

QUESTION 1. Mark Scheme

Cambridge IGCSE – May/June 2016



4

Riders	Reject	Height	Output
0	0		
1		1.4	
2		1.3	
	1	1.1	
3		1.3	
	2	1.0	
4		1.5	
5		1.2	
6		1.3	
7		1.4	
8		1.3	
			Ready to go 2

(1 mark)

(1 mark)

(1 mark)

(1 mark)

[4]

5

- FOR (... TO ... NEXT)...
- ... a set number of iterations
- WHILE (... DO ... ENDWHILE) ...
- ... used where the loop may never be executed/whilst a specified condition exists

[4]

6

- (a) – all (fields) have (1 mark) duplicate entries (1 mark)
- none (of the fields) (1 mark) have unique entries(1 mark)

[2]

(b) – e.g. StaffNumber

- Uniquely identifies each member of staff//no duplicates//different for each member of staff

[2]

Page 6	Mark Scheme	Syllabus	Paper
	Cambridge IGCSE – May/June 2016	0478	21

(c)

Field:	Department	Name		
Table:	STAFFPHONE	STAFFPHONE		
Sort:	Ascending	Ascending		
Show:	☒	☒	☐	☐
Criteria:				
or:				

(2 marks)

(2 marks)

(1 mark for correct order and number of fields shown)

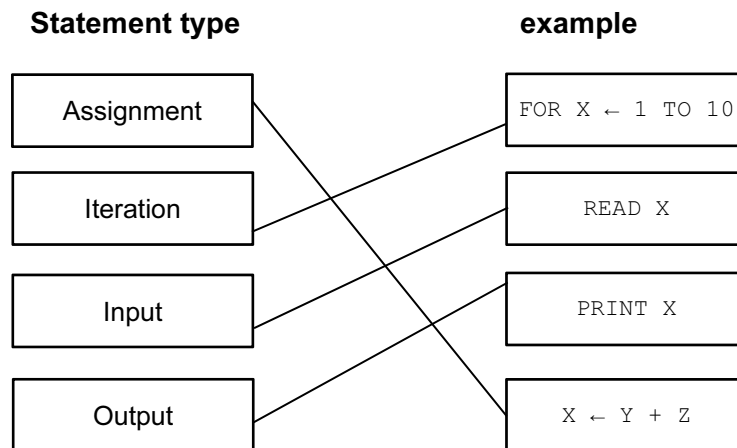
[5]

QUESTION 2.

Mark Scheme
Cambridge IGCSE – May/June 2016



- 4 1 mark for each correct line, maximum 3 (zero correct 0, one correct 1, two correct 2, three correct 3), each box must have only one connection.



[3]

- 5 – data structure (one—dimensional) array
– reason to simplify programming/ make programs shorter, etc.

[2]

- 6 – IF (... THEN ... ELSE ... ENDIF)
– CASE (... OF ... OTHERWISE ... ENDCASE)

[2]

- 7 (a) – 7

[1]

- (b) – Brochure Number.....
– Uniquely identifies each record/each Brochure Number different/no duplicates

[2]

- (c) – Number of Seats – number/integer
– Price in \$ – currency/real

[2]

- (d) 1 mark for each correct result, 1 mark for the results in descending order of price

- Recliner sofa 1,200 RS23
– Recliner chair 600 RC01

[3]

QUESTION 3.

PUBLISHED



Question	Answer	
5(a)	Any two from: - Loop with 300 repetitions (starting at 1) / Loops from 1 to 300 - Values input/stored (in consecutive/different locations) in an array (at position I) - Increases the loop counter/I value by 1 (and returns to the start of the loop)	
5(b)	Any one from: REPEAT (... UNTIL) WHILE (... DO ... ENDWHILE)	1
5(c)	- Prompt and input number (1) - Checking the input number is between 0 and 100 - both limits (1) - Correct error message (1) Many correct algorithms. This is an example only. <pre> OUTPUT "Enter a number between 0 and 100 " INPUT Number IF Number < 0 OR Number > 100 THEN OUTPUT "The number you have entered is outside the specified range" ENDIF </pre>	3

Question	Answer				Marks
6	HighF	HighC	TempF	OUTPUT	5
	-100	-100			
			68		
	68	18	46		
	68	18	50		
	68	18	86		
	86	27	65		
	86	27	50		
	86	27	40		
	86	27	30		
	86	27	-1	The highest temperature is, 86 Fahrenheit, 27 Celsius.	
	(1 Mark) (1 Mark) (1 Mark) (2 Marks – see below)				
The literal correct output is “The highest temperature is, 86 Fahrenheit, 27 Celsius.” 1 mark for values 86 and 27, 1 mark for correct output words, spacing and punctuation.					

