



CANDIDATE
NAME

--

CENTRE
NUMBER

--	--	--	--	--

CANDIDATE
NUMBER

--	--	--	--

0478/23

October/November 2021

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

- Answer all questions.
- **Do not attempt Tasks 1, 2 and 3** in the copy of the pre-release material on page 2; these are for information only.
- Use a black or dark blue pen. You may use an HB pencil for any diagrams or graphs.
- Write your name, centre number and candidate number in the boxes at the top of the page.
- Write your answer to each question in the space provided.
- Do **not** use an erasable pen or correction fluid.
- Do **not** write on any bar codes.
- Calculators must **not** be used in this paper.

- The total mark for this paper is 50.
- The number of marks for each question or part question is shown in brackets [].
- No marks will be awarded for using brand names of software packages or hardware.

This document has **16** pages. Any blank pages are indicated.

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 following tasks before the examination to answer Question 1.

Pre-release material

A cruise ship has a speciality restaurant where tables can be booked for any of the three sessions: lunch, early dinner or late dinner. A booking for a table can only be made on the day. There are twenty tables available to book in the restaurant. Today's date and how many tables are available for lunch, early dinner and late dinner are displayed on a screen at the entrance to the restaurant.

Only one table can be booked at a time. When a booking is made, the name of the passenger making the booking is recorded, together with their cabin number and any special dietary requirements.

Write and test a program or programs for a computer system to manage the daily restaurant bookings.

- 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 – at the start of the day

Write a program to set up the screen display at the start of the day showing the date and how many tables are available for lunch, early dinner and late dinner. Bookings are to be stored in three separate arrays: lunch, early dinner and late dinner. Initialise further arrays to record for each table booked: the name of the passenger making the booking, their cabin number and any special dietary requirements.

Task 2 – making a table booking at the restaurant

Check if there is a table available for the session requested. If a table is available, record the passenger's name, cabin number and any special dietary requirements. Mark the table as booked for that session. Display the name and cabin number for the passenger to check. Update the screen display and mark a session as fully booked if all the tables are now booked.

Task 3 – special dietary requirements

The recording of special dietary requirements is confusing the restaurant staff. It has been decided to use fixed options instead of a description. Only one option can be chosen for each booking.

The options are:

- gluten-free
- vegetarian
- vegan
- diabetic
- none.

Update **Task 2** to allow for this.

Update **Task 2** to count and display how many tables have vegetarian or vegan diners during the day.

1 All variables, constants and other identifiers must have meaningful names.

- (a) Identify **one** constant that you could have used for **Task 1**. Give the value that would be assigned to this constant. State the use of this constant.

Constant

Value

Use

..... [3]

- (b) Describe the arrays that you have set up in **Task 1** to record today's data about the restaurant tables.

.....

 [4]

- (c) Sometimes the restaurant has fewer tables available for a session.

Explain how you would change your **Task 1** program to allow input of the number of tables available for a session.

..... [3]

- (d) Part of **Task 2** is to check if a table is available. Write an algorithm to do this, using either pseudocode, programming statements or a flowchart. Assume that **Task 1** has been completed.

This image shows a single sheet of white paper with horizontal ruling lines. 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 for **Task 3** updated **Task 2**.

Include any programming statements used and fully explain the purpose of each statement.

[4]

Section B starts on Page 8.

Section B

- 2 An algorithm has been written in pseudocode to generate 50 positive random integers with values less than or equal to 100. These numbers are stored in the array NumRand[]

The function RandUp(X, Y) generates a random integer greater than X and less than or equal to Y . For example, RandUp(1, 4) generates 2 or 3 or 4

```

1 Count ← 0
2 WHILE Counter > 50 DO
3     NumRand[Counter] ← RandUp(1, 100)
4     Counter ← Counter - 2
5 ENDWHILE

```

- (a) Find the **four** errors in the pseudocode and write a correction for each error.

Error 1

Correction

.....

Error 2

Correction

.....

Error 3

Correction

.....

Error 4

Correction

.....

.....

[4]

- (b) The pseudocode for this algorithm could be shortened by the use of a FOR ... NEXT loop.

Rewrite the algorithm using a FOR ... NEXT loop.

