



CANDIDATE  
NAME

--

CENTRE  
NUMBER

--	--	--	--	--

CANDIDATE  
NUMBER

--	--	--	--

## 0478/21

May/June 2020

**1 hour 45 minutes**

You must answer on the question paper.

No additional materials are needed.

- Answer **all** questions.
- **Do not attempt Tasks 1, 2 and 3** in the copy of the pre-release material on page 2; these are for information only.
- Use a black or dark blue pen. You may use an HB pencil for any diagrams or graphs.
- Write your name, centre number and candidate number in the boxes at the top of the page.
- Write your answer to each question in the space provided.
- Do **not** use an erasable pen or correction fluid.
- Do **not** write on any bar codes.
- Calculators must **not** be used in this paper.

- The total mark for this paper is 50.
- The number of marks for each question or part question is shown in brackets [ ].
- No marks will be awarded for using brand names of software packages or hardware.

This document has **16** pages. Blank pages are indicated.

## Section A

You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

**DO NOT** attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

## Pre-release material

A shop sells a range of mobile devices, SIM cards and accessories as shown in the table:

Category	Item code	Description	Price (\$)
Phone	BPCM	Compact	29.99
Phone	BPSH	Clam Shell	49.99
Phone	RPSS	RoboPhone – 5-inch screen and 64 GB memory	199.99
Phone	RPLL	RoboPhone – 6-inch screen and 256 GB memory	499.99
Phone	YPLS	Y-Phone Standard – 6-inch screen and 64 GB memory	549.99
Phone	YPLL	Y-Phone Deluxe – 6-inch screen and 256 GB memory	649.99
Tablet	RTMS	RoboTab – 8-inch screen and 64 GB memory	149.99
Tablet	RTLTM	RoboTab – 10-inch screen and 128 GB memory	299.99
Tablet	YTLM	Y-Tab Standard – 10-inch screen and 128 GB memory	499.99
Tablet	YTLL	Y-Tab Deluxe – 10-inch screen and 256 GB memory	599.99
SIM card	SMNO	SIM Free (no SIM card purchased)	0.00
SIM card	SMPG	Pay As You Go (SIM card purchased)	9.99
Case	CSST	Standard	0.00
Case	CSLX	Luxury	50.00
Charger	CGCR	Car	19.99
Charger	CGHM	Home	15.99

Write and test a program or programs for this shop.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All arrays, variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

**Task 1** – Setting up the system.

Write a program to:

- use appropriate data structures to store the item code, description and price information for the mobile devices, SIM cards and accessories
- allow the customer to choose a specific phone or tablet
- allow phone customers to choose whether the phone will be SIM Free or Pay As You Go
- allow the customer to choose a standard or luxury case
- allow the customer to choose the chargers required (none, one or both may be purchased)
- calculate the total price of this transaction
- output a list of the items purchased and the total price.

**Task 2** – Allow a customer to order multiple mobile devices.

Extend **Task 1** to:

- offer the customer the opportunity to purchase an additional mobile device
- if required, perform bulleted steps 2 to 7 of **Task 1** for each additional mobile device and calculate a running total for the customer
- once no further devices are required, output the total the customer will need to pay.

**Task 3** – Offering discounts.

Extend the program to allow a discount of 10% off the price of every additional phone or tablet purchased.

Output the new total the customer will need to pay and the amount of money saved.

1 All variables, constants and other identifiers must have meaningful names.

(a) (i) Identify **two** arrays you could have used for **Task 1** and, in each case, state its purpose.

Array 1 .....

Purpose .....

.....

.....

Array 2 .....

Purpose .....

.....

.....

[4]

(ii) Identify **two** variables you could have used for **Task 1** and, in each case, state its purpose.

Variable 1 .....

Purpose .....

.....

.....

Variable 2 .....

Purpose .....

.....

.....

[4]

(b) Explain why the item code data could **not** be stored as a real data type and identify the most suitable data type for the item code data.

.....

.....

.....

.....

.....

..... [2]

- (c) Write an algorithm to show how you input your choice of mobile device and SIM card (part of **Task 1**), using **either** pseudocode, programming statements **or** a flowchart. It is **not** necessary to show initialisation or setting up of arrays containing product details.

This image shows a full page of white paper with horizontal dashed lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



- (d) Explain how your program completes **Task 3**. Any programming statements used in your answer must be fully explained.

..... [5]

**Section B starts on page 8.**

## Section B

- 2 Tick (✓) **one** box in each row to identify if the statement about structure diagrams is true or false.

