



--

--	--	--	--	--

--	--	--	--

0478/23

May/June 2019

1 hour 45 minutes

No Additional Materials are required.

No calculators allowed.

READ THESE INSTRUCTIONS FIRST

Write in dark blue or black pen.

You may use an HB pencil for any diagrams, graphs or rough working.

Do not use staples, paper clips, glue or correction fluid.

DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.

DO NOT ATTEMPT TASKS 1, 2 AND 3 in the pre-release material; these are for information only.

You are advised to spend no more than **40 minutes** on **Section A** (Question 1).

No marks will be awarded for using brand names of software packages or hardware.

Any businesses described in this paper are entirely fictitious.

At the end of the examination, fasten all your work securely together.

The number of marks is given in brackets [] at the end of each question or part question.

The maximum number of marks is 50.

This syllabus is regulated for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

This document consists of **15** printed pages and **1** blank page.

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 discount stationery store sells items in bulk at a discount to its customers. Customers can order from a small range of items available on that day. The greater the quantity of an item they buy in one order, the greater the discount for that item.

Quantity of an item	Discount
10	5%
20	10%
50	25%
100	35%
500	50%

In addition to this, new customers can use a \$10 voucher, provided their order comes to more than \$50. Also, returning customers are given a discount voucher with a monetary value.

Write and test a program or programs for the discount stationery store.

- 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 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 – Stationery available on the day.

Using arrays, set up a list of at least 10 different items available that day. For every item include: item code, description, price and the amount in stock. All item codes must be different. Display the item code, description, price and the amount in stock on the screen.

Task 2 – Customer places an order.

The customer chooses an item code from the list of currently available items. Any items that are out of stock are not displayed and any items that have low stock of 10 or fewer are highlighted. The customer enters the number they wish to purchase. If there are not enough items in stock the number is rejected. The price to pay and any discount is shown on the screen. When the order is complete, any customer discount voucher is applied and the total price is shown. If the customer confirms the order, the stock numbers are then updated.

Task 3 – At the end of the day.

List the item codes of any items that have sold out. Display the item code of any items of which none have been sold. Display the item code and description of the item of which the greatest quantity have been sold today.

- 1 (a) All variables, constants and other identifiers must have meaningful names.

Describe the data structures you have used in **Task 1** to record the items available that day. Include some sample data for each data structure you have described.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

- (b) Explain how your program for **Task 1** ensures all the item codes are different.

.....

.....

.....

..... [2]

- (c) Explain how your program for **Task 2** highlights items where the stock level is low.

.....

.....

.....

.....

.....

..... [3]

- (d) Write an algorithm for **Task 2** using **either** pseudocode, programming statements **or** a flowchart. Assume that **Task 1** has been completed and the list of items currently available has already been displayed.

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.

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

..... [5]

Question 2 begins on page 8.

Section B

- 2 (a) An algorithm has been written in pseudocode to input the weight of 500 items and reject any that are over-weight or under-weight, then print the percentage rejected.

```

Count ← 1
Reject ← 0
Over ← 62
Under ← 58
REPEAT
    INPUT ItemWeight
    IF ItemWeight > Over AND ItemWeight < Under
        THEN
            Reject ← Reject - 1
        ENDIF
    Count ← Count + 1
UNTIL Count > = 500
Reject ← Reject / 100
PRINT "Percentage rejected is ", Reject

```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1

.....

Correction

.....

Error 2

.....

Correction

.....

Error 3

.....

Correction

.....

Error 4

.....

Correction

.....

[4]

- (b) Describe how you would change the corrected algorithm to calculate the number accepted instead of rejected, using a variable `Accept`, and print a warning if fewer than 50% are accepted.

.....

.....

.....

.....