.....

.....

.....

..... [3]

- 3 A program has been written to check the length and content of a password. The password must be eight or more characters long and contain at least one special character, for example, `Secret!*!`

- (a) (i) State suitable examples of normal and erroneous test data that could be used to test this program. For each example give the reason for your choice of test data.

Normal test data example

Reason

.....

Erroneous test data example

Reason

.....

[4]

- (ii) Explain why two pieces of boundary test data are required for this program. Give an example of each piece of boundary test data.

.....

.....

.....

.....

..... [3]

- (b) Describe **two** methods of verification that could be used to verify this data as it is input.

Method 1

.....

.....

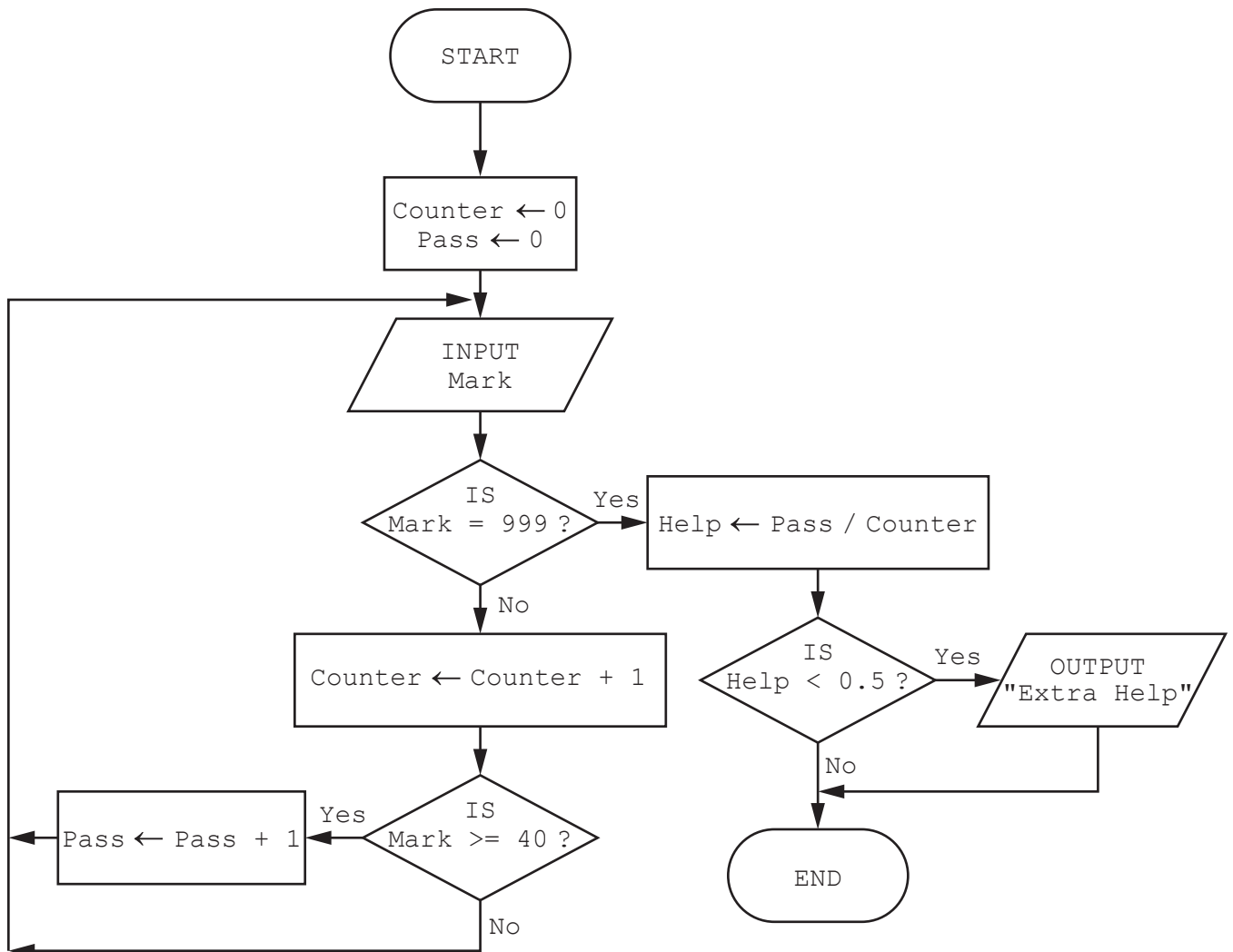
Method 2

.....