Statement	True (✓)	False (✓)
A structure diagram is a piece of code that is available throughout the structure of a program.		
A structure diagram shows the hierarchy of a system.		
A structure diagram is another name for an array.		
A structure diagram shows the relationship between different components of a system.		

[2]

- 3 Programs can perform validation and verification checks when data is entered.

- (a) Give the names of **two** different validation checks and state the purpose of each one.

Check 1 .....

Purpose .....

.....

.....

Check 2 .....

Purpose .....

.....

.....

[4]

- (b) Give the name of **one** verification check.

..... [1]

- (c) Describe the difference between validation and verification.

.....

.....

.....

..... [2]



- 4 The pseudocode algorithm shown should allow numbers to be entered and should allow 50 numbers to be stored in an array.

```
Count ← 0
REPEAT
  INPUT Values[Count]
  Count ← Count + 1
UNTIL Count = 0
```

- (a) Explain why the algorithm will never end.

.....

.....

.....

.....

..... [2]

- (b) Re-write the original pseudocode so that it terminates correctly **and** also prevents numbers below 100 from being stored in the array `Values[ ]`

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

- (c) Describe how you could change your pseudocode in **part (b)** so that it prevents numbers below 100 and above 200 from being stored in the array `Values[ ]`

.....

.....

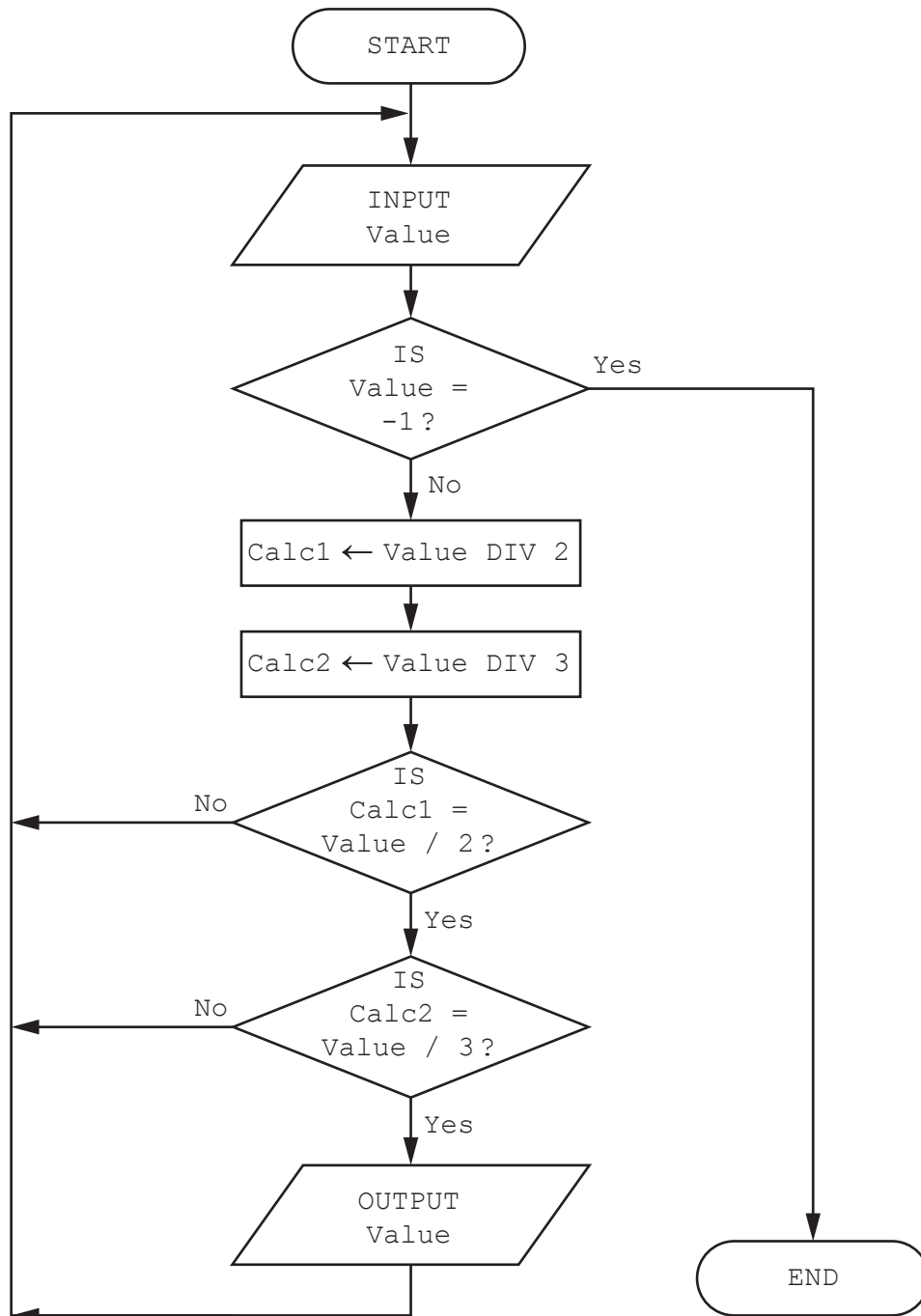
.....