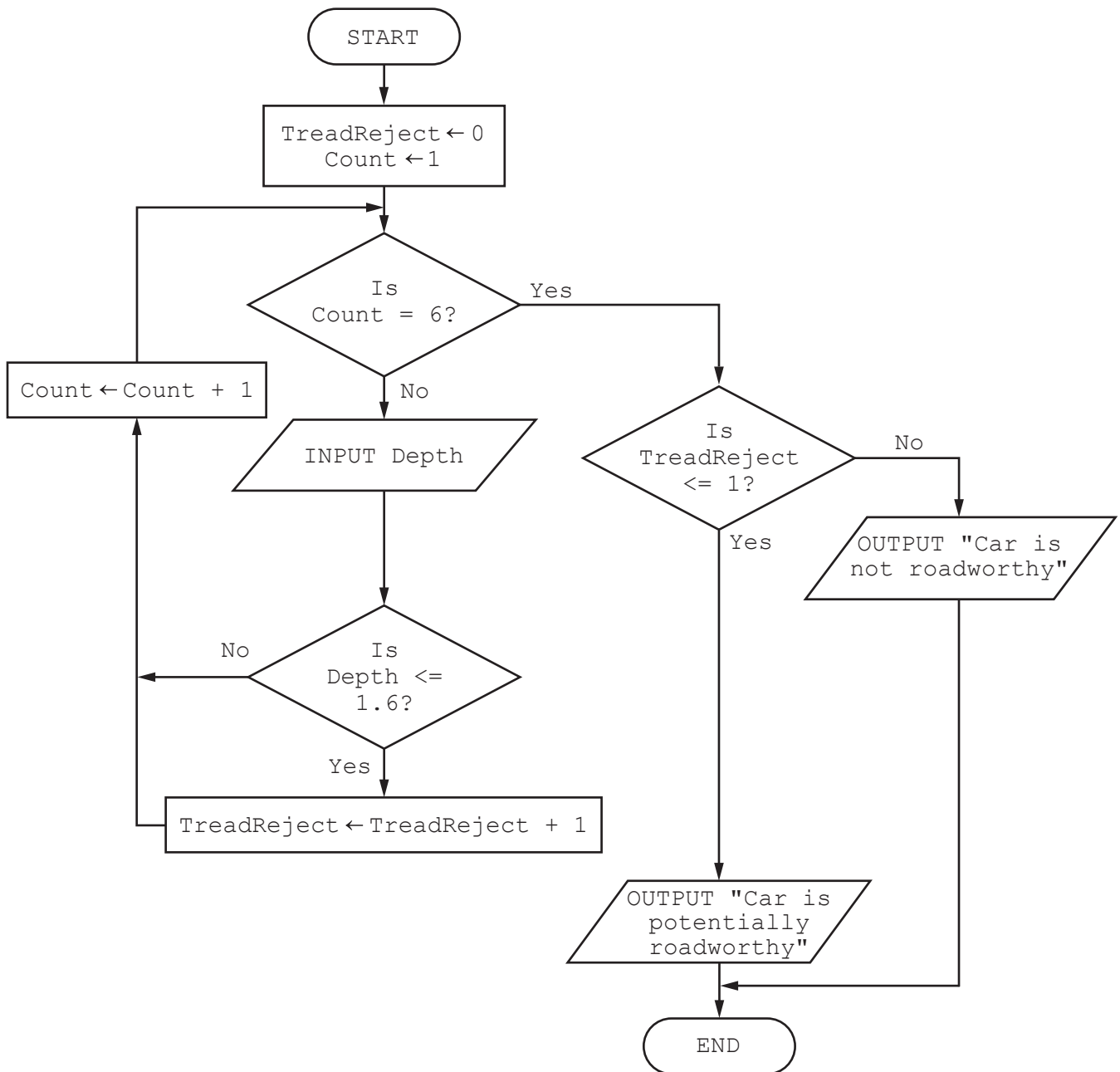
.....

.....

.....

..... [4]

- 3 This flowchart inputs the tread depth of five tyres, four on the car and a spare tyre. Any tread depth of 1.6 mm or less is rejected. To be potentially roadworthy, a car must have four tyres with a tread depth greater than 1.6 mm.



