start, Number) OUTPUT LCASE(Quote)</pre>
----	---

15(a)	<pre>01//02//06//10 04(07) and/or 08 03(12)</pre>
15(b)	<p>One mark for each error identified and corrected</p> <p>Line 04 < should be ></p> <p>Line 08 <code>Count</code> should be <code>Counter</code></p> <p>Line 11 <code>ENDWHILE</code> should be <code>ENDIF</code></p> <pre>01 Max ← List[1] 02 Min ← List[1] 03 FOR Counter ← 2 TO 1000 04 IF List[Counter] > Max 05 THEN 06 Max ← List[Counter] 07 ENDIF 08 IF List[Counter] < Min 09 THEN 10 Min ← List[Counter] 11 ENDIF 12 NEXT Counter 13 OUTPUT "Maximum value is ", Max 14 OUTPUT "Minimum value is ", Min</pre>

16(a)	<p>One mark for any two correct lines</p> <pre> DECLARE P : STRING P ← "The world" DECLARE Q : CHAR Q ← 'W' </pre>
16(b)	<p>One mark for each point (max four)</p> <ul style="list-style-type: none"> • converting P to upper case • finding the length of P • using a loop to check for position of Q • using the string operation substring • storing the loop counter in Position if the value is found <p>For example:</p> <pre> P ← UCASE(P) Counter ← 1 Position ← 0 REPEAT IF SUBSTRING(P, Counter, 1) = Q THEN Position ← Counter ENDIF Counter ← Counter + 1 UNTIL Position <> 0 OR Counter = LENGTH(P) </pre>
16(c)	5

17	<p>One mark per mark point (max six)</p> <ul style="list-style-type: none"> • Declaration of appropriate string variable(s) • Input of a line of text • Correct use of LCASE • Correct use of LENGTH • Opening of at least one of the required text files using given names (Main.txt, Lowercase.txt) • Storing of correct line of text to Main.txt • Storing of correct lines to Lowercase.txt • Closing of both text files • Correct output <p>For example:</p> <pre> DECLARE LineOfText : STRING DECLARE LowercaseText: STRING INPUT LineOfText OPENFILE Main.txt FOR WRITE WRITEFILE Main.txt, LineOfText CLOSEFILE Main.txt LowercaseText ← LCASE(LineOfText) OUTPUT LowercaseText, LENGTH(LineOfText) OPENFILE Lowercase.txt FOR WRITE WRITEFILE Lowercase.txt, LowercaseText CLOSEFILE Lowercase.txt </pre>
----	---

18(a)	<p>One mark per mark point, max four</p> <ul style="list-style-type: none"> Line 01 / DECLARE People : ARRAY[1:50, 1:3] OF REAL should be DECLARE People : ARRAY[1:50, 1:3] OF STRING Line 10 / Count ← 100 should be Count ← 1 Line 12 / CASE OF should be REPEAT Line 27 / UNTIL NOT Count should be UNTIL NOT Continue // UNTIL Continue = FALSE <p>Correct algorithm:</p> <pre> 01 DECLARE People : ARRAY[1:50, 1:3] OF STRING 02 DECLARE Count : INTEGER 03 DECLARE Response : CHAR 04 DECLARE Continue : BOOLEAN 05 FOR I ← 1 TO 50 06 FOR J ← 1 TO 3 07 People[I, J] ← "" 08 NEXT J 09 NEXT I 10 Count ← 1 11 Continue ← TRUE 12 REPEAT 13 OUTPUT "Enter the last name" 14 INPUT People[Count, 1] 15 OUTPUT "Enter the first name" 16 INPUT People[Count, 2] 17 OUTPUT "Enter the city" 18 INPUT People[Count, 3] 19 OUTPUT "Do you want to enter another name (Y or N)?" 20 INPUT Response </pre>
18(a)	<pre> 21 IF Response = 'N' 22 THEN 23 Continue ← FALSE 24 ELSE 25 Count ← Count + 1 26 ENDIF 27 UNTIL NOT Continue // UNTIL Response = 'N' </pre>
18(b)	<ul style="list-style-type: none"> Use of appropriate loop Method to check array maximum not exceeded Method to check current / next array element not empty Output of all three array elements per array row (and no more) <p>Example algorithm:</p> <pre> Count ← 1 WHILE Count ≤ 50 AND People[Count, 1] <> "" DO OUTPUT People[Count, 1] OUTPUT People[Count, 2] OUTPUT People[Count, 3] Count ← Count + 1 ENDWHILE </pre>
18(c)	<p>One mark per mark point, max four</p> <p>MP1 Declare/use a variable that is set to the maximum size of the array</p> <p>MP2 ... at the start of the program</p> <p>MP3 After line 18</p> <p>MP4 ... check that the value of the counting variable is not greater than the array maximum variable</p> <p>MP5 ... and if it is do not allow any more entries / set the value of Response to 'N' / add additional condition to UNTIL statement that checks if the counting variable is at maximum</p>