

# Cambridge IGCSE™

CANDIDATE  
NAME

--

CENTRE  
NUMBER

--	--	--	--	--

CANDIDATE  
NUMBER

--	--	--	--

## COMPUTER SCIENCE

**0478/23**

## Paper 2 Problem-solving and Programming

May/June 2020

**1 hour 45 minutes**

You must answer on the question paper.

No additional materials are needed.

## INSTRUCTIONS

- 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.

## INFORMATION

- 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 **12** pages. 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 tasks before the examination to answer Question 1.

### Pre-release material

A school snack shop sells the following items:

Category	Item code	Description	Price (\$)	Healthy?
Fruit	FR1	Apple	1.00	Yes
Fruit	FR2	Banana	1.25	Yes
Fruit	FR3	Orange	1.00	Yes
Fruit	FR4	Peach	1.30	Yes
Sandwich	SN1	Cheese	1.75	Yes
Sandwich	SN2	Cheese and Tomato	2.20	Yes
Sandwich	SN3	Chicken	2.00	Yes
Sandwich	SN4	Chicken Salad	2.35	Yes
Sandwich	SN5	Salad (Vegetarian)	2.00	Yes
Crisps	CR1	Salted	1.00	No
Crisps	CR2	Cheese and Onion	1.00	No
Crisps	CR3	Barbecue	1.00	No
Crisps	CR4	Prawn Cocktail	1.00	No
Confectionery	CN1	Biscuits	0.95	No
Confectionery	CN2	Chocolate Bar	1.05	No
Confectionery	CN3	Toffee and Chocolate Bar	1.25	No
Confectionery	CN4	Chocolate Wafers	1.30	No
Drink	DR1	Cola	1.20	No
Drink	DR2	Lemon and Lime Soda	1.20	No
Drink	DR3	Orange Soda	1.20	No
Drink	DR4	Orange Juice	1.50	Yes
Drink	DR5	Apple Juice	1.50	Yes

Write and test a program or programs for this snack shop.

- 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 arrays, 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 show the items for sale and to allow items to be selected

Write a program to:

- use arrays to store the item code, description, price and whether or not the item is healthy
- allow a customer to choose an item to purchase
- output the description of each chosen item, its price and whether or not it is healthy
- allow the customer to choose additional items until they decide to stop
- calculate and output the total price of the chosen items.

**Task 2** – Specifying quantities and introducing restrictions

Extend **Task 1** to:

- allow the customer to input the quantity of each item they would like to purchase
- count the number of healthy and unhealthy items chosen by the customer. For example, two apples and a peach count as three healthy items
- check that the customer has chosen at least one healthy item for every two unhealthy items. If so, allow the customer to make the purchase. Otherwise, the purchase cannot proceed, a message is output to say the purchase is cancelled and the customer has to start again.

**Task 3** – Offering discounts

Extend the program to:

- either** apply a 10% discount to the total price, when purchasing both healthy **and** unhealthy items, only if the number of healthy items matches or exceeds the number of unhealthy items
- or** apply a 20% discount to the total price if all the items purchased are healthy items
- calculate then output the new total price and the amount of money saved.

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

(a) (i) Identify **one** array you could have used for **Task 1** and state its purpose.

Array .....

Purpose .....

.....

.....

[2]

(ii) Identify **one** variable you could have used for **Task 2** and state its purpose.

Variable .....

Purpose .....

.....

.....

[2]

(iii) Identify **one** constant you could have used for **Task 3** and state its purpose.

Constant .....

Purpose .....

.....

.....

[2]

(b) Explain why a student would be advised **not** to store the Healthy? data as a real data type and identify the most suitable data type for the Healthy? data.

.....

.....

.....

.....

.....

.....

[2]

- (c) Write an algorithm for **Task 1** using **either** pseudocode, programming statements **or** a flowchart. It is **not** necessary to show initialisation or setting up of arrays.

This image shows a full page of white paper with horizontal dashed lines, typical of primary school writing paper. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.



- (d) Explain how your program completes **Task 3**. Assume **Task 1** and **Task 2** have been completed. Any programming statements used in your answer must be fully explained.

