Cambridge Assessment International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/23**

Paper 2 **May/June 2021**

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 2021 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

© UCLES 2021

[Turn over

PUBLISHED

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.

© UCLES 2021

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.

© UCLES 2021

Please note the following further points:

The words in **bold** in the mark scheme are important text that needs to be present, or some notion of it needs to be present. It does not have to be the exact word, but something close to the meaning.

If a word is underlined, this **exact** word must be present.

A single forward slash means this is an alternative word. A double forward slash means that this is an alternative mark point.

Ellipsis (…) on the end of one-mark point and the start of the next means that the candidate **cannot** get the second mark point without being awarded the first one. If a MP has ellipsis at the beginning, but there is no ellipsis on the MP before it, then this is just a follow-on sentence and **can** be awarded **without** the previous mark point.

© UCLES 2021

Question	Answer	Marks
	Section A	
1(a)(i)	**One** mark per bullet point • Constant `NumberofOptions` • Value 5 • Use Storing the maximum number of options (for the referendum)	**3**
1(a)(ii)	**One** mark per bullet point • Variable `UniqueNumber` • Use Storing/inputting the voter's unique identification number	**2**
1(a)(iii)	**One** mark for each valid mark point (**Max 4**) • Suitable name e.g. `IdNumber` • Suitable data type e.g. Integer • Suitable array length e.g. 170 (elements) • Suitable sample data e.g. 100 • Suitable use e.g. to store the unique numbers	**4**

© UCLES 2021

Question	Answer	Marks

1(b)	Any **six** from:
	MP1 Initialisation of totalling variables
	MP2 Loop to cycle through all the data
	MP3 Totalling the votes for at least one of the options
	MP4 Totalling the votes for all five options
	MP5 Separate totals for students and staff
	MP6 Output to show one set of results
	MP7 Output to show more than one set of results with messages
	MP8 Comprehensive output to show student, staff and combined results, with messages
	
	Example answers
	Version 1
	`TotalAS ← 0; TotalBS ← 0; TotalCS ← 0; TotalDS ← 0; TotalsES ← 0; TotalAF ← 0;`
	`TotalBF ← 0; TotalCF ← 0; TotalDF ← 0; TotalEF ← 0`
	`FOR Count ← 0 TO 150`
	`  TotalAS ← TotalAS + StudentA[Count]`
	`  TotalBS ← TotalBS + StudentB[Count]`
	`  TotalCS ← TotalCS + StudentC[Count]`
	`  TotalDS ← TotalDS + StudentD[Count]`
	`  TotalES ← TotalES + StudentE[Count]`
	`NEXT Count`
	`FOR Count = 0 TO 20`
	`  TotalAF ← TotalAF + StaffA [Count]`
	`  TotalBF ← TotalBF + StaffA [Count]`
	`  TotalCF ← TotalCF + StaffA [Count]`
	`  TotalDF ← TotalDF + StaffA [Count]`
	`  TotalEF ← TotalEF + StaffA [Count]`
	`NEXT Count`
	`OUTPUT "Option  Students  Staff  Total"`
	`OUTPUT "A ", TotalAS, TotalAF, (TotalAS + TotalAF)`
	`OUTPUT "B ", TotalBS, TotalBF, (TotalBS + TotalBF)`
	`OUTPUT "C ", TotalCS, TotalCF, (TotalCS + TotalCF)`
	`OUTPUT "D ", TotalDS, TotalDF, (TotalDS + TotalDF)`
	`OUTPUT "E ", TotalES, TotalEF, (TotalES + TotalEF)`
		6

