



COMPUTER SCIENCE

0478/22

Paper 2

October/November 2017

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 October/November 2017 series for most Cambridge IGCSE[®], Cambridge International A and AS Level components and some Cambridge O Level components.

© IGCSE is a registered trademark.

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

This document consists of **8** printed pages.



Cambridge Assessment
International Education

Question	Answer	Marks
1(a)(i)	1 mark for appropriate variable name, 1 mark for appropriate data type, 1 mark for appropriate use. Many correct answers, they must be meaningful. These are examples only. – HireTotal, integer, running total of money taken (for the day) – HoursHired, real, running total of hours hired for the day – Returned, real, hour and fraction of hour when next returned	3
1(a)(ii)	1 mark for appropriate constant name, 1 mark for appropriate value. Many correct answers, they must be meaningful. These are examples only. – HourPrice, 20.00 – HalfHourPrice 12.00	2
1(b)	1 mark for validation check, all checks must be different, 1 mark for the reason and 1 mark for the test data. The only inputs for task 1 can be length of hire, money taken, time of hire and time of return. There are many possible correct answers these are examples only. Validation check – range check for time of hire Reason – cannot be hired before 10:00 returned after 17:00 Test data – 12:00, 19:00 Validation check – type check for money taken Reason – must be a numeric value Test data – 20.00, bob	6

Question	Answer	Marks
1(c)	– any loop for 10 boats (1 mark) Four from: – Initialisation – check HoursHired against MaxHoursHired store the BoatNumber ... update MaxHoursHired if greater – check if HoursHired = 0 if so add 1 to NumberBoatsUnused – update daily totals (for hours and money) – output report with messages (including totals for hours and money, and number of boats unused and the most used boat). Max 4 marks Example: <pre> MaxHoursHired ← 0 TotalHoursHired ← 0 TotalMoney ← 0 NumberBoatsUnused ← 0 FOR BoatNumber ← 1 to 10 TotalMoney ← TotalMoney + Money(BoatNumber) TotalHoursHired ← TotalHoursHired + HoursHired(BoatNumber) IF HoursHired(BoatNumber) = 0 THEN NumberBoatsUnused ← NumberBoatsUnused + 1 ENDIF IF HoursHired(BoatNumber) > MaxHoursHired THEN MostUsed ← BoatNumber MaxHoursHired ← HoursHired(BoatNumber) ENDIF NEXT BoatNumber PRINT "Boats were hired for ", TotalHoursHired, " hours" PRINT "Total amount of money taken was ", TotalMoney PRINT NumberBoatsUnused, " boats were not used" Print "Boat number ", MostUsed, " was used most"</pre>	5

Question	Answer	Marks
1(d)	Maximum 4 marks in total for question part e.g. Explanation (may include reference to program statements) – check all boats for... – ... return time < current time // current booking slot available or return time > current time// current booking slot not available – keep a running total of those available – display number of boats Example: <pre> FOR BoatNumber ← 1 to 10 loop to check for all boats IF ReturnTime(BoatNumber) <= CurrentTime check return time against current time THEN BoatsAvailable ← BoatsAvailable + 1 keep a running total ENDIF NEXT BoatNumber PRINT "Number of boats available ", BoatsAvailable display number of boats </pre>	4

Question	Answer	Marks
2	1 mark for each, there may be other solutions, award full marks for any working solution any six from: - initialise total (outside loop) - Input number of numbers (outside loop with validation) - Loop using input value - Input number (inside loop) - Update Total (inside loop) - Calculate average - Print average and total (outside loop) Sample algorithm: <pre> INPUT NumberCount Total ← 0 FOR Count ← 1 TO NumberCount INPUT Number Total ← Total + Number NEXT Average ← Total/NumberCount PRINT Total, Average </pre>	6

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)	Accept	Reject	Count	Sack	OUTPUT	5
	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	
← (1 mark) →← (1 mark) →← (1 mark) →← (1 mark) →← (1 mark) →						
5(b)	– change to Is Count = 50? – remove IS Sack > 50.5?					2

Question	Answer	Marks																														
6(a)	– 1 mark for each field suitable name, 1 mark for appropriate data type and appropriate data sample The following are examples there are many different correct answers. – Engine Number, text, 21012 – Class, text, P6 – Service Date, date, 4/3/2017	6																														
6(b)	– Engine Number // Correct field number	1																														
6(c)	<table><tr><td>Field:</td><td>Engine Number</td><td>Class</td><td>Service Date</td><td></td></tr><tr><td>Table:</td><td>TRAIN</td><td>TRAIN</td><td>TRAIN</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>Like 'P*' // Like 'P?'</td><td><10/11/2016</td><td></td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td></tr></table> (1 mark) (1 mark) (1 mark)	Field:	Engine Number	Class	Service Date		Table:	TRAIN	TRAIN	TRAIN		Sort:					Show:	☒	☐	☐	☐	Criteria:		Like 'P*' // Like 'P?'	<10/11/2016		or:					3
Field:	Engine Number	Class	Service Date																													
Table:	TRAIN	TRAIN	TRAIN																													
Sort:																																
Show:	☒	☐	☐	☐																												
Criteria:		Like 'P*' // Like 'P?'	<10/11/2016																													
or:																																