QUESTION 4.

PUBLISHED



Question	Answer			
3	Weight	Reject	Total Weight	OUTPUT
		0	0	
	13		13	
	17		30	
	26	1		
	25		55	
	5		60	
	10		70	
	15		85	
	35	2		
	20		105	
			85	Weight of items 85 Number of items rejected 2
	(1mark)	(1 mark)	(1 mark to 1st 85) (1 mark 105, 85)	(1 mark)

Question	Answer	Marks
4(a)	Error - Count $\leftarrow$ 0 Correction - Count $\leftarrow$ 1 or Error - UNTIL Count > 100 Correction - UNTIL Count >= 100 or UNTIL Count = 100 or UNTIL Count > 99	2
4(b)	- use of FOR with correct start and end values ... - ... use of NEXT - ... removal of increment for Count Sample algorithm Sum $\leftarrow$ 0 FOR Count $\leftarrow$ 1 TO 100 INPUT Number Sum $\leftarrow$ Sum + Number NEXT // NEXT Count PRINT Sum	3
5(a)	for each field name (1), data type and sample (1) The following are examples there are many different correct answers. - EarTag (1), text, EAR1011 (1) - DOB (1), date, 4/3/2017 (1) - Gender (1), text, M (1) - Weight (1), number, 5.9 (1)	8



Question	Answer			
5(b)	EarTag			
5(c)	Field:	EarTag	Gender	Weight
	Table:	SHEEP	SHEEP	SHEEP
	Sort:			
	Show:	☒	☐	☐
	Criteria:		= 'M'	> 10
	or:			
		(1 mark)	(1 mark)	(1 mark)

QUESTION 5.

0478/21

Question	Answer	Marks
3(a)	One mark for each correct validation check (max two) • Range • Length • Type • Check Digit One mark for each correct related purpose (max two) e.g. • To make sure the data entered falls within a specific set of values • To make sure the data entered is no longer than specified • To make sure the data entered follows rules related to whether it is numbers or letters • To make sure an identification code entered is genuine or possible	4
3(b)	One mark for correct verification check (max one) • Double (data) entry • Visual check	1
3(c)	Any two correct statements (max two) e.g. • Validation checks if the data entered is possible/it cannot check if data has been entered correctly. • Verification checks if the data entered matches the data submitted for entry/ it does not check if data matches set criteria.	2

Question	Answer	Marks
4(a)	Any two correct statements (max two) e.g. • The value of the variable Count begins as 0 ... • ... and is incremented by 1 before it is tested by the loop condition • Count will never be 0 at the end of the loop	2

Question	Answer	Marks
4(b)	<pre> Count ← 0 REPEAT INPUT Number IF Number >= 100 THEN Values[Count] ← Number ENDIF Count ← Count + 1 UNTIL Count = 50 </pre> One mark – separate INPUT statement One mark – IF statement attempted One mark – IF statement completely correct One mark – termination of loop updated	4
4(c)	Any two correct statements (max two) e.g. • Alter the IF statement/add a second IF statement/comparison that's already there ... • ... so that additional criteria set an upper limit of ≤ 200	2

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> One mark for each correct column (max four)	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																																														
5(b)	Any two correct statements e.g. • The program outputs a value • That is divisible by 6 // 2 and 3	2																																												

Question	Answer	Marks
2	Real Integer Char/String String Boolean	5

Question	Answer	Marks
3(a)	One mark for error and correction Line 1 <code>HighestMark ← 0</code> Line 7 <code>INPUT Mark[Count]</code> Line 10 <code>HighestMarkStudents ← HighestMarkStudents + 1</code> Line 14 <code>HighestMark ← Mark[Count]</code>	4
3(b)	Any six from: Add variable <code>LowestMark</code> Set this to a high value for example 100 Add variable <code>LowestMarkStudents</code> Set this to zero Check if <code>Mark[Count] = LowestMark</code> True – add 1 to <code>LowestMarkStudents</code> Check if <code>Mark[Count] < LowestMark</code> True – set <code>LowestMarkStudenta</code> to 1 and set <code>LowestMark</code> to <code>Mark[Count]</code> Add extra output statement	6

