Cambridge Assessment International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/22**

Paper 2 **February/March 2022**

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 February/March 2022 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

© UCLES 2022

[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 2022

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 2022

Question	Answer	Marks
	Section A	
1(a)	Many correct answers, the names used must be meaningful. The names given are examples only. **One** mark per mark point, max **three** Constant name `MaxNumberPlayers` Value 4 Why constant used This number will not change whilst the program is running **One** mark per mark point, max **three** Variable name `NumberOfPlayers` Data type Integer/int Why variable used A value between 2 and 4 inclusive is input and stored at the start of the round	**6**
1(b)	**One** mark per mark point, max **four** MP1 using a `FOR … NEXT` // `REPEAT … UNTIL` // `DO … WHILE` loop MP2 starting at 1/0 and finishing at 18/17 **or** 9/8 // 18/9 iterations MP3 setting the elements in the player scores arrays to zero MP4 … for all four arrays // number of arrays for players in this round MP5 setting variables / array for total scores to 0 MP6 use of assignment / append statement(s) Any programming statements included must be explained.	**4**

© UCLES 2022

Question	Answer	Marks

1(c)	**One** mark per mark point, max **six**
	
	MP1 loop through the number of holes played
	MP2 for each hole work / loop through the number of players actually playing
	MP3 for each player display their name
	MP4 … prompt to enter the number of strokes played for the hole
	MP5 … input the number of strokes twice
	MP6 … validate both inputs are the same
	MP7 … store the number of strokes in the appropriate array
	MP8 … update the total score for that player
	MP9 … prompt and input to ask if the player wants to see the number of strokes played so far
	MP10 … check if required then output number of strokes
	
	Example Answer
	`FOR Hole ← 1 TO NumberOfHoles`
	`  FOR Player ← 1 TO NumberOfPlayers`
	`  REPEAT`
	`  OUTPUT PlayerName[Player], "Please enter the number of strokes played for hole ",`
	`  Hole`
	`  INPUT NumberStrokes`
	`  OUTPUT PlayerName[Player], " please re-enter the number"`
	`  INPUT NumberStrokesAgain`
	`  UNTIL NumberStrokes = NumberStrokesAgain`
	`  IF Player = 1`
	`  THEN`
	`  Player1[Hole] ← NumberStrokes`
	`  Player1Total ← Player1Total + NumberStrokes`
	`  OUTPUT "Do you want to see number total of strokes played so far Y/N? "`
	`  INPUT SeeTotal`
	`  IF SeeTotal = "Y"`
	`  THEN`
	`  OUTPUT "Total number of strokes so far ", Player1Total`
	`  ENDIF`
	`  ENDIF`
	`  IF Player = 2`
		6

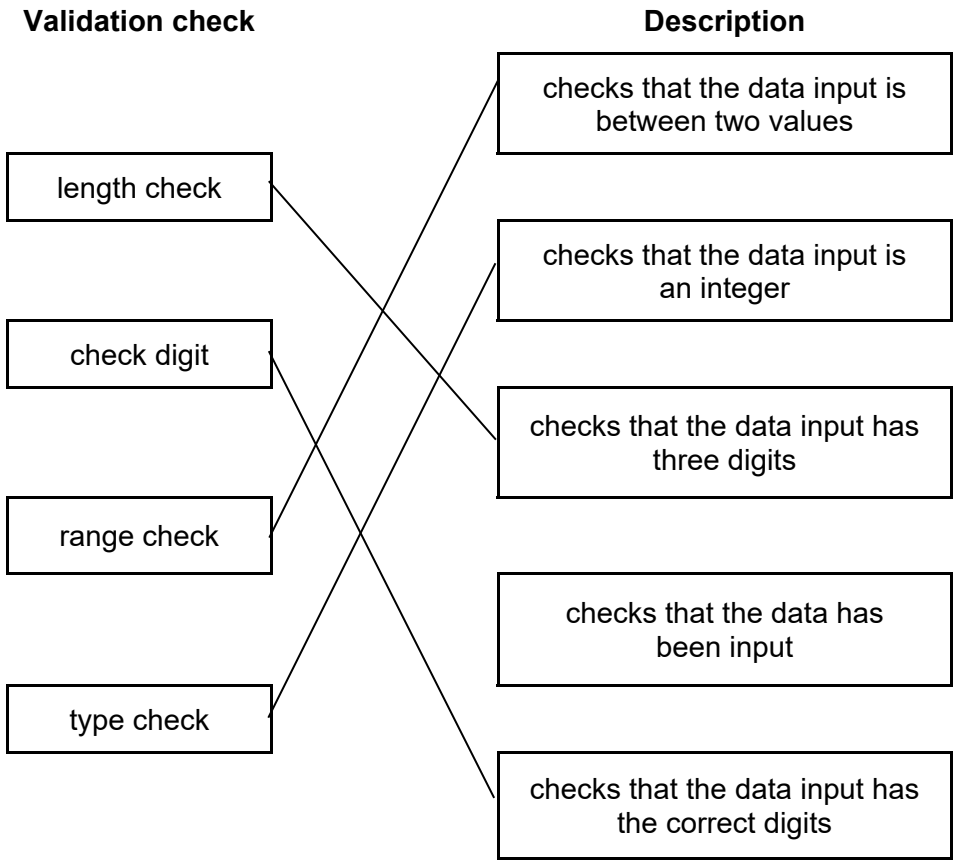
Question	Answer	Marks
1(c)	THEN Player2[Hole] ← NumberStrokes Player2Total ← Player2Total + NumberStrokes OUTPUT "Do you want to see number total of strokes played so far Y/N? " INPUT SeeTotal IF SeeTotal = "Y" THEN OUTPUT "Total number of strokes so far ", Player2Total ENDIF ENDIF IF Player = 3 THEN Player3[Hole] ← NumberStrokes Player3Total ← Player3Total + NumberStrokes OUTPUT "Do you want to see number total of strokes played so far Y/N? " INPUT SeeTotal IF SeeTotal = "Y" THEN OUTPUT "Total number of strokes so far ", Player3Total ENDIF ENDIF IF Player = 4 THEN Player4[Hole] ← NumberStrokes Player4Total ← Player4Total + NumberStrokes OUTPUT "Do you want to see number total of strokes played so far Y/N? " INPUT SeeTotal IF SeeTotal = "Y" THEN OUTPUT "Total number of strokes so far ", Player4Total ENDIF ENDIF NEXT Player NEXT Hole	