Question	Answer	Marks

1(b)	**Version 2**
	`TotalAS ← 0; TotalBS ← 0; TotalCS ← 0; TotalDS ← 0; TotalsES ← 0; TotalAF ← 0;`
	`TotalBF ← 0; TotalCF ← 0; TotalDF ← 0; TotalEF ← 0`
	`OUTPUT "Are you a student? (Y or N)"`
	`INPUT Student`
	`IF Student = "Y" THEN`
	`  OUTPUT "Enter your number"`
	`  INPUT Number`
	`  IF Students[Number] = 0 THEN`
	`    REPEAT`
	`      OUTPUT "Enter preference for A"`
	`      INPUT Vote`
	`    UNTIL Vote >= 1 and Vote <= 5`
	`    TotalAS ← TotalAS + Vote`
	`    REPEAT`
	`      OUTPUT "Enter preference for B"`
	`      INPUT Vote`
	`    UNTIL Vote >= 1 and Vote <= 5`
	`    TotalBS ← TotalBS + Vote`
	`    REPEAT`
	`      OUTPUT "Enter preference for C"`
	`      INPUT Vote`
	`    UNTIL Vote >= 1 and Vote <= 5`
	`    TotalCS ← TotalCS + Vote`
	`    REPEAT`
	`      OUTPUT "Enter preference for D"`
	`      INPUT Vote`
	`    UNTIL Vote >= 1 and Vote <= 5`
	`    TotalDS ← TotalDS + Vote`
	`    REPEAT`
	`      OUTPUT "Enter preference for E"`
	`      INPUT Vote`
	`    UNTIL Vote >= 1 and Vote <= 5`
	`    TotalES ← TotalES + Vote`	

© UCLES 2021

Question	Answer	Marks

1(b)	Student[Number] ← 1
	ELSE
	OUTPUT "Enter your number"
	INPUT Number
	IF staff[number] = 0 THEN
	REPEAT
	OUTPUT "Enter preference for A"
	INPUT Vote
	UNTIL Vote >= 1 and Vote <= 5
	TotalAS ← TotalAS + Vote
	REPEAT
	OUTPUT "Enter preference for B"
	INPUT Vote
	UNTIL Vote >= 1 and Vote <= 5
	TotalBS ← TotalBS + Vote
	REPEAT
	OUTPUT "Enter preference for C"
	INPUT Vote
	UNTIL Vote >= 1 and Vote <= 5
	TotalCS ← TotalCS + Vote
	REPEAT
	OUTPUT "Enter preference for D"
	INPUT Vote
	UNTIL Vote >= 1 and Vote <= 5
	TotalDS ← TotalDS + Vote
	REPEAT
	OUTPUT "Enter preference for E"
	INPUT Vote
	UNTIL Vote >= 1 and Vote <= 5
	TotalES ← TotalES + Vote
	Staff[Number] ← 1
	ENDIF	

Question	Answer	Marks

1(b)	`OUTPUT "Option  Students  Staff  Total"`
	`OUTPUT "A ", TotalAS, TotalAF, (TotalAS + TotalAF)`
	`OUTPUT "B ", TotalBS, TotalBF, (TotalBS + TotalBF)`
	`OUTPUT "C ", TotalCS, TotalCF, (TotalCS + TotalCF)`
	`OUTPUT "D ", TotalDS, TotalDF, (TotalDS + TotalDF)`
	`OUTPUT "E ", TotalES, TotalEF, (TotalES + TotalEF)`	

1(c)	Any **two** from: MP1 Change the maximum preferences constant to 6 MP2 Change the input prompt to show six options MP3 Change the loop counter MP4 Change the number of inputs for each person to six MP5 Add another array/variable to store/total the new option votes	**2**
1(d)	Explanation Any **three** from: MP1 Initialise count variables/array for each option MP2 Using a conditional statement to identify preference 1 in at least one option (in the input/stored data) MP3 … identify preference 1 in all options MP4 ... and adding 1 to the appropriate option count in at least one option MP5 … for students only MP6 … using a loop to cycle through the whole array/set of inputs for each option	**3**

© UCLES 2021

Question	Answer	Marks
	Section B	
2	**Three** marks for three or four correct **Two** marks for two correct **One** mark for one correct **Data Type** → **Description** Real → May be any number String → May contain any combination of characters Integer → Must be a whole number Boolean → Must be one of two values	**3**

Question	Answer	Marks
3	**One** mark per mark point for each piece of test data (3 × two marks) • Correct validation check name • Correct use identified **Example answers** **ld27@cambridgeuniversity.com** • Validation check name — Length (check) • Use — Counts the number of characters in the data to make sure it isn't too long (max length 320 characters). **2021** • Validation check name — Range (check) • Use — Checks that the number entered fits within given parameters **Ericson-Bower** • Validation check name — Type (check) • Use — Checks the type of data entered (in this case) to make sure no numbers are present	**6**