Question	Answer	Marks																												
4(a)	One mark for correct input (PointsWon and PointsLost) One mark for correct calculations (Difference) One mark for correct output <table><tr><th>PointsWon</th><th>PointsLost</th><th>Difference</th><th>OUTPUT</th></tr><tr><td>5000</td><td>4474</td><td>526</td><td>Keep on trying</td></tr><tr><td>6055</td><td>2000</td><td>4055</td><td>Well done move up</td></tr><tr><td>7900</td><td>9800</td><td>-1900</td><td>Sorry move down</td></tr><tr><td>3000</td><td>2150</td><td>850</td><td>Keep on trying</td></tr><tr><td>-1</td><td>6700</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>	PointsWon	PointsLost	Difference	OUTPUT	5000	4474	526	Keep on trying	6055	2000	4055	Well done move up	7900	9800	-1900	Sorry move down	3000	2150	850	Keep on trying	-1	6700							3
PointsWon	PointsLost	Difference	OUTPUT																											
5000	4474	526	Keep on trying																											
6055	2000	4055	Well done move up																											
7900	9800	-1900	Sorry move down																											
3000	2150	850	Keep on trying																											
-1	6700																													
4(b)	Any three from: • Add extra decision box ... • ... before checking for difference greater than or equal to 1000 // change Is difference >= 1000 to >= 1000 and <= 5000 • Check for difference greater than 5000 • Add extra Output 'Fantastic leap up two levels'... • ... before flowline returns to input	3																												

QUESTION 7.

0478/23

Question	Answer	Marks
3	Line 2 and Line 4 errors - One mark for each correct identification and correction of error Error 1 line number: Line 2 Correction: REPEAT Error 2 line number: Line 4 Correction: IF Number < 0 OR Number > 499 Line 8 error - One mark for correct identification of error and one mark for each correction of error Error 2 line number: Line 8 Correction: UNTIL Number >= 0 AND Number <= 499	6

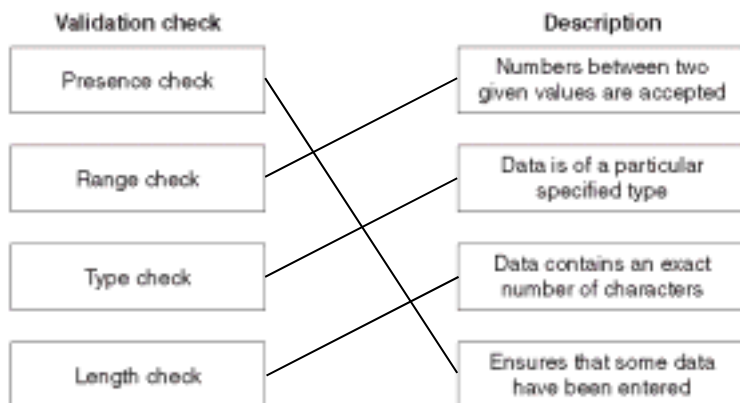
Question	Answer	Marks
4(a)	One mark for each correct check (max two) - Length (check) - Type Check - Format Check	2
4(b)	One mark for each suitable piece of test data and one mark for each relevant reason (max four) - LL9999LL999 - Too long 5678987 - All numeric - CB12EU - No space is present	4

QUESTION 8. Mark Scheme

Cambridge IGCSE – October/November 2016



- 4 1 mark for each correct link up to maximum of 3 marks



[3]

- 5 For each example 1 mark for **correct structure**, 1 mark for **appropriate content** inside loop and 1 mark for **reason**. There are many correct answers these are only samples

```
REPEAT
  ..INPUT Number
  Total ← Total + Number
UNTIL Number = 0
  – at least one repeat is required
```

```
WHILE Number <> -1 DO
  ..INPUT Number
  Total ← Total + Number
ENDWHILE
  – the loop may never be executed
```

[6]

- 6 (a) – Town has duplicate entries/all fields can have duplicate entries
– fields other than Town not suitable identifiers

[2]

- (b) – Performance number ...
– ... uniquely identifies each performance

[2]

QUESTION 9.