..... [2]

5 The flowchart represents an algorithm.

The predefined function `DIV` gives the value of the result of integer division, for example,  $y \leftarrow 9 \text{ DIV } 4$  gives  $y$  a value of 2

An input value of  $-1$  ends the algorithm.



(a) Complete the trace table for the input data:

50, 33, 18, 15, 30, -1, 45, 12, 90, 6

Value	Calc1	Calc2	OUTPUT

[4]

(b) Describe the purpose of the algorithm.

.....

.....

..... [2]

- 6 A garden centre sells garden tools and stores details of these in a database table named TOOLS. **Code** is the primary key in the TOOLS table.

Code	Description	Price (\$)	Quantity_Stock	Quantity_Ordered
GFLG	Garden Fork	50.00	1	50
GSLG	Garden Spade	50.00	11	0
GHLG	Garden Hoe	45.00	8	0
HFSM	Hand Fork	9.99	42	0
HSSM	Hand Spade	9.99	40	0
HWSM	Hand Weeder	9.99	11	0
HS20	Hose (20 metres)	45.00	10	0
HS35	Hose (35 metres)	60.00	2	0
HS50	Hose (50 metres)	75.00	20	60
YBLG	Yard Brush	24.99	100	0
LMHD	Lawn Mower	99.99	5	0
LMBT	Lawn Mower (Battery)	249.99	7	0
LMPT	Lawn Mower (Petrol)	349.99	10	25
TRBT	Edge Trimmer (Battery)	79.99	15	0
TRPT	Edge Trimmer (Petrol)	59.99	20	0
SHSM	Shears	40.00	40	0
HCSM	Hedge Clippers	40.00	45	0

- (a) State the purpose of the primary key in the TOOLS table.

.....

.....

..... [1]

- (b) List the output from the data shown in the table TOOLS that would be given by this query-by-example.

Field:	Code	Description	Price (\$)	Quantity_Stock	Quantity_Ordered
Table:	TOOLS	TOOLS	TOOLS	TOOLS	TOOLS
Sort:					Descending
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			>40	>0	>0
or:					

.....

.....

.....

.....

.....

..... [3]

- (c) Complete the query-by-example grid to output the tools where the quantity in stock is below 25. Only show the Code, Description and Quantity\_Stock fields in ascending order of Code.

Field:				
Table:				
Sort:				
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

[3]





**BLANK PAGE**

---

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge Assessment International Education Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at [www.cambridgeinternational.org](http://www.cambridgeinternational.org) after the live examination series.

Cambridge Assessment International Education is part of the Cambridge Assessment Group. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which itself is a department of the University of Cambridge.





# Cambridge IGCSE™

---

COMPUTER SCIENCE

0478/21

Paper 2

May/June 2020

MARK SCHEME

Maximum Mark: 50

---

**Published**

Students did not sit exam papers in the June 2020 series due to the Covid-19 global pandemic.

This mark scheme is published to support teachers and students and should be read together with the question paper. It shows the requirements of the exam. The answer column of the mark scheme shows the proposed basis on which Examiners would award marks for this exam. Where appropriate, this column also provides the most likely acceptable alternative responses expected from students. Examiners usually review the mark scheme after they have seen student responses and update the mark scheme if appropriate. In the June series, Examiners were unable to consider the acceptability of alternative responses, as there were no student responses to consider.

Mark schemes should usually be read together with the Principal Examiner Report for Teachers. However, because students did not sit exam papers, there is no Principal Examiner Report for Teachers for the June 2020 series.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the June 2020 series for most Cambridge IGCSE™ and Cambridge International A & AS Level components, and some Cambridge O Level components.

---

This document consists of **10** printed pages.