Complete Trace table 1 for the tread depth input data:
1.7, 1.9, 1.4, 1.8, 2.0

TreadReject	Count	Depth	OUTPUT

Trace table 1

Complete Trace table 2 for the tread depth input data:
1.2, 1.9, 1.4, 1.8, 2.4

TreadReject	Count	Depth	OUTPUT

Trace table 2

[4]

- 4 For each of the **four** checks in the table, place a tick in the correct column to show whether it is an example of a **validation** or **verification** check.

Statements	Validation	Verification
Range check		
Double entry		
Check digit		
Presence check		

[4]

- 5 Identify and describe **three** loop structures that are available in pseudocode.

Loop structure 1

.....

Description

.....

.....

Loop structure 2

.....

Description

.....

.....

Loop structure 3

.....

Description

.....

.....

[6]

Question 6 begins on page 14.

- 6 A database table, TRAIN, is used to keep a record of train departures from a station.

Train Number	Platform	Destination	Departure Time	Status
1A37	1	Newtown	08:00	On time
2X19	2	Anytown	08:10	Late
1A29	1	Bigcity	08:15	On time
1A28	2	Anytown	08:30	Cancelled
1A67	3	Gardenvillage	08:45	On time
1A37	1	Newtown	08:50	On time
1A24	2	Charter Train	09:00	Late
1A67	3	Gardenvillage	09:15	On time

- (a) Explain why the field **Train Number** could not be used as a primary key.

.....
 [1]

- (b) A query-by-example has been written to display only the train numbers and platforms of all trains leaving after 08:30 that are late.

Field:	Train Number	Platform	Departure Time	Status
Table:	TRAIN	TRAIN	TRAIN	TRAIN
Sort:				
Show:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:		= Y	< 08:30	
or:				= "Late"

Explain why the query-by-example is incorrect, and write a correct query-by-example.

Explanation

.....

.....

.....

.....

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

[7]

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 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.



COMPUTER SCIENCE

0478/23

Paper 2

May/June 2019

MARK SCHEME

Maximum Mark: 50

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 International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the May/June 2019 series for most Cambridge IGCSE™, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

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
Section A		
1(a)	<p>4 marks for: Data Structure(s) max 2</p> <ul style="list-style-type: none"> • Arrays • Variable(s) for // data entry such as Item Code // index / constant for minimum number of items <p>Further description max 3</p> <ul style="list-style-type: none"> • Data type(s) one or more • Use(s) one or more • Name(s) one or more e.g. Item_Code, Description, Price, Stock • Sample data for appropriate arrays e.g. 1234, Notepad, 0.050, 1000 // Sample data for variable or constant e.g. 10 <p>1 mark for:</p> <ul style="list-style-type: none"> • All four names and suitable data samples 	5
1(b)	<ul style="list-style-type: none"> • Keep a counter/number • Add one every time a new item code is added <p>or</p> <ul style="list-style-type: none"> • Keep a list of item codes used • Using a loop check item codes is not already in the list before a new item code is added 	2
1(c)	<ul style="list-style-type: none"> • Check items in stock array • ... with a value of 10 or less • Display these values in a different way e.g. in red 	3