[4]

- (e) Describe how you could alter your program to restrict the maximum number of items purchased by a customer to **six**.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [2]

## Section B

2 Draw a line to connect each **Description** to the most appropriate **Pseudocode example**.

**Description**

A loop that will iterate  
at least once

A loop that will not be  
executed on the first test  
if the condition is false

A conditional statement

Totalling

Counting

**Pseudocode example**

CASE ... OF ... OTHERWISE ... ENDCASE

Number  $\leftarrow$  Number + 1

WHILE ... DO ... ENDWHILE

Sum  $\leftarrow$  Sum + NewValue

REPEAT ... UNTIL

[4]



- 3 This section of pseudocode is to be used as a validation check that will continue until a number between 0 and 499 inclusive is entered.

```

1      PRINT "Input a number from 0 to 499 inclusive"
2      FOR Number ← 1 TO 10
3          INPUT Number
4          IF Number < 0 AND Number > 499
5              THEN
6                  PRINT "Invalid number, please try again"
7          ENDIF
8      UNTIL Number = 0 OR Number = 499
9      PRINT Number, " is within the correct range"

```

There are **three** lines in this pseudocode that contain errors. In each case, state the line number to identify the incorrect line and write out the corrected line in full.

Error 1 line number .....

Correction .....

.....

Error 2 line number .....

Correction .....

.....

Error 3 line number .....

Correction .....

.....

[6]

4 A code must take the form LL9 9LL where L is a letter and 9 is a digit.

- (a) A presence check has already been used to ensure data has been entered. Name **two** other types of validation check that can be used to test the code is valid.

Check 1 .....

Check 2 .....

[2]

- (b) Give **one** example of invalid test data for each of the validation checks you have named in **part (a)** and in each case, give a reason why it fails the check. Each example of test data must be different.

Check 1 Invalid Test Data .....

.....

Reason .....

.....

.....

.....

Check 2 Invalid Test Data .....

.....

Reason .....

.....

.....

.....

[4]

5 This algorithm finds prime numbers.

The pre-defined function DIV gives the value of the result of integer division, for example,  $y \leftarrow 9 \text{ DIV } 4$  gives  $y$  a value of 2

```

Flag ← False
INPUT Number
WHILE Number <> 0
  Divisor ← 2
  WHILE Divisor ≤ Number / 2
    Value ← Number DIV Divisor
    IF Number / Divisor = Value
      THEN
        Flag ← True
      ENDIF
    Divisor ← Divisor + 1
  ENDWHILE
  IF Flag = False
    THEN
      OUTPUT Number, " is prime"
    ENDIF
  INPUT Number
  Flag ← False
ENDWHILE

```

Complete the trace table for the algorithm using the input data:

5, 6, 8, 0, 11, 13

Flag	Number	Divisor	Value	OUTPUT

[5]

- 6 Draw the flowchart symbol for **Decision** and the flowchart symbol for **Process**.

Decision	Process

[2]

- 7 A convenience store which sells general groceries wants to set up a database table called STOCK. The table will contain fields including a description of the item, the price of the item and the number in stock for each item. The STOCK table also has a fourth field to be used as a primary key.

- (a) Complete the table to suggest a suitable field name for each of the **four** fields in the table STOCK. Give the purpose of the data to be stored in each field.

Field name	Purpose of field contents

[4]

- (b) Complete the query-by-example grid to output stock items where the quantity in stock has fallen below 20. Only show the primary key and description of the items.

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

[3]

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](http://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

May/June 2020

MARK SCHEME

Maximum Mark: 50

---

<p><b>Published</b></p>
-------------------------

Students did not sit exam papers in the June 2020 series due to the Covid-19 global pandemic.

This mark scheme is published to support teachers and students and should be read together with the question paper. It shows the requirements of the exam. The answer column of the mark scheme shows the proposed basis on which Examiners would award marks for this exam. Where appropriate, this column also provides the most likely acceptable alternative responses expected from students. Examiners usually review the mark scheme after they have seen student responses and update the mark scheme if appropriate. In the June series, Examiners were unable to consider the acceptability of alternative responses, as there were no student responses to consider.

Mark schemes should usually be read together with the Principal Examiner Report for Teachers. However, because students did not sit exam papers, there is no Principal Examiner Report for Teachers for the June 2020 series.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the June 2020 series for most Cambridge IGCSE™ and Cambridge International A & AS Level components, and some Cambridge O Level components.

---

This document consists of **11** printed pages.