Mark Scheme

Cambridge IGCSE – October/November 2016



Section B

2 1 mark for identifying each error, 1 mark for the corresponding change

- line 2 or `Counter = 100`
- `Counter = 0`

- line 6 or `UNTIL Num < 0`
- `UNTIL Num >= 0`

- line 7 or `Total = Total + 1`
- `Total = Total + Num`

- line 8 or `Counter = Counter + Num`
- `Counter = Counter + 1`

[8]

3 Trace table for input value 33

X	A	B	OUTPUT
33	4	1	1
4			4
←	(1 mark)		→ (1 mark)

Trace table for input value 75

X	A	B	OUTPUT
75	9	3	3
9	1	1	1
1			1
←	(1 mark)		→ (1 mark)

[4]



- 4 For each example: **1 mark** for **correct structure**, **1 mark** for **appropriate content**, **the reason**. There are many correct answers these are only examples

```
IF X > 0 AND X <= 10
  THEN PRINT 'In Range'
  ELSE PRINT 'Out of Range'
ENDIF
```

– e.g. checking a condition that may be complex//uses relational operators// checking for a range of values// only 2 options

```
CASE X OF
  1 : PRINT 'Option 1'
  2 : PRINT 'Option 2'
  3 : PRINT 'Option 3'
  OTHERWISE PRINT 'Incorrect choice'
ENDCASE
```

– e.g. checking for discrete/large number/more than 2 of values

[6]

- 5 (a) – 6

[1]

(b)

– Play	text
– No Seats Stalls	number
– Price Stalls Seats \$	currency

[3]

- (c) **1 mark** for **correct plays**, **1 mark** for **correct dates** with each play and no extra fields or text, **1 mark** for the **order**

As You Like It	01/07/2016
Julius Caesar	22/07/2016
Macbeth	14/07/2016

[3]

QUESTION 10.

PUBLISHED

17



Question	Answer
3(c)	1 mark for explanation and 1 mark for an expansion - Library routine is a list of instructions // block of code // subroutine ... that is used often which is given a name ... and which can be called from other programs - Library routines make writing programs easier and faster as the code is already written - Library routines make program testing easier as the code has already been tested and debugged

Question	Answer	Marks
4(a)	1 mark for each correct line Pseudocode description A loop that will iterate at least once. A conditional statement to deal with many possible outcomes. A loop that will iterate a set number of times. A conditional statement with different outcomes for true and false. Pseudocode statement FOR...TO...NEXT IF...THEN...ELSE...ENDIF WHILE...DO...ENDWHILE CASE...OF...OTHERWISE...ENDCASE REPEAT...UNTIL <i>(Note: Lines connect 'A loop that will iterate at least once.' to 'FOR...TO...NEXT' and 'WHILE...DO...ENDWHILE'. 'A conditional statement to deal with many possible outcomes.' connects to 'IF...THEN...ELSE...ENDIF'. 'A loop that will iterate a set number of times.' connects to 'WHILE...DO...ENDWHILE' and 'REPEAT...UNTIL'. 'A conditional statement with different outcomes for true and false.' connects to 'CASE...OF...OTHERWISE...ENDCASE'.)</i>	4
4(b)	1 mark per bullet: - Appropriate loop controls - Read from array - Print from array (the last two points can be in one statement) Note reading and printing MUST be within the same loop Example algorithm: <pre> Count ← 0 WHILE Count < 50 DO OUTPUT Name[Count] Count ← Count + 1 ENDWHILE </pre>	3



Question	Answer						
5(a)	Flag	Count	Name[1]	Name[2]	Name[3]	Name[4]	Temp
			Jamal	Amir	Eve	Tara	
	0	1	Amir	Jamal	Eve	Tara	Jamal
	1	2	Amir	Jamal	Eve	Tara	Jamal
	1	3	Amir	Eve	Jamal	Tara	Jamal
	1	4	Amir	Eve	Jamal	Tara	Jamal
	0	1	Amir	Eve	Jamal	Tara	Jamal
	0	2	Amir	Eve	Jamal	Tara	Jamal
	0	3	Amir	Eve	Jamal	Tara	Jamal
	0	4	Amir	Eve	Jamal	Tara	Jamal
	(1 Mark) (1 Mark) (-----1 Mark-----) (-----1 Mark-----) (1 Mark)						
	5(b)	1 mark per bullet: • Sorting the names • Ascending order / A to Z / lowest to highest / Alphabetic order					
							2

Question	Answer	Marks
6(a)	1 mark for any sensible appropriate field name 1 mark for data type, purpose + example data Example 1: Field Name: SPECIESID Data Type: Alphanumeric Purpose: Primary key Example Data: SP06583 Example 2: Field name: NUMBER Data Type: Integer Purpose: To record how many of that species there are at the park Example Data: 30	2

QUESTION 11.