.....

[4]

- 4 The algorithm, shown by this flowchart, allows the input of examination marks for a class of students. A mark of 999 ends the process. If a mark is 40 or over then a pass grade is awarded. The number of pass grades is calculated for the whole class. If this is under 50% of the class, the class is offered extra help.



Complete a trace table for the algorithm using this input data:
88, 24, 60, 30, 44, 17, 25, 22, 54, 6, 999, -1

Counter	Pass	Mark	Help	OUTPUT

[5]

- 5 A database table, COMPUTER, is used to keep a record of computers available for sale.

The following data is stored for each computer:

- CATEGORY – desktop, laptop or tablet
- WEIGHT – weight in kilograms
- MANUFACTURER – ICN, Linoldo, Pear or JoeSing
- PRICE – price in \$
- CODE – a unique code allocated by the manufacturer, e.g. P771
- STOCK – quantity in stock.

A database management system uses these data types:

Text Number Currency Boolean

The CATEGORY field and MANUFACTURER field have a data type of text.

- (a) Select the most appropriate data type for each field from the **four** types shown. State the reason why you chose the data type.

WEIGHT data type

Reason

.....

PRICE data type

Reason

.....

CODE data type

Reason

.....

STOCK data type

Reason

.....

[4]

- (b) Complete the query-by-example grid to display only the category, manufacturer, price and code of the computers with weight of less than 2.5 kilograms.

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

[3]

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.



Cambridge IGCSE™

COMPUTER SCIENCE

0478/23

Paper 2

October/November 2021

MARK SCHEME

Maximum Mark: 50

<p>Published</p>

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

This document consists of **9** printed pages.

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.

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)	Constant name <code>MaxNoTables</code> Value <code>20</code> Use Storing the maximum number of tables available for a session	3
1(b)	Any three from: MP1 Identifier / name of array used MP2 Description of purpose of an identified array MP3 Length of an identified array used MP4 Type of data in an identified array MP5 Explanation of number of arrays used, must be capable of storing all data required MP6 Sample data for an identified array One mark MP7 Identifying more than one array E.g. 3 sets of 4 arrays of twenty elements for each session, for example for lunch, <code>TableLunch</code> of type Boolean, <code>PassengerLunch</code> , <code>CabinLunch</code> and <code>DietReqLunch</code> all type string	4
1(c)	Any three from: MP1 Input the maximum number of tables available for a session MP2 Input the maximum number of tables available for every session MP3 Storing each value input in a variable / an array MP4 Validation check MP5 Change the constant used for number of tables to a variable MP6 Using the value input instead of 20 for max value for loop counter etc	3

Question	Answer	Marks
1(d)	<p>Any six from:</p> <p>MP1 Input session</p> <p>MP2 ... with prompt</p> <p>MP3 Validate input</p> <p>MP4 Check number of tables available / check each table ...</p> <p>MP5 ... for the session that has been input</p> <p>MP6 If no tables available any session that has been input</p> <p>MP7 ... output suitable message</p> <p>Sample</p> <pre> REPEAT OUTPUT "Which session do you want to book a table?" INPUT Session UNTIL Session >= 1 AND Session <= 3 IF TablesAvailable[Session] = 0 THEN OUTPUT "No tables available " ENDIF </pre>	6
1(e)	<p>Explanation</p> <p>Any four from:</p> <p>MP1 Any changes required for the Array data types in Task 1</p> <p>MP2 How the program displayed the options in Task 3</p> <p>MP3 How the program selected the choice in Task 3</p> <p>MP4 How the program dealt with incorrect choices in Task 3</p> <p>MP5 How the program counted number of tables with vegetarian diners</p> <p>MP6 How the program counted number of tables with vegan diners</p> <p>MP7 How the program output number of tables with vegetarian/vegan diners</p> <p>Programming statements should be used and must be explained.</p>	4

