Cambridge Assessment
International Education

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 ItemCode

Purpose To store the product codes of the items in the school snack shop

[2]

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

Variable Quantity

Purpose To allow for the input of quantity of a product

[2]

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

Constant off10

Purpose To store the matching healthy items options discount rate.

[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.

A real data type would be having numerals as its data. The Healthy column of this program design is holding text. It is having a data of 'yes' or 'no'. The best data type in this case would be boolean.

[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.

```
// Setting up the database / system

For x ← 1 to 22
    OUTPUT " Enter Category"
    INPUT Category [x]
    OUTPUT " Enter Item Code"
    INPUT ItemCode[x]
    OUTPUT " Enter Description"
    INPUT Description[x]
    OUTPUT "Enter Price"
    INPUT Price [x]
    OUTPUT " Is it healthy or Not" Yes or No"
    INPUT Health [x]

END FOR
 // allowing the customer to purchase an item
 OUTPUT "Do You wish to purchase an Item"
 INPUT Answer
  WHILE Answer = "Yes" DO
     OUTPUT " Select an Item below"
      FOR Index ← 1 To 22
          OUTPUT Category [Index], Index
      Next Index
     INPUT SelectedItemIndex
     OUTPUT Description [SelectedItemIndex]
     OUTPUT 'The price is", Price[SelectedItemIndex]
     OUTPUT "The health Status is, Health[SelectedItem
                                               Index]
```

```
        TotalPrice ← TotalPrice + Price[SelectedItemIndex]

        OUTPUT "Do you wish to purchase another Item Yes or No"

        INPUT Answer

ENDWHILE

OUTPUT "The total price of the Item(s) is"

OUTPUT TotalPrice
```

[6]

(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.

Include two variables healthyCount and UnhealthyