



CANDIDATE
NAME

--

CENTRE
NUMBER

--	--	--	--	--

CANDIDATE
NUMBER

--	--	--	--

0478/21

October/November 2022

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 program is needed for a swimming club to manage the annual audit of its membership details.

The data stored about each member includes name, age, gender, type of membership, whether a team member or **not**, annual fee and if the fee has been paid. Team members are allowed a discount of 10% on their annual membership fee.

The types of membership available are:

Membership	Age range in years	Annual fee
junior	≥ 2 and < 18	\$10.00
adult	≥ 18 and < 50	\$20.00
senior	≥ 50 and < 80	\$15.00
golden	80 and over	free

Write and test a program or programs for the swimming club annual audit:

- Your program or programs must include appropriate prompts for the entry of data. Data must be validated on entry.
- All outputs, including error messages, 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 – setting up the system to store the membership details

Set up your program to:

- store the members' details in arrays
- populate the arrays with at least 20 members' details
- use the array index as the annual membership number.

Task 2 – providing annual audit statistics

Count the number of current members by membership type. For each type of membership count the number of members who did **not** pay the annual fee. Display these counts as a percentage of the total number of members for each membership type. Calculate and display the total of the annual fees expected and the annual fees received for this year.

Task 3 – updating the membership details for the next year

Extend your program to:

- check if any members have **not** paid and output a list of all these members
 - remove these members from the system
- update the age of all the members by one year and whether they are in a team
 - update the type of membership and the annual fee if required
- set the fee for every member as **not** yet paid
- display lists of current team members grouped by membership type.

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

- (a) (i) Describe the arrays you have set up in **Task 1** to record the members' details. Include in your description the name, data type and sample data for each array.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

- (ii) Explain how your program for **Task 1** populated the arrays with the members' details.

.....

.....

.....

.....

.....

.....

..... [2]

- (b) Explain how your program removed the members that have **not** paid their fees from the system in **Task 3**.

.....

.....

.....

..... [2]

- (c) Write an algorithm for the part of **Task 3** that checks the age of all the members, and whether they are in a team, then updates the type of membership and the annual fee if required.

Use pseudocode, programming statements or a flowchart.

[illegible]

(d) Explain how your program completes these parts of **Task 2**:

- Count the number of members of each membership type who did **not** pay the annual fee.
- Display these counts as a percentage of the total number of members for each membership type.

Any programming statements that you include in your answer must be fully explained.

..... [5]

Section B starts on page 8.

Section B

2 An algorithm has been written to:

- set 100 elements of the array `Reading[1:100]` to zero
- input integer values between 1 and 100
- end the process with an input of `-1`
- reject all other values
- count and output the number of times each value is input, starting with the largest value.

(a) Complete the pseudocode algorithm:

```

01  FOR Count ← 1 TO .....
02      Reading[Count] ← 0
03  NEXT Count
04  OUTPUT "Please enter next reading "
05  INPUT Value
06  WHILE Value <> -1 DO
07      IF Value <= 0 OR .....
08          THEN
09              OUTPUT "Reading out of range"
10          ELSE
11              Reading[Value] ← .....
12          ENDIF
13      OUTPUT "Please enter next reading "
14      .....
15  ENDWHILE
16  Count ← 100
17  REPEAT
18      OUTPUT "There are ", ..... ,
          " readings of ", Count
19      Count ← .....
20  UNTIL Count = 0

```

[6]

(b) Describe how the algorithm could be changed so that it does **not** output any counts of zero.

.....

.....

.....

.....

.....

..... [3]

- 3 (a) A PIN (personal identification number) is input into a banking app by the user. Before the PIN is accepted, the following validation checks are performed:

- check 1 – each character must be a digit
- check 2 – there must be exactly four digits
- check 3 – the value of the PIN must be between 1000 and 9999 inclusive.

Describe each validation check.

Check 1

.....

.....

.....

Check 2

.....

.....

.....

Check 3

.....

.....

.....

[6]

- (b) The PIN can be changed by the user.

Describe how the new PIN could be verified before use.

.....

.....

.....

.....

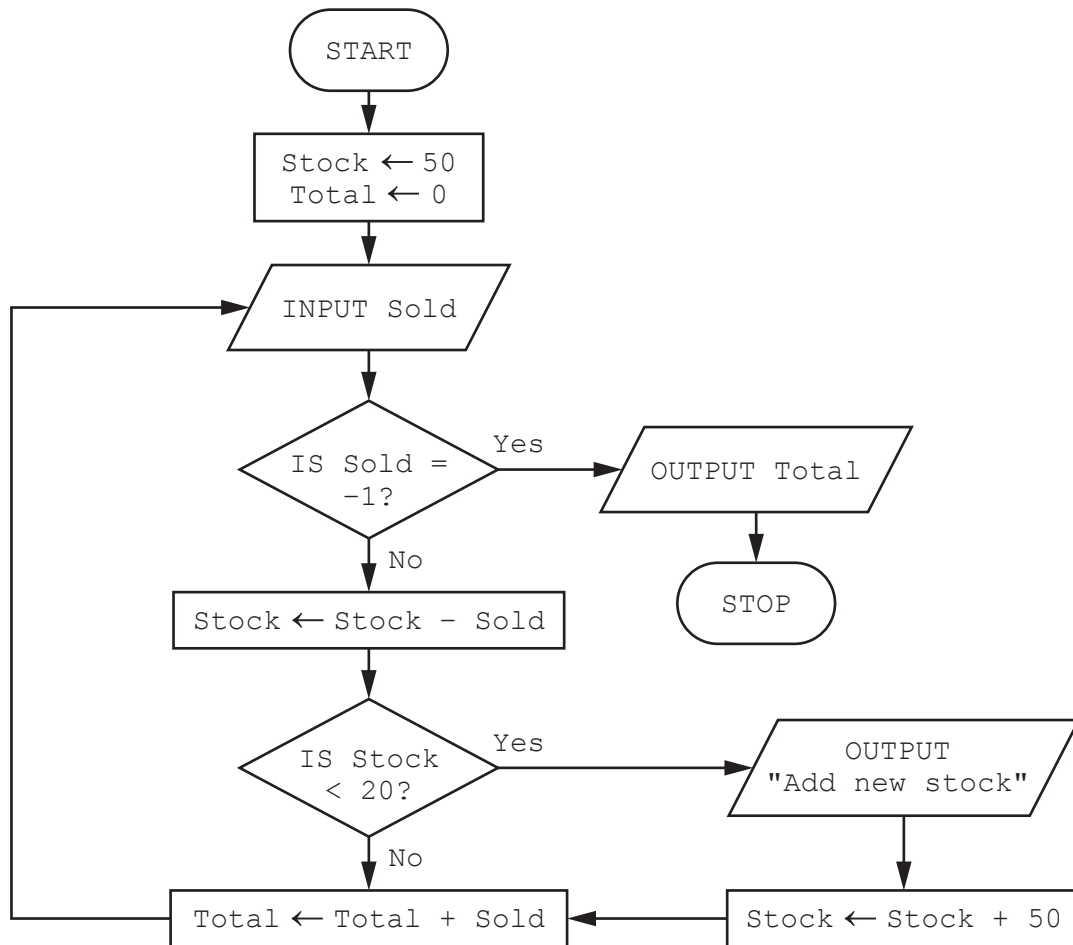
.....

.....

[3]

Question 4 starts on page 12.

- 4 This algorithm makes sure that there are enough fresh bread rolls available for customers to buy.



- (a) Complete the trace table for the algorithm using this input data:
24, 12, 6, 30, 12, 18, -1, 24

Sold	Stock	Total	OUTPUT

[4]

- (b) Identify the problem that will occur if the input data starts with a value of 70.
Explain how you would correct this problem.

.....

.....

.....

.....

.....

.....

.....

..... [3]

- 5 A database table, WAREHOUSE, is used to keep a record of items for sale. The table has these fields:

- ItemCode – code to identify each type of item
- Description – brief description of each item
- Manufacturer – name of manufacturer
- Level – number in stock
- Price – price in dollars (\$).

- (a) State which field you would choose for the primary key. Give a reason for your choice.

Field

Reason

..... [2]

- (b) Complete the query-by-example grid to display only the item code and the manufacturer where the number in stock is below 10.

Field:				
Table:				
Sort:				
Show:	<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 Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.



Cambridge IGCSE™

COMPUTER SCIENCE

0478/21

Paper 2 Problem-solving and Programming

October/November 2022

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 2022 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.

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 mark point has an ellipsis at the beginning, but there is no ellipsis on the mark point before it, then this is just a follow-on sentence and **can** be awarded **without** the previous mark point.

Question	Answer	Marks
	Section A	
1(a)(i)	<p>Many correct answers, the names used must be meaningful. The names given are examples only.</p> <p>One mark per mark point, max five</p> <p>MP1 name one e.g. Name, Age, Gender, Type, TeamMember, AnnualFee, Paid MP2 one appropriate data type must match name e.g. string, integer, char, string, Boolean, real, Boolean MP3 sample data for one appropriate array MP4 all arrays mentioned have appropriate names MP5 all arrays mentioned have appropriate data types MP6 sample data for all arrays e.g. Sue, 9, F, Junior, True, 9.00, False</p>	5
1(a)(ii)	<p>One mark per mark point, max two</p> <p>MP1 use a loop e.g. <code>FOR...NEXT</code> / <code>REPEAT...UNTIL</code> / <code>WHILE...DO</code> MP2 to input the values for each element in every array MP3 append each value input to the end of the array</p>	2
1(b)	<p>One mark per mark point, max two</p> <p>MP1 use a loop/ <code>FOR</code>/<code>REPEAT</code>/<code>WHILE</code> to check all members MP2 use a conditional statement / <code>IF</code> to see if the paid field is marked as False MP3 only transfer the data of those members whose paid field is marked as True to a set of new arrays // delete the details of the members who have not paid</p>	2

Question	Answer	Marks
1(c)	<p>One mark per mark point, max six</p> <p>MP1 loop through all the members MP2 any suitable correct comparison of age MP3 if 18 Change membership type to adult MP4 if 50 Change membership type to senior MP5 if 80 Change membership type to golden MP6 check if team member MP7 no store appropriate full fee MP8 yes store 90% of appropriate full fee</p> <p>Example</p> <pre> FOR Member ← 1 TO Total CASE Age[Member] OF 18 : Type[Member] ← "Adult" 50 : Type[Member] ← "Senior" 80 : Type[Member] ← "Golden" ENDCASE IF Team[Member] THEN CASE Type[Member] OF "Junior" : Fee[Member] ← JuniorFee * 0.9 "Adult" : Fee[Member] ← AdultFee * 0.9 "Senior" : Fee[Member] ← SeniorFee * 0.9 ENDCASE ELSE CASE Type[Member] OF "Junior" : Fee[Member] ← JuniorFee "Adult" : Fee[Member] ← AdultFee "Senior" : Fee[Member] ← SeniorFee "Golden" : Fee[Member] ← GoldenFee ENDCASE ENDIF NEXT Member </pre>	6

Question	Answer	Marks
1(d)	<p>Explanation</p> <p>One mark per mark point, max five</p> <p>MP1 how the program checked all the members // use of loop MP2 how the program identified a member who has not paid their (annual) fee // use of condition MP3 how the program kept a running total of members who have not paid their (annual) fees MP4 for each of the three types of membership that require a fee // excluding golden members MP5 how the program calculated a percentage for members who had not paid... MP6 for each of the three types of membership that require a fee MP7 how the program displayed a percentage for members who had not paid MP8 all three percentages displayed with suitable messages</p> <p>Programming statements when used must be explained.</p>	5

Question	Answer	Marks
	Section B	
2(a)	<p>One mark per mark point, max six</p> <ul style="list-style-type: none"> • Line 1 100 • Line 7 Value > 100 // Value >= 101 • Line 11 Reading[Value] + 1 • Line 14 INPUT Value • Line 18 Reading[Count] • Line 19 Count - 1 	6