**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
1(a)(i)	<p>Any array related to <b>Task 1</b> – <b>one</b> mark e.g.  ItemCode  Description</p> <p>Correct purpose related to <b>Task 1</b> - <b>one</b> mark e.g.  ...to store the product code of the items in stock  ...to store descriptions/names of the items on sale</p>	<b>2</b>
1(a)(ii)	<p>Any variable related to <b>Task 2</b> – <b>one</b> mark e.g.  Quantity  HealthyNum</p> <p>Correct purpose related to <b>Task 2</b> - <b>one</b> mark e.g.  ... to allow input of the quantity of a product  ... to store/track the number of healthy items in the order</p>	<b>2</b>
1(a)(iii)	<p>Any constant related to <b>Task 3</b> – <b>one</b> mark e.g.  Off10  Off20</p> <p>Correct purpose related to <b>Task 3</b> - <b>one</b> mark e.g.  ... to store the 'matching healthy items' option discount rate  ... to store the 'all healthy items' discount rate</p>	<b>2</b>
1(b)	<p>Any <b>two</b> correct statements e.g.</p> <ul style="list-style-type: none"> <li>• The Healthy? data is not numerical</li> <li>• Boolean</li> </ul>	<b>2</b>



Question	Answer	Marks
1(c)	<p>Any <b>six</b> from</p> <ul style="list-style-type: none"> <li>• Input of item/item code (with message)</li> <li>• Attempt at use of validation of input for item/item code</li> <li>• Fully functional validation of input for item/item code</li> <li>• Find location of item/item code in array</li> <li>• Retrieve item description, price and whether it is healthy</li> <li>• Check if another item is required – message and input</li> <li>• Use of loop to repeat purchase option</li> <li>• Running total of price</li> <li>• Output of description of each item, price and healthy status (with message)</li> <li>• Output total price <u>with message</u> (outside final loop)</li> </ul> <p><b>Example answer</b></p> <pre> AnotherItem ← "Y" While AnotherItem = "Y"     SnackCodeFlag ← False     OUTPUT "Input an Item Code"     WHILE SnackCodeFlag = False         INPUT SnackCode         Count ← 0         WHILE Count &lt; 22 DO             IF SnackCode = ItemCode[Count]                 THEN                     SnackCodeFlag ← True                     OUTPUT "Item Description ", Description[Count], " Item Price ",                         Price[Count], " Item is Healthy? ", Healthy[Count]                     TotalPrice ← TotalPrice + Price[Count]                     Count ← 22                 ENDIF             Count ← Count + 1         ENDWHILE     IF SnackCodeFlag = False         THEN </pre>	6

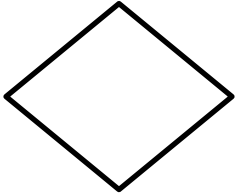

Question	Answer	Marks
1(c)	<pre> OUTPUT "Your Item Code doesn't exist, please try again" ENDIF ENDWHILE OUTPUT "Would you like another item? (Answer Y or N)" INPUT AnotherItem ENDWHILE OUTPUT "The total price is ", TotalPrice </pre>	
1(d)	<p>Any <b>four</b> from</p> <ul style="list-style-type: none"> <li>• Explanation of comparing numbers of healthy and non-healthy items</li> <li>• Explanation of checking for zero unhealthy items</li> <li>• Explanation of applying the 10% discount rule</li> <li>• Explanation of applying the 20% discount rule</li> <li>• Explanation of calculating the new total price and money saved</li> <li>• Explanation of output</li> </ul>	<b>4</b>
1(e)	<p><b>One</b> for each correct statement (max <b>two</b>)</p> <ul style="list-style-type: none"> <li>• Use a loop structure when making purchases... / introduce a count of items when making purchases ... /modify the existing loop structure</li> <li>• ... that terminates after six iterations</li> </ul>	<b>2</b>

Question	Answer	Marks
2	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p><b>Description</b></p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">A loop that will iterate at least once</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">A loop that will not be executed on the first test if the condition is false</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">A conditional statement</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Totalling</div> <div style="border: 1px solid black; padding: 5px;">Counting</div> </div> <div style="text-align: center;"> <p><b>Pseudocode example</b></p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">CASE ... OF ... OTHERWISE ... ENDCASE</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Number ← Number + 1</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">WHILE ... DO ... ENDWHILE</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">Sum ← Sum + NewValue</div> <div style="border: 1px solid black; padding: 5px;">REPEAT ... UNTIL</div> </div> </div> <p> <b>One</b> mark – <b>one</b> correct link  <b>Two</b> marks – <b>two</b> correct links  <b>Three</b> marks – <b>three</b> correct links  <b>Four</b> marks – <b>four/five</b> correct links </p>	<b>4</b>

Question	Answer	Marks
3	<p>Line 2 and Line 4 errors - <b>One</b> mark for each correct identification <b>and</b> correction of error</p> <p>Error 1 line number: Line 2 Correction: REPEAT</p> <p>Error 2 line number: Line 4 Correction: IF Number &lt; 0 OR Number &gt; 499</p> <p>Line 8 error - <b>One</b> mark for correct identification of error and <b>one</b> mark for <b>each</b> correction of error Error 2 line number: Line 8 Correction: UNTIL Number &gt;= 0 <b>AND</b> Number &lt;= 499</p>	6

Question	Answer	Marks
4(a)	<p><b>One</b> mark for each correct check (max <b>two</b>)</p> <ul style="list-style-type: none"> <li>• Length (check)</li> <li>• Type Check</li> <li>• Format Check</li> </ul>	2
4(b)	<p><b>One</b> mark for each suitable piece of test data and <b>one</b> mark for each relevant reason (max <b>four</b>)</p> <ul style="list-style-type: none"> <li>• LL9999LL999</li> <li>• Too long</li> <li>• 5678987</li> <li>• All numeric</li> <li>• CB12EU</li> <li>• No space is present</li> </ul>	4

Question	Answer					Marks																																																															
5		<table><tr><th>Flag</th><th>Number</th><th>Divisor</th><th>Value</th><th>OUTPUT</th></tr><tr><td>False</td><td>5</td><td>2</td><td>2</td><td></td></tr><tr><td></td><td></td><td>3</td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td>5 is prime</td></tr><tr><td>False</td><td>6</td><td>2</td><td>3</td><td></td></tr><tr><td>True</td><td></td><td>3</td><td>2</td><td></td></tr><tr><td>True</td><td></td><td>4</td><td></td><td></td></tr><tr><td>False</td><td>8</td><td>2</td><td>4</td><td></td></tr><tr><td>True</td><td></td><td>3</td><td>2</td><td></td></tr><tr><td>True</td><td></td><td>4</td><td>2</td><td></td></tr><tr><td></td><td></td><td>5</td><td></td><td></td></tr><tr><td>False</td><td>0</td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>	Flag	Number	Divisor	Value	OUTPUT	False	5	2	2				3							5 is prime	False	6	2	3		True		3	2		True		4			False	8	2	4		True		3	2		True		4	2				5			False	0										5
	Flag	Number	Divisor	Value	OUTPUT																																																																
	False	5	2	2																																																																	
			3																																																																		
					5 is prime																																																																
	False	6	2	3																																																																	
	True		3	2																																																																	
	True		4																																																																		
	False	8	2	4																																																																	
	True		3	2																																																																	
	True		4	2																																																																	
			5																																																																		
	False	0																																																																			
One mark for each correct column																																																																					

Question	Answer	Marks
6	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>Decision</b>   </div> <div style="text-align: center;"> <b>Process</b>   </div> </div> <p><b>One</b> mark for each correct symbol</p>	2

Question	Answer	Marks										
7(a)	<table><tr><th>Fieldname</th><th>Purpose of field contents</th></tr><tr><td>CodeNo</td><td>Primary key to identify products</td></tr><tr><td>Product</td><td>To describe the product</td></tr><tr><td>Price</td><td>The price of individual item</td></tr><tr><td>NumInStock</td><td>How many are in stock</td></tr></table> <p>One mark for each correct fieldname and description pair</p>	Fieldname	Purpose of field contents	CodeNo	Primary key to identify products	Product	To describe the product	Price	The price of individual item	NumInStock	How many are in stock	4
Fieldname	Purpose of field contents											
CodeNo	Primary key to identify products											
Product	To describe the product											
Price	The price of individual item											
NumInStock	How many are in stock											

Question	Answer				Marks
7(b)	Field:	CodeNo	Product	NumInStock	3
	Table:	STOCK	STOCK	STOCK	
	Sort:				
	Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
	Criteria:			<20	
	or:				
	One mark for each completely correct column (max <b>three</b> )				