0478/22

Question	Answer	Marks
3	1 mark for each correct line, max 3 marks. Data Structure Constant Array Table Variable Description A collection of related data. A value that can change whilst a program is running. A value that never changes whilst a program is running. A series of elements of the same data type.	3

Question	Answer	Marks
4	2 marks for identification, 1 mark for description, 1 mark for reason. Identification: <pre> CASE OF ... OTHERWISE ... (ENDCASE) or ... OF ... (OTHERWISE) ... ENDCASE </pre> Description: – a statement that allows for multiple selections // not any of the above Reason: – to simplify pseudocode/ make pseudocode more understandable etc.	4

Question	Answer	Marks																																																																											
5(a)	<table><tr><th>Accept</th><th>Reject</th><th>Count</th><th>Sack</th><th>OUTPUT</th></tr><tr><td>0</td><td>0</td><td>0</td><td></td><td></td></tr><tr><td>1</td><td></td><td>1</td><td>50.4</td><td></td></tr><tr><td>2</td><td></td><td>2</td><td>50.3</td><td></td></tr><tr><td></td><td>1</td><td>3</td><td>49.1</td><td></td></tr><tr><td>3</td><td></td><td>4</td><td>50.3</td><td></td></tr><tr><td>4</td><td></td><td>5</td><td>50.0</td><td></td></tr><tr><td>5</td><td></td><td>6</td><td>49.5</td><td></td></tr><tr><td>6</td><td></td><td>7</td><td>50.2</td><td></td></tr><tr><td>7</td><td></td><td>8</td><td>50.3</td><td></td></tr><tr><td>8</td><td></td><td>9</td><td>50.5</td><td></td></tr><tr><td></td><td>2</td><td>10</td><td>50.6</td><td>8 2</td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table> ← (1 mark) →← (1 mark) →← (1 mark) →← (1 mark) →← (1 mark) →	Accept	Reject	Count	Sack	OUTPUT	0	0	0			1		1	50.4		2		2	50.3			1	3	49.1		3		4	50.3		4		5	50.0		5		6	49.5		6		7	50.2		7		8	50.3		8		9	50.5			2	10	50.6	8 2																5
Accept	Reject	Count	Sack	OUTPUT																																																																									
0	0	0																																																																											
1		1	50.4																																																																										
2		2	50.3																																																																										
	1	3	49.1																																																																										
3		4	50.3																																																																										
4		5	50.0																																																																										
5		6	49.5																																																																										
6		7	50.2																																																																										
7		8	50.3																																																																										
8		9	50.5																																																																										
	2	10	50.6	8 2																																																																									
5(b)	– change to <code>Is Count = 50?</code> – remove <code>IS Sack > 50.5?</code>	2																																																																											



QUESTION 12.

Question	Answer	
3(a)	- Inputs a series of values - Finds the total - Prints out the average	
3(b)	Three from: - Use of loop structure - Allow input to define the limit of the loop / use sentinel value - Keeping a count of the number of values - It could use a totalling process to keep a running total	3
3(c)	Marks awarded as follows (maximum five marks): - Initialise Total - Enter limit - Suitable loop structure - Correct input - Correct totalling - Correct output e.g. <pre>Total ← 0 INPUT CounterLimit FOR LoopCounter ← 1 To CounterLimit INPUT Number Total ← Total + Number NEXT LoopCounter OUTPUT "The average equals ", Total / CounterLimit</pre>	5



Question	Answer											
4(a)	Index	Count	Value	PassMarks								OUTPUT
				[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	
	0											
	0	0	58	58								
	1	1	40									
	1	2	67		67							
	2	3	85			85						
	3	4	12									
	3	5	13									
	3	6	75				75					
	4	7	82					82				
	5											Number passed 5
	1 mark	1 mark	1 mark	1 mark	1 mark							1 mark

4(b)	One from: - Stores numbers greater than 50 in an array - Outputs number of times pass mark has been met - Find the number of pass marks	1
------	--	---

QUESTION 13.

