SPECIMEN MATERIAL

GCSE
COMPUTER SCIENCE
8520/1

PAPER 1

Mark scheme

Specimen 2015

v1.1

Mark schemes are prepared by the Lead Assessment Writer and considered, together with the relevant questions, by a panel of subject teachers. This mark scheme includes any amendments made at the standardisation events which all associates participate in and is the scheme which was used by them in this examination. The standardisation process ensures that the mark scheme covers the students' responses to questions and that every associate understands and applies it in the same correct way. As preparation for standardisation each associate analyses a number of students' scripts. Alternative answers not already covered by the mark scheme are discussed and legislated for. If, after the standardisation process, associates encounter unusual answers which have not been raised they are required to refer these to the Lead Assessment Writer.

It must be stressed that a mark scheme is a working document, in many cases further developed and expanded on the basis of students' reactions to a particular paper. Assumptions about future mark schemes on the basis of one year's document should be avoided; whilst the guiding principles of assessment remain constant, details will change, depending on the content of a particular examination paper.

Further copies of this mark scheme are available from aqa.org.uk

The following annotation is used in the mark scheme:

; - means a single mark
// - means alternative response
/ - means an alternative word or sub-phrase
A - means acceptable creditworthy answer. Also used to denote a valid answer that goes beyond the expectations of the GCSE syllabus.
R - means reject answer as not creditworthy
NE - means not enough
I - means ignore
DPT - in some questions a specific error made by a candidate, if repeated, could result in the candidate failing to gain more than one mark. The DPT label indicates that this mistake should only result in a candidate losing one mark on the first occasion that the error is made. Provided that the answer remains understandable, subsequent marks should be awarded as if the error was not being repeated.

Level of response marking instructions

Level of response mark schemes are broken down into levels, each of which has a descriptor. The descriptor for the level shows the average performance for the level. There are marks in each level.

Before you apply the mark scheme to a student's answer read through the answer and annotate it (as instructed) to show the qualities that are being looked for. You can then apply the mark scheme.

Step 1 Determine a level

Start at the lowest level of the mark scheme and use it as a ladder to see whether the answer meets the descriptor for that level. The descriptor for the level indicates the different qualities that might be seen in the student's answer for that level. If it meets the lowest level then go to the next one and decide if it meets this level, and so on, until you have a match between the level descriptor and the answer. With practice and familiarity you will find that for better answers you will be able to quickly skip through the lower levels of the mark scheme.

When assigning a level you should look at the overall quality of the answer and not look to pick holes in small and specific parts of the answer where the student has not performed quite as well as the rest. If the answer covers different aspects of different levels of the mark scheme you should use a best fit approach for defining the level and then use the variability of the response to help decide the mark within the level, ie if the response is predominantly level 3 with a small amount of level 4 material it would be placed in level 3 but be awarded a mark near the top of the level because of the level 4 content.

Step 2 Determine a mark

Once you have assigned a level you need to decide on the mark. The descriptors on how to allocate marks can help with this. The exemplar materials used during standardisation will help. There will be an answer in the standardising materials which will correspond with each level of the mark scheme. This answer will have been awarded a mark by the Lead Examiner. You can compare the student's answer with the example to determine if it is the same standard, better or worse than the example. You can then use this to allocate a mark for the answer based on the Lead Examiner's mark on the example.

You may well need to read back through the answer as you apply the mark scheme to clarify points and assure yourself that the level and the mark are appropriate.

Indicative content in the mark scheme is provided as a guide for examiners. It is not intended to be exhaustive and you must credit other valid points. Students do not have to cover all of the points mentioned in the Indicative content to reach the highest level of the mark scheme.

An answer which contains nothing of relevance to the question must be awarded no marks.

<table>
<tr><th>Qu</th><th>Part</th><th>Marking guidance</th><th>Total marks</th></tr>
<tr><td>01</td><td>1</td><td>2 marks for AO1 (recall)

1 mark for each correct lozenge shaded;
A (It saves the programmer time)
C (Program code written in high-level language is often easier for humans to understand)
R. if more than two lozenges are shaded.</td><td>2</td></tr>
<tr><td>01</td><td>2</td><td>Marks are for AO1 (understanding)

1 mark for each correct answer to a maximum of two.