**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)	<p>Any meaningful array related to <b>Task 1 – one</b> mark (max <b>two</b>) e.g.  ItemCode  Description</p> <p>Correct purpose for each array related to <b>Task 1 – one</b> mark (max <b>two</b>) e.g.  ...to store the item codes  ...to store the descriptions of the items for sale</p>	<b>4</b>
1(a)(ii)	<p>Any meaningful variable related to <b>Task 1 – one</b> mark (max <b>two</b>) e.g.  PurchaseItem  TotalPrice</p> <p>Correct purpose for each variable related to <b>Task 1 – one</b> mark (max <b>two</b>) e.g.  ... to allow input of an item/code for purchase  ... to store/calculate the total price of the transaction</p>	<b>4</b>
1(b)	<p>Any <b>one</b> correct statement e.g.</p> <ul style="list-style-type: none"> <li>• The Code data is made up of letters/alphabetic characters/not numbers</li> <li>• Real data must have numerical value/would not be used in calculations</li> </ul> <p><b>One</b> mark for:</p> <ul style="list-style-type: none"> <li>• String</li> </ul>	<b>2</b>

Question	Answer	Marks
1(c)	<p>Any <b>five</b> from</p> <ul style="list-style-type: none"> <li>• Input with message to allow choice of mobile device</li> <li>• Attempt to validate input to check for valid item code</li> <li>• Accurate validation of input to check for valid item code</li> <li>• Determination of whether device is phone or tablet</li> <li>• Restriction to only allow input for SIM card required if mobile device is a phone</li> <li>• Input with message to find out if a SIM card is required</li> </ul> <p><b>Example answer</b></p> <pre> OUTPUT "Which type of phone or tablet would you like? Input the Item Code" DeviceFlag ← False WHILE DeviceFlag = False   INPUT DeviceCode   Count ← 0   WHILE Count &lt; 10 DO     IF DeviceCode = ItemCode[Count]       THEN         IF Count &lt; 6           THEN             DeviceType ← "Phone"           ELSE             DeviceType ← "Tablet"           ENDIF         DeviceFlag ← True         Count ← 10       ENDIF     Count ← Count + 1   ENDWHILE   IF DeviceFlag = False     THEN </pre>	<b>5</b>

Question	Answer	Marks
1(c)	<pre>       OUTPUT "Your code doesn't exist, please try again"     ENDIF   ENDWHILE   IF DeviceType ← "Phone"     THEN       OUTPUT "Would you like a SIM Card? (Answer Y or N)"       INPUT SimRequired     ENDIF </pre>	
1(d)	<p>Any <b>five</b> from</p> <ul style="list-style-type: none"> <li>• Explanation of finding if more than one device is purchased</li> <li>• Explanation of application of device discount</li> <li>• Explanation of calculating the discount(s) and finding the new price</li> <li>• Explanation of outputting new total</li> <li>• Explanation of outputting amount saved</li> <li>• Explanation of messages used</li> </ul>	<b>5</b>

Question	Answer	Marks															
2	<table border="1"> <thead> <tr> <th>Statement</th><th>True (✓)</th><th>False (✓)</th></tr> </thead> <tbody> <tr> <td>A structure diagram is a piece of code that is available throughout the structure of a program.</td><td></td><td>✓</td></tr> <tr> <td>A structure diagram shows the hierarchy of a system.</td><td>✓</td><td></td></tr> <tr> <td>A structure diagram is another name for an array.</td><td></td><td>✓</td></tr> <tr> <td>A structure diagram shows the relationship between different components of a system.</td><td>✓</td><td></td></tr> </tbody> </table> <p><b>Two</b> marks for four correct rows. <b>One</b> mark for three correct rows.</p>	Statement	True (✓)	False (✓)	A structure diagram is a piece of code that is available throughout the structure of a program.		✓	A structure diagram shows the hierarchy of a system.	✓		A structure diagram is another name for an array.		✓	A structure diagram shows the relationship between different components of a system.	✓		<b>2</b>
Statement	True (✓)	False (✓)															
A structure diagram is a piece of code that is available throughout the structure of a program.		✓															
A structure diagram shows the hierarchy of a system.	✓																
A structure diagram is another name for an array.		✓															
A structure diagram shows the relationship between different components of a system.	✓																

Question	Answer	Marks
3(a)	<p><b>One</b> mark for each correct validation check (max <b>two</b>)</p> <ul style="list-style-type: none"> <li>• Range</li> <li>• Length</li> <li>• Type</li> <li>• Check Digit</li> </ul> <p><b>One</b> mark for each correct related purpose (max <b>two</b>) e.g.</p> <ul style="list-style-type: none"> <li>• To make sure the data entered falls within a specific set of values</li> <li>• To make sure the data entered is no longer than specified</li> <li>• To make sure the data entered follows rules related to whether it is numbers or letters</li> <li>• To make sure an identification code entered is genuine or possible</li> </ul>	<b>4</b>
3(b)	<p><b>One</b> mark for correct verification check (max <b>one</b>)</p> <ul style="list-style-type: none"> <li>• Double (data) entry</li> <li>• Visual check</li> </ul>	<b>1</b>
3(c)	<p>Any <b>two</b> correct statements (max <b>two</b>) e.g.</p> <ul style="list-style-type: none"> <li>• Validation checks if the data entered is possible/it cannot check if data has been entered correctly.</li> <li>• Verification checks if the data entered matches the data submitted for entry/ it does not check if data matches set criteria.</li> </ul>	<b>2</b>