Question	Answer	Marks
1(d)	<p>Five from:</p> <p>MP1 Enter the item code</p> <p>MP2 ...and check if valid</p> <p>MP3 Enter number to purchase and check if less than or equal to number in stock for that item number</p> <p>MP4 Display error message(s) if needed</p> <p>MP5 Calculate price to pay</p> <p>MP6 Using the number to purchase calculate discount ...</p> <p>MP7 Check for new customer AND \$50 spend/returning customer eligibility ...</p> <p>MP8 ... application of discount(s) if available</p> <p>MP9 Check order confirmed ...</p> <p>MP10 ... update appropriate stock level</p> <p>Sample answer</p> <pre> REPEAT PRINT "Enter Item Code" INPUT ItemCode Found ← False Count ← 1 REPEAT IF ItemCode = ItemList(Count) THEN Found ← True ENDIF Count ← Count + 1 UNTIL Found OR Count = NoItems UNTIL Found REPEAT PRINT "Enter Number to purchase" INPUT Number UNTIL Number > 0 and Number <= Stock(ItemCode) PriceToPay ← Number * Price (ItemCode) IF Number >= 10 AND Number < 20 THEN PriceToPay ← PriceToPay * 0.95 ENDIF IF Number >= 20 AND Number < 50 THEN PriceToPay ← PriceToPay * 0.9 ENDIF IF Number >= 50 AND Number < 100 THEN PriceToPay ← PriceToPay * 0.75 ENDIF IF Number >= 100 AND Number < 500 THEN PriceToPay ← PriceToPay * 0.65 ENDIF IF Number >= 500 THEN PriceToPay ← PriceToPay * 0.5 ENDIF PRINT "Enter Value of discount voucher, 0 if no voucher" INPUT Voucher PriceToPay ← PriceToPay - Voucher PRINT "Price to pay is ", PriceToPay PRINT "Order confirmed? Y/N" INPUT Confirm IF Confirm = "Y" THEN Stock(ItemCode) ← Stock(ItemCode) - Number ENDIF </pre>	5

Question	Answer	Marks
1(e)	<p>Explanation</p> <p>Five from:</p> <p>MP1 Search item arrays</p> <p>MP2 For stock level = 0 ...</p> <p>MP3 ... list these as sold out</p> <p>MP4 Check stock level at end of day with original stock level//search items sold arrays</p> <p>MP5 ... if the same, display/store item code as not sold//if 0, display/store item code as not sold</p> <p>MP6 ... update largest value if greater than existing greatest value ... //use max function on items sold array ... //other method to find highest number sold ...</p> <p>MP7 ... display this item number with a suitable message</p> <p>All programming statements used must be explained.</p>	5

Question	Answer	Marks
Section B		
2(a)	<p>1 mark for each error identified + suggested correction</p> <p>Count \leftarrow 1 should be Count \leftarrow 0 or Count \geq 500 should be Count $>$ 500 AND should be OR Reject \leftarrow Reject - 1 should be Reject \leftarrow Reject + 1 Reject \leftarrow Reject/100 should be Reject \leftarrow Reject/5 or Reject \leftarrow 100 / 500</p>	4
2(b)	<p>MP1 Add Accept \leftarrow 0 at start</p> <p>MP2 Add ELSE Accept \leftarrow Accept + 1 after THEN <u>AND</u> Over and Under defined/position described OR Add Accept \leftarrow Accept + 1 after THEN <u>AND</u> Replace IF statement with ...\leq Over AND ...\geq UNDER... /position described</p> <p>MP3 Add Accept \leftarrow Accept/5 after UNTIL AND correct loop/position described</p> <p>MP4 ADD IF Accept $<$ 50 THEN PRINT "Less than 50% accepted" at end</p> <pre> Accept \leftarrow 0 Count \leftarrow 1 // 0 Reject \leftarrow 0 Over \leftarrow 62 Under \leftarrow 58 REPEAT INPUT ItemWeight IF ItemWeight $>$ Over OR ItemWeight $<$ Under // IF ItemWeight \leq Over AND ItemWeight \geq Under THEN Reject \leftarrow Reject + 1 ELSE Accept \leftarrow Accept + 1//ELSE not required ENDIF Count \leftarrow Count + 1 UNTIL Count $>$ 500 // \geq 500 Accept \leftarrow Accept / 5 IF Accept $<$ 50 THEN PRINT "Less than 50% accepted" ENDIF </pre>	4