Examples include:
Assembly language is easier for humans to read;
A human programmer is less likely to make mistakes in assembly language;
It is easier to remember the assembly instructions than the binary codes;
Assembly language instructions can be given labels to help readability;</td><td>2</td></tr>
<tr><td>02</td><td>1</td><td>3 marks for AO2 (apply)

Mark as follows:
1 mark for all correct values in the correct order for column <code>a</code>;
1 mark for all correct values in the correct order for column <code>b</code>;
1 mark for column <code>c</code> having only the value 7;

The completed trace table should have these values although the candidate may have entered the values on different rows (do not penalise as long as the order of the values is correct).

<table><tr><th>a</th><th>b</th><th>c</th></tr><tr><td>3</td><td>4</td><td>7</td></tr><tr><td>4</td><td>0</td><td></td></tr><tr><td>5</td><td>-5</td><td></td></tr><tr><td>6</td><td>-11</td><td></td></tr><tr><td>7</td><td>-18</td><td></td></tr><tr><td></td><td></td><td></td></tr></table></td><td>3</td></tr>
<tr><td>02</td><td>2</td><td>2 mark for AO2 (apply)

1 mark for each correct answer to a maximum of two.

The value of c is constant/does not change;
(a+b)/a/b may change;
There is a logical error;</td><td>2</td></tr>
</table>

02	3	**Mark is for AO2 (apply)** 0;	1

03	1	**Mark is for AO2 (apply)** **B** (0) only; **If more than one lozenge shaded then mark is not awarded**	1

03	2	**Mark is for AO2 (apply)** **C** (≤) only; **If more than one lozenge shaded then mark is not awarded**	1

03	3	**Mark is for AO2 (apply)** **C** (1) only; **If more than one lozenge shaded then mark is not awarded**	1

03	4	**Mark is for AO2 (apply)** **D** (`y + 1`) only; **If more than one lozenge shaded then mark is not awarded**	1

04	1	**5 marks for AO2 (apply)** **Mark as follows (max 5):** **1 mark** for only the value 43 in column `val`; **1 mark** for all correct values in the correct order for column `left`; **1 mark** for all correct values in the correct order for column `right`; **1 mark** for all correct values in the correct order for column `mid`; **1 mark** for all correct values in the correct order for column `arr[mid]` (the final 4 in mid is already given); The completed trace table should have these values although the candidate may have entered the values on different rows (do not penalise as long as the order of the values is correct). **val** | **left** | **right** | **mid** | **arr[mid]** 43 | 1 | 6 | 3 | 13 | 4 | 5 | 5 | 655 | (4) | 4 | 4 | 43	5

04	2	**Mark is for AO2 (apply)** The array/`arr` is unsorted/not in order;	1

04	3	**Mark is for AO2 (apply)** **C** `WHILE left < right AND left > right` only; **If more than one lozenge shaded then mark is not awarded**	1

04	4	**4 marks for AO3 (program)**
		Mark as follows:
		1 mark for using selection;
		1 mark for setting found is assigned the value true and the value false in two separate places (**A.** if selection logic is incorrect);
		1 mark for using the correct Boolean condition (**A.** if `right` has been used instead of `left`);
		1 mark for overall correctness of the solution;
		
		An example answer is:
		`IF val = arr[left] THEN`
		`  found ← true`
		`ELSE`
		`  found ← false`
		`ENDIF`
		
		Another example using `right` instead of `left` is:
		
		`found ← false`
		`IF val = arr[right] THEN`
		`  found ← true`
		`ENDIF`
			4

05	1	**3 marks for AO2 (apply)** **3 marks** if the answer given is 600,000 bits (**I.** no units); If the answer is **incorrect** award then: **1 mark** for converting 2 minutes into 120 seconds; **1 mark** for showing multiplication of the number of seconds (I. if not correct) by 1000 by 5 (A. multiplying just by 5000);	3

05	2	**2 marks for AO2 (apply)** **2 marks** if the answer given is 3 KB (**I.** no units); If the answer is **incorrect** award then: **1 mark** for for performing a division by 8, evidenced by answer or	2

		method being shown; **1 mark** for performing a division 1000, evidenced by answer or method being shown; **MAX 1** if answer is incorrect. **A.** if students have used 1024 instead of 1000.	