Question	Answer	Marks
4(a)	<p>Any <b>two</b> correct statements (max <b>two</b>) e.g.</p> <ul style="list-style-type: none"> <li>• The value of the variable Count begins as 0 ...</li> <li>• ... and is incremented by 1 before it is tested by the loop condition</li> <li>• Count will never be 0 at the end of the loop</li> </ul>	<b>2</b>

Question	Answer	Marks
4(b)	<pre> Count ← 0 REPEAT   INPUT Number   IF Number &gt;= 100     THEN       Values[Count] ← Number     ENDIF Count ← Count + 1 UNTIL Count = 50 </pre> <p> <b>One</b> mark – separate INPUT statement  <b>One</b> mark – IF statement attempted  <b>One</b> mark – IF statement completely correct  <b>One</b> mark – termination of loop updated </p>	<b>4</b>
4(c)	<p>Any <b>two</b> correct statements (max <b>two</b>) e.g.</p> <ul style="list-style-type: none"> <li>• Alter the IF statement/add a second IF statement/comparison that's already there ...</li> <li>• ... so that additional criteria set an upper limit of <math>\leq 200</math></li> </ul>	<b>2</b>



Question	Answer				Marks																																												
5(a)	<table><tr><th>Value</th><th>Calc1</th><th>Calc2</th><th>OUTPUT</th></tr><tr><td>50</td><td>25</td><td>16</td><td></td></tr><tr><td>33</td><td>16</td><td>11</td><td></td></tr><tr><td>18</td><td>9</td><td>6</td><td>18</td></tr><tr><td>15</td><td>7</td><td>5</td><td></td></tr><tr><td>30</td><td>15</td><td>10</td><td>30</td></tr><tr><td>-1</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>				Value	Calc1	Calc2	OUTPUT	50	25	16		33	16	11		18	9	6	18	15	7	5		30	15	10	30	-1																				4
	Value	Calc1	Calc2	OUTPUT																																													
	50	25	16																																														
	33	16	11																																														
	18	9	6	18																																													
	15	7	5																																														
	30	15	10	30																																													
	-1																																																
One mark for each correct column (max four)																																																	
5(b)	Any two correct statements e.g.				2																																												
	<ul style="list-style-type: none"><li>The program outputs a value</li><li>That is divisible by 6 // 2 and 3</li></ul>																																																

Question	Answer					Marks																																				
6(a)	To uniquely identify a product (in TOOLS)					1																																				
6(b)	<table><tr><td>HS50</td><td>Hose (50 metres)</td><td>60</td></tr><tr><td>GFLG</td><td>Garden Fork</td><td>50</td></tr><tr><td>LMPT</td><td>Lawn Mower (Petrol)</td><td>25</td></tr></table> <p><b>One</b> mark for correct data <b>One</b> mark for correct format <b>One</b> mark for data in correct order</p>					HS50	Hose (50 metres)	60	GFLG	Garden Fork	50	LMPT	Lawn Mower (Petrol)	25	3																											
HS50	Hose (50 metres)	60																																								
GFLG	Garden Fork	50																																								
LMPT	Lawn Mower (Petrol)	25																																								
6(c)	<table><tr><td>Field:</td><td>Code</td><td>Description</td><td>Quantity_Stock</td><td></td><td></td></tr><tr><td>Table:</td><td>TOOLS</td><td>TOOLS</td><td>TOOLS</td><td></td><td></td></tr><tr><td>Sort:</td><td>Ascending</td><td></td><td></td><td></td><td></td></tr><tr><td>Show:</td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>Criteria:</td><td></td><td></td><td>&lt;25</td><td></td><td></td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td><td></td></tr></table> <p><b>One</b> mark for each completely correct column (max <b>three</b>)</p>					Field:	Code	Description	Quantity_Stock			Table:	TOOLS	TOOLS	TOOLS			Sort:	Ascending					Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Criteria:			<25			or:						3
Field:	Code	Description	Quantity_Stock																																							
Table:	TOOLS	TOOLS	TOOLS																																							
Sort:	Ascending																																									
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																																					
Criteria:			<25																																							
or:																																										