© UCLES 2021

Question	Answer	Marks

4(a)	**One** mark for error identified and suggested correction
	
	Line 3 – should be `FullScore ← 0`
	Line 7 – should be `FullScore ← FullScore + Score`
	Line 8 – should be `NEXT Allow ENDFOR //` alternatively Line 5 could be `REPEAT` with `StoreLoop ← 0` just above it and `StoreLoop ← StoreLoop + 1` between lines 7 and 8.
	Line 11 – should be `INPUT Another`
	
	Correct Algorithm 1
	`1 Count ← 0`
	`2 REPEAT`
	`3 FullScore ← 0`
	`4 INPUT Number`
	`5 FOR StoreLoop ← 1 TO Number`
	`6 INPUT Score`
	`7 FullScore ← FullScore + Score`
	`8 NEXT`
	`9 OUTPUT "The full score is ", FullScore`
	`10 OUTPUT "Another set of scores (Y or N)?"`
	`11 INPUT Another`
	`12 IF Another = "N"`
	`13 THEN`
	`14 Count ← 1`
	`15 ENDIF`
	`16 UNTIL Count = 1`
		4

Question	Answer	Marks
4(a)	**Correct Algorithm 2** 1 Count ← 0 2 REPEAT 3 FullScore ← 0 4 INPUT Number StoreLoop ← 0 5 REPEAT 6 INPUT Score 7 FullScore ← FullScore + Score StoreLoop ← StoreLoop + 1 8 UNTIL StoreLoop = Number 9 OUTPUT "The full score is ", FullScore 10 OUTPUT "Another set of scores (Y or N)?" 11 INPUT Another 12 IF Another = "N" 13 THEN 14 Count ← 1 15 ENDIF 16 UNTIL Count = 1	

Question	Answer	Marks

4(b)	**One** mark per mark point (**Max 4**)
	
	MP1 After line 6 // replace line 6
	MP2 `ScoreArray[StoreLoop] ← Score` //
	`INPUT ScoreArray[StoreLoop]`
	
	MP3 between lines 8 and 10
	MP4 `AverageScore ← FullScore/Number`
	MP5 `OUTPUT "The average score is ", AverageScore`
	
	Example correct algorithm for reference from part 4(a)
	
	`1 Count ← 0`
	`2 REPEAT`
	`3 FullScore ← 0`
	`4 INPUT Number`
	`5 FOR StoreLoop ← 1 TO Number`
	`6 INPUT Score`
	`7 FullScore ← FullScore + Score`
	`8 NEXT`
	`9 OUTPUT "The full score is ", FullScore`
	`10 OUTPUT "Another set of scores (Y or N)?"`
	`11 INPUT Another`
	`12 IF Another = "N"`
	`13 THEN`
	`14 Count ← 1`
	`15 ENDIF`
	`16 UNTIL Count = 1`
		4

Question	Answer	Marks
5(a)	**One** mark for each correct column	**5**

Op	Value1	Value2	Ans	OUTPUT
1				
	87			
		14	101	
3				
	2			
		30	60	
5				
	10			
		6		Input Error
4				
	10			
		2	5	
0				

Question	Answer	Marks
5(b)	To work as a calculator // to add, subtract, multiply or divide a pair of numbers	1
5(c)	To output/store the result/the value of `Ans` // Adding prompts for data entry.	1

Question	Answer	Marks
6(a)	**Two** marks for six correct field names **One** mark for at least three correct field names	2

Field name
LicenceNo
Mileage
TyreFLft
TyreFRgt
TyreRLft
TyreRRgt

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(b)</td><td>
One mark for correct fieldnames
One mark for correct table names and show fields
One mark for correct sort
One mark for correct search criteria in all columns
<table>
<tr><td>Field:</td><td>LicenceNo</td><td>Mileage</td><td>TyreFLft</td><td>TyreFRgt</td><td>TyreRLft</td><td>TyreRRgt</td></tr>
<tr><td>Table:</td><td>TREAD</td><td>TREAD</td><td>TREAD</td><td>TREAD</td><td>TREAD</td><td>TREAD</td></tr>
<tr><td>Sort:</td><td>Ascending</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Show:</td><td>☑</td><td>☑</td><td>☑</td><td>☑</td><td>☑</td><td>☑</td></tr>
<tr><td>Criteria:</td><td></td><td></td><td><2</td><td><2</td><td><2</td><td><2</td></tr>
<tr><td>or:</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td><td>4</td></tr>
</table>

© UCLES 2021