Question	Answer	Marks
1(d)	Explanation **One** mark per mark point, max **four** MP1 Work/ Loop through all the total scores MP2 compare each total score using selection / `IF` statements // use an appropriate function MP3 select the score with the **lowest** value MP4 … also select the player name for that score MP5 output the player name and **either** the difference between par and their score **or** their score MP6 compare the score with the lowest value with the value of par using selection / `IF` statements MP7 … identify as "over par" if the value is greater than par or output "under par" if the value is less than par or output "par" if there is no difference Any programming statements included must be explained.	**4**

Question	Answer	Marks
	Section B	
2(a)	**One** mark per mark point, max **four** • `100` • `AND Age < 12` • `Count12to18 + 1` • `CountOver18`	**4**

2(b)	**One** mark suitable IF construct, **one** mark correct assignment statement, for example
	`IF Age < 7`
	`  THEN`
	`    CountUnder7 ← CountUnder7 + 1`
	`ENDIF`
	One mark suitable message, **one** mark correct use of `countUnder7` variable, for example
	`OUTPUT "There are ", CountUnder7, " students aged under 7."`
		4

Question	Answer	Marks
3	**One** mark for each correct **single** line from the validation check, max **four**	**4**



<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>4(a)</td><td>One mark each for columns Number and OUTPUT
Two marks for column C first four values (1) last three values (1)
Two marks for column D first six values (1) last four values (1)

<table>
<tr><th>Number</th><th>C</th><th>D</th><th>OUTPUT</th></tr>
<tr><td>7</td><td>0</td><td>3</td><td></td></tr>
<tr><td></td><td></td><td>2</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
<tr><td>6</td><td>0</td><td>3</td><td></td></tr>
<tr><td></td><td>1</td><td>2</td><td></td></tr>
<tr><td></td><td>2</td><td>1</td><td>2</td></tr>
<tr><td>5</td><td>0</td><td>2</td><td></td></tr>
<tr><td></td><td></td><td>1</td><td></td></tr>
<tr><td>4</td><td>0</td><td>2</td><td></td></tr>
<tr><td></td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>-1</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td></tr>
</table>
</td><td>6</td></tr>
<tr><td>4(b)</td><td>One mark per mark point, max two
• to count the factors / the numbers that go into (other than 1 or itself) of a number
• to output the number of factors</td><td>2</td></tr>
</table>

© UCLES 2022

Question	Answer	Marks
4(c)(i)	**One** mark per mark point, max **two** • the value of D becomes zero • division by zero error • endless loop	**2**
4(c)(ii)	**One** mark per mark point, max **two** • after the decision box to test if the number is -1 • insert another decision box to test if the number is less than 4 / less than or equal to 3 • return to INPUT Number if true	**2**

Question	Answer	Marks
5	Explanation **One** mark per mark point, max **three** • field, FlowerID, not required / should not be displayed • Type field not included and displayed • Fragrance field should not be displayed • Fragrance criteria should not be Y / should be N	**6**

Field:	Type	Fragrance	Style	Colour
Table:	FLOWER	FLOWER	FLOWER	FLOWER
Sort:				
Show:	☑	☐	☑	☑
Criteria:		= N		
or:				

query-by-example grid
One mark per mark point, max **three**

- One mark for changing Flower ID to Type
- One mark for changing Criteria in Fragrance to N
- One mark for changing Show in Fragrance to ☐

© UCLES 2022