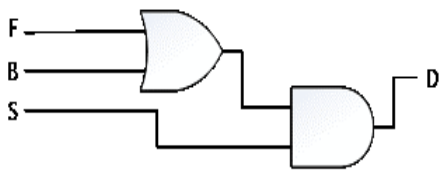
05	3	**4 Marks for AO2 (apply)** **Mark as follows:** **1 mark** for each correct frequency-value pair up to a maximum of four; The correct answer is: `6 0, 7 1, 12 0, 7 1` A. equivalent symbols such as (6,0), (7,1) etc A. if frequency and value are the other way around, ie 0 6, 1 7, etc	4

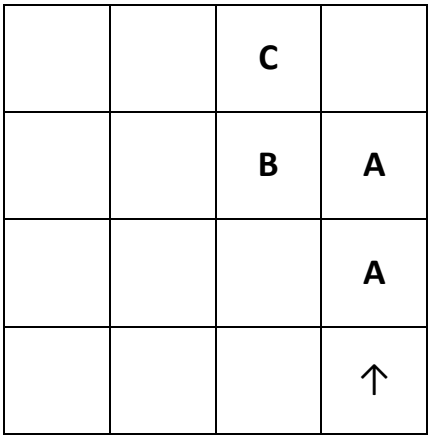
05	4	**Mark is for AO1 (understanding)** **C** (It will not lose any of the original data) only; **If more than one lozenge shaded then mark is not awarded**	1

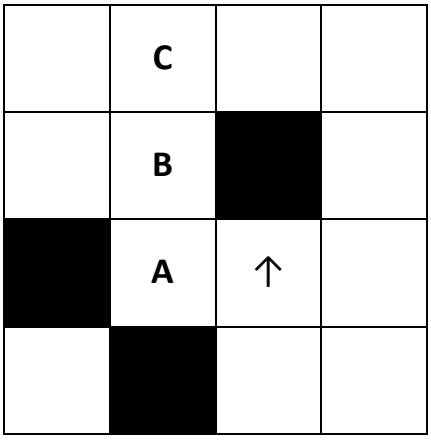
06		**7 marks for AO3 (program)**
		Mark as follows:
		1 mark if it outputs 12am if the input is 0;
		1 mark if it outputs 12pm if the input is 12;
		1 mark for selecting just the inputs 1 to 11;
		1 mark if it outputs 1am to 11am correctly given the corresponding inputs 1 to 11;
		1 mark for selecting just the inputs 13 to 23;
		1 mark for adjusting the value of hour to be 12 less (using subtraction, modulus or similar);
		1 mark if it outputs 1pm to 11pm correctly given the corresponding inputs 13 to 23;
		
		An example answer is:
		`IF hour = 0 THEN`
		`  OUTPUT 12`
		`  OUTPUT 'am'`
		`ELSE`
		`  IF hour < 12 THEN`
		`  OUTPUT hour`
		`  OUTPUT 'am'`
		`  ELSE`
		`  IF hour = 12 THEN`
		`  OUTPUT 12`
		`  OUTPUT 'pm'`
		`  ELSE`
		`  hour ← hour - 12`
		`  OUTPUT hour`
		`  OUTPUT 'pm'`
		`  ENDIF`
		`  ENDIF`
		`ENDIF`
			7

07	1	**Mark is for AO1 (recall)** 1 mark if all cells are correct; **A** | **B** | **A OR B** 0 | 0 | 0 0 | 1 | 1 1 | 0 | 1 1 | 1 | 1	1

07	2	**3 marks for AO2 (apply)** **Mark as follows:** **1 mark** for all X correct; **1 mark** for all Y correct (allow follow on if X is incorrect but it correctly shows B AND X); **1 mark** for all Z correct (allow follow on if complete negation of Y even if Y is incorrect); The correct truth table is: **A** | **B** | **X** | **Y** | **Z** 0 | 0 | **0** | **0** | **1** 0 | 1 | **0** | **0** | **1** 1 | 0 | **0** | **0** | **1** 1 | 1 | **1** | **1** | **0**	3

07	3	**3 marks for AO2 (apply)** **Mark as follows:** **1 mark** if F and B are both inputs to an OR gate; **1 mark** if S and the output of F and B (allowing for earlier errors) are inputs to an AND gate; **1 mark** if the output of the gate that has S as one input, and the output of F and B as another input, has its output connected to D; The completed circuit is shown here:  	3

08	1	**3 marks for AO2 (apply)** **Mark as follows:** **1 mark** for the robot moving to **both** squares marked **A**; **1 mark** for the robot moving to the square marked **B**; **1 mark** for the robot moving to the square marked **C**; 	3

08	2	**3 marks for AO2 (apply)** **Mark as follows:** **1 mark** for the robot moving to the square marked **A**; **1 mark** for the robot moving to the square marked **B**; **1 mark** for the robot moving to the square marked **C**; 	3