Question	Answer	Marks
2(b)	<p>One mark per mark point, max three</p> <ul style="list-style-type: none"> • use an IF/conditional statement • to check if Reading[Count] not equal to zero • before outputting the value // between statements 17 and 18 // code sample showing position <pre>IF Reading[Count] <> 0 THEN OUTPUT ENDIF</pre>	3

Question	Answer	Marks
3(a)	<p>Two marks per check, description must match name of check if given, max six</p> <ul style="list-style-type: none"> • Check 1 use a type check <ul style="list-style-type: none"> ○ to ensure that the value is a number / integer • Check 2 use a length check <ul style="list-style-type: none"> ○ to ensure that there are only 4 characters / digits • Check 3 use a range check <ul style="list-style-type: none"> ○ to ensure that the value is >= 1000 and <=9999 	6
3(b)	<p>One mark per mark point, max three</p> <p>MP1 input the new PIN MP2 input the new PIN again // ask the user to check the number on screen MP3 check that both PINs are the same // confirm that it is the PIN to use MP4 check that the new PIN is not the same as the old PIN</p>	3

Question	Answer	Marks																																												
4(a)	<p>One mark for each correct column, max four</p> <table><tr><th>Sold</th><th>Stock</th><th>Total</th><th>OUTPUT</th></tr><tr><td></td><td>50</td><td>0</td><td></td></tr><tr><td>24</td><td>26</td><td>24</td><td></td></tr><tr><td>12</td><td>14</td><td></td><td>Add new stock</td></tr><tr><td></td><td>64</td><td>36</td><td></td></tr><tr><td>6</td><td>58</td><td>42</td><td></td></tr><tr><td>30</td><td>28</td><td>72</td><td></td></tr><tr><td>12</td><td>16</td><td></td><td>Add new stock</td></tr><tr><td></td><td>66</td><td>84</td><td></td></tr><tr><td>18</td><td>48</td><td>102</td><td></td></tr><tr><td>-1</td><td></td><td></td><td>102</td></tr></table>	Sold	Stock	Total	OUTPUT		50	0		24	26	24		12	14		Add new stock		64	36		6	58	42		30	28	72		12	16		Add new stock		66	84		18	48	102		-1			102	4
Sold	Stock	Total	OUTPUT																																											
	50	0																																												
24	26	24																																												
12	14		Add new stock																																											
	64	36																																												
6	58	42																																												
30	28	72																																												
12	16		Add new stock																																											
	66	84																																												
18	48	102																																												
-1			102																																											
4(b)	<p>One mark for identification of error, max one</p> <ul style="list-style-type: none">the stock level will fall below zero / become negative <p>One mark per mark point, max two</p> <ul style="list-style-type: none">before subtracting the amount Sold from Stocktest that the stock level / Stock is greater than the rolls to be sold / Soldprovide a suitable error message / ask to re-input	3																																												

Question	Answer	Marks																														
5(a)	<p>One mark per mark point, max two</p> <ul style="list-style-type: none">• ItemCode• uniquely identifies each item	2																														
5(b)	<p>One mark per mark point, max three</p> <ul style="list-style-type: none">• correct rows Field, Table and Sort• correct row Show• correct Criteria row <10 or <=9 <table><tr><td>Field:</td><td>ItemCode</td><td>Manufacturer</td><td>Level</td><td></td></tr><tr><td>Table:</td><td>WAREHOUSE</td><td>WAREHOUSE</td><td>WAREHOUSE</td><td></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><10</td><td></td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td></tr></table>	Field:	ItemCode	Manufacturer	Level		Table:	WAREHOUSE	WAREHOUSE	WAREHOUSE		Sort:					Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Criteria:			<10		or:					3
Field:	ItemCode	Manufacturer	Level																													
Table:	WAREHOUSE	WAREHOUSE	WAREHOUSE																													
Sort:																																
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																												
Criteria:			<10																													
or:																																