Question	Answer	Marks
Section B		
2(a)	<p>Line 1 should be Counter</p> <p>Line 2 should be < 50</p> <p>Line 3 should be RandUp(0,100) // RandUp(-1,100)</p> <p>Line 4 should be Counter ← Counter + 1</p> <pre> 1 Counter ← 0 2 WHILE Counter < 50 DO 3 NumRand[Counter] ← RandUp(0, 100) 4 Counter ← Counter + 1 5 ENDWHILE </pre>	4
2(b)	<p>Any three from:</p> <pre> FOR Counter ← 0 TO 49 // FOR Counter ← 1 TO 50 NumRand[Counter] ← RandUp(0,100) / RandUp(-1,100) NEXT // NEXT Counter </pre>	3

Question	Answer	Marks
3(a)(i)	<p>one mark for sample, one mark for reason max four</p> <p>Normal Sample any password with at least 8 characters and one special character e.g. Password!</p> <p>Reason to test that normal data is accepted and processed correctly</p> <p>Erroneous Sample any value that would be rejected e.g. secret</p> <p>Reason to test that erroneous data is rejected</p>	4
3(a)(ii)	<p>Reason to test that Secret? which has 7 characters is rejected and Secret?? which has 8 characters is accepted</p> <p>Boundary Sample 1 – Secret?</p> <p>Boundary Sample 2 – Secret??</p>	3

Question	Answer	Marks
3(b)	Any two methods, one mark for method, one mark for description max four E.g. Asking the user to enter the password twice and comparing the values (1) only accepting the data if both entries are identical (1) Displaying the password as it is entered (1) so the user can put right errors have been made as the password was typed (1)	4

Question	Answer	Marks																																																																	
4	<p>One mark for each correct column</p> <table><tr><th>Counter</th><th>Pass</th><th>Mark</th><th>Help</th><th>OUTPUT</th></tr><tr><td>0</td><td>0</td><td></td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>88</td><td></td><td></td></tr><tr><td>2</td><td></td><td>24</td><td></td><td></td></tr><tr><td>3</td><td>2</td><td>60</td><td></td><td></td></tr><tr><td>4</td><td></td><td>30</td><td></td><td></td></tr><tr><td>5</td><td>3</td><td>44</td><td></td><td></td></tr><tr><td>6</td><td></td><td>17</td><td></td><td></td></tr><tr><td>7</td><td></td><td>25</td><td></td><td></td></tr><tr><td>8</td><td></td><td>22</td><td></td><td></td></tr><tr><td>9</td><td>4</td><td>54</td><td></td><td></td></tr><tr><td>10</td><td></td><td>6</td><td></td><td></td></tr><tr><td></td><td></td><td>999</td><td>0.4</td><td>Extra Help</td></tr></table>	Counter	Pass	Mark	Help	OUTPUT	0	0				1	1	88			2		24			3	2	60			4		30			5	3	44			6		17			7		25			8		22			9	4	54			10		6					999	0.4	Extra Help	5
Counter	Pass	Mark	Help	OUTPUT																																																															
0	0																																																																		
1	1	88																																																																	
2		24																																																																	
3	2	60																																																																	
4		30																																																																	
5	3	44																																																																	
6		17																																																																	
7		25																																																																	
8		22																																																																	
9	4	54																																																																	
10		6																																																																	
		999	0.4	Extra Help																																																															

Question	Answer	Marks
5(a)	<p>WEIGHT – Number, comparisons / calculations may be required</p> <p>PRICE – Currency, the price is in dollars / money</p> <p>CODE – Text, no calculations required, could be numbers or characters</p> <p>STOCK – Number, comparisons / calculations may be required</p>	4

Question	Answer					Marks	
5(b)	Field:	CATEGORY	MANUFACTURER	PRICE	CODE	WEIGHT	3
	Table:	COMPUTER	COMPUTER	COMPUTER	COMPUTER	COMPUTER	
	Sort:						
	Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Criteria:					< 2.5	
	or:						
	<p>One mark for correct rows Field and Table</p> <p>One mark for correct Show row</p> <p>One mark for correct Criteria row</p>						