08	3	**4 marks for AO3 (program)** **Mark as follows:** **1 mark** for 1 correct instruction in the correct place; **2 marks** for 2 correct instructions in the correct place; **3 marks** for 3 correct instructions in the correct place; **4 marks** for all 5 correct instructions in the correct place; The two correct versions are shown below. The correct instructions are shown in bold: **distance ← 4** WHILE distance > 0 **Forward(distance)** **TurnLeft()** Forward(distance) **TurnLeft()** **distance ← distance – 1** ENDWHILE OR **distance ← 4** WHILE distance > 0 **Forward(distance)** **TurnLeft()** Forward(distance) **distance ← distance – 1** **TurnLeft()** ENDWHILE **I.** Minor spelling mistakes/incorrect case/missing brackets.	4

<table>
<tr><td>09</td><td>1</td><td>4 marks for AO2 (apply)
Mark as follows:
1 mark for <code>num2</code> changing to 24;
1 mark for <code>num2</code> changing to 9;
1 mark for <code>num1</code> changing to 6;
1 mark for both <code>num1</code> and <code>num2</code> ending as 3;

The completed trace table is shown here (candidates' values may appear on different rows):

<table><tr><th>num1</th><th>num2</th></tr><tr><td>15</td><td>39</td></tr><tr><td></td><td>24</td></tr><tr><td></td><td>9</td></tr><tr><td>6</td><td></td></tr><tr><td></td><td>3</td></tr><tr><td>3</td><td></td></tr></table></td><td>4</td></tr>
</table>

09	2	**Mark is for AO1 (understanding)** 4;	1

09	3	**Mark is for AO1 (understanding)** 3;	1

09	4	**Mark is for AO1 (understanding)** 4;	1

09	5	**2 marks for AO1 (understanding)** 1 mark for the benefit of using a subroutine; 1 mark for the explanation that matches the benefit; Examples include: It allows code reuse; because the code can be called in different places in the code without being written out in full; It is easier to test; because it can be tested in isolation from the rest of the code; It is easier to maintain; because as long as the interface is maintained it can be safely changed or rewritten;	2

10	1	**Mark is for AO2 (apply)**	1

		B (`bitmappixel`) only; **R. if more than one lozenge shaded**	

10	2	**2 marks for AO2 (apply)** **1 mark for A** `a[1] = b[1] OR a[1] = c[1];` **1 mark for E** `a[LENGTH(a) - LENGTH(c)] = c[3];` **If more than two lozenges are shaded then no marks are awarded**	2

10	3	**9 marks for AO3 (program)**
		Mark as follows:
		1 mark for comparing the length of the two strings;
		1 mark for returning false if the prefix is longer than the word;
		1 mark for using iteration to compare characters in turn;
		1 mark for using a condition that will compare all characters in the prefix if needed (ie the condition may have the opposite value if two characters in word and pre are found to be different);
		1 mark for using an index variable to access the individual characters in the array;
		1 mark for using selection to check if the characters of both arrays are the same (or not the same if the opposite logic is used);
		1 mark to record false if pre is not a prefix word;
		1 mark to record true if the pre is a prefix word;
		1 mark to return the correct value of either true or false;
		
		Two complete pseudo code answers are given below:
		
		`SUBROUTINE Prefix(word, pre)`
		`  IF LEN(pre) > LEN(word) THEN`
		`  RETURN false`
		`  ELSE`
		`  i ← 1`
		`  is_prefix ← true`
		`  WHILE i ≤ LEN(pre)`
		`  IF word[i] ≠ pre[i] THEN`
		`  is_prefix ← false`
		`  ENDIF`
		`  i ← i + 1`
		`  ENDWHILE`
		`  RETURN is_prefix`
		`  ENDIF`
		`ENDSUBROUTINE`
			9

		SUBROUTINE Prefix(word, pre)
		  IF LEN(pre) > LEN(word) THEN
		  RETURN false
		  ELSE
		  i ← 1
		  WHILE i ≤ LEN(pre)
		  IF word[i] ≠ pre[i] THEN
		  RETURN false
		  ENDIF
		  i ← i + 1
		  ENDWHILE
		  RETURN true
		  ENDIF
		ENDSUBROUTINE	

Copyright © 2016 AQA and its licensors. All rights reserved.
AQA Education (AQA) is a registered charity (registered charity number 1073334) and a company limited by guarantee registered in England and Wales (company number 3644723). Registered address: AQA, Devas Street, Manchester M15 6EX

7 June 2016