Question	Answer				Marks																																								
3	<table><tr><th>TreadReject</th><th>Count</th><th>Depth</th><th>OUTPUT</th></tr><tr><td>0</td><td>1</td><td></td><td></td></tr><tr><td></td><td>2</td><td>1.7</td><td></td></tr><tr><td></td><td>3</td><td>1.9</td><td></td></tr><tr><td>1</td><td>4</td><td>1.4</td><td></td></tr><tr><td></td><td>5</td><td>1.8</td><td></td></tr><tr><td></td><td>6</td><td>2.0</td><td></td></tr><tr><td></td><td></td><td></td><td>Car is potentially roadworthy</td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>				TreadReject	Count	Depth	OUTPUT	0	1				2	1.7			3	1.9		1	4	1.4			5	1.8			6	2.0					Car is potentially roadworthy									4
	TreadReject	Count	Depth	OUTPUT																																									
	0	1																																											
		2	1.7																																										
		3	1.9																																										
	1	4	1.4																																										
		5	1.8																																										
		6	2.0																																										
				Car is potentially roadworthy																																									
	<table><tr><th>TreadReject</th><th>Count</th><th>Depth</th><th>OUTPUT</th></tr><tr><td>0</td><td>1</td><td></td><td></td></tr><tr><td>1</td><td>2</td><td>1.2</td><td></td></tr><tr><td></td><td>3</td><td>1.9</td><td></td></tr><tr><td>2</td><td>4</td><td>1.4</td><td></td></tr><tr><td></td><td>5</td><td>1.8</td><td></td></tr><tr><td></td><td>6</td><td>2.4</td><td></td></tr><tr><td></td><td></td><td></td><td>Car is not roadworthy</td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>				TreadReject	Count	Depth	OUTPUT	0	1			1	2	1.2			3	1.9		2	4	1.4			5	1.8			6	2.4					Car is not roadworthy									
	TreadReject	Count	Depth	OUTPUT																																									
	0	1																																											
	1	2	1.2																																										
		3	1.9																																										
	2	4	1.4																																										
		5	1.8																																										
		6	2.4																																										
				Car is not roadworthy																																									
1 mark for each correct pairs of columns.																																													

Question	Answer	Marks															
4	<table border="1"> <thead> <tr> <th>Statements</th><th>Validation</th><th>Verification</th></tr> </thead> <tbody> <tr> <td>Range check</td><td>✓</td><td></td></tr> <tr> <td>Double entry</td><td></td><td>✓</td></tr> <tr> <td>Check digit</td><td>✓</td><td></td></tr> <tr> <td>Presence check</td><td>✓</td><td></td></tr> </tbody> </table> <p>1 mark for each correct row</p>	Statements	Validation	Verification	Range check	✓		Double entry		✓	Check digit	✓		Presence check	✓		4
Statements	Validation	Verification															
Range check	✓																
Double entry		✓															
Check digit	✓																
Presence check	✓																

Question	Answer	Marks
5	<ul style="list-style-type: none"> • FOR ... TO ... NEXT • fixed number of repetitions • REPEAT ... UNTIL • always executed // condition tested at end • WHILE ... DO ... ENDWHILE • may not be executed // condition tested at beginning 	6

Question	Answer	Marks																														
6(a)	Number is repeated/not unique	1																														
6(b)	<p>Three from: Train number not displayed Departure time before 8:30 Criteria of =Y for Platform not required/incorrect All late trains will be shown/the condition should be on the line above</p> <p>4 marks, one mark for each correct column</p> <table><tr><td>Field:</td><td>Train Number</td><td>Platform</td><td>Departure Time</td><td>Status</td></tr><tr><td>Table:</td><td>TRAIN</td><td>TRAIN</td><td>TRAIN</td><td>TRAIN</td></tr><tr><td>Sort:</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 type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>Criteria:</td><td></td><td></td><td>> 08:30</td><td>= "Late"</td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td></tr></table>	Field:	Train Number	Platform	Departure Time	Status	Table:	TRAIN	TRAIN	TRAIN	TRAIN	Sort:					Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Criteria:			> 08:30	= "Late"	or:					7
Field:	Train Number	Platform	Departure Time	Status																												
Table:	TRAIN	TRAIN	TRAIN	TRAIN																												
Sort:																																
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																												
Criteria:			> 08:30	= "Late"																												
or:																																