PUBLISHED

19



Question	Answer										
4	<table> <tr> <th>Validation Check</th><th>Description</th></tr> <tr> <td>Range check</td><td>Checks that some data is entered.</td></tr> <tr> <td>Presence check</td><td>Checks for a maximum number of characters in the data entered.</td></tr> <tr> <td>Length check</td><td>Checks that the characters entered are all numbers.</td></tr> <tr> <td>Type check</td><td>Checks that the value entered is between an upper value and a lower value.</td></tr> </table> One mark for each correct line, up to maximum of three marks.	Validation Check	Description	Range check	Checks that some data is entered.	Presence check	Checks for a maximum number of characters in the data entered.	Length check	Checks that the characters entered are all numbers.	Type check	Checks that the value entered is between an upper value and a lower value.
Validation Check	Description										
Range check	Checks that some data is entered.										
Presence check	Checks for a maximum number of characters in the data entered.										
Length check	Checks that the characters entered are all numbers.										
Type check	Checks that the value entered is between an upper value and a lower value.										

Question	Answer	Marks
5	Totalling: - Adding the weight of each basket to the total weight as each weight is entered <code>Total = Total + Weight</code> Counting: - Adding one to/incrementing the number of baskets as each weight is entered <code>BasketCount = BasketCount + 1</code>	4



Question	Answer
6	Constants Two from: - The value cannot be changed accidentally during the execution of the program - Value only needs to be changed once if circumstances change/during the initialisation process Variables Two from: - Stores a value that can change during the execution of the program - Can use a variable without knowing its value Arrays Two from: - A list of items of the same data type stored under a single name - To reduce the number of variables used - Any item can be found using an index number to show its place in the list

Question	Answer	Marks																														
7(a)	Number is repeated/not unique	1																														
7(b)	- Item number not displayed/Amount column not required - Not Like 'Delivered' will also show cancelled items <table><tr><td>Field:</td><td>Item number</td><td>Order number</td><td>Status</td><td></td></tr><tr><td>Table:</td><td>SALES</td><td>SALES</td><td>SALES</td><td></td></tr><tr><td>Sort:</td><td></td><td></td><td></td><td></td></tr><tr><td>Show:</td><td>☒</td><td>☒</td><td>☐</td><td>☐</td></tr><tr><td>Criteria:</td><td></td><td></td><td>Like "Not started"</td><td></td></tr><tr><td>or:</td><td></td><td></td><td>Like "In progress"</td><td></td></tr></table> - Correct Item number column - Correct Order number column and any additional column not shown - Correct status column	Field:	Item number	Order number	Status		Table:	SALES	SALES	SALES		Sort:					Show:	☒	☒	☐	☐	Criteria:			Like "Not started"		or:			Like "In progress"		5
Field:	Item number	Order number	Status																													
Table:	SALES	SALES	SALES																													
Sort:																																
Show:	☒	☒	☐	☐																												
Criteria:			Like "Not started"																													
or:			Like "In progress"																													

QUESTION 14.

PUBLISHED

19



Question	Answer
2	Two from - Sub-program / system not the whole program / system - To perform a frequently used operation within a program - That can be called when needed - That can be reused by another program

Question	Answer	Marks
3	- FOR (... TO ... NEXT) loop - WHILE (... DO ... ENDWHILE) loop - REPEAT (... UNTIL) loop	3

Question	Answer	Marks
4(a)	Conditional / selection statement	1
4(b)	Four from: MP1 CASE statement with identifier Response MP2 Correct structure used for choices... MP3 correct statements used for choices MP4 OTHERWISE and correct statement MP5 Single ENDCASE included e.g. CASE OF Response // CASE Response OF 1 : X ← X + Y 2 : X ← X - Y 3 : X ← X * Y 4 : X ← X / Y OTHERWISE OUTPUT "No response" ENDCASE	4

Question	Answer						Marks
5(a)	First	Last	UserIn	Middle	Found	OUTPUT	6
	0	16			FALSE		
	0	16	10	8	FALSE		
	0	7	10	3	FALSE		
	4	7	10	5	TRUE	TRUE	
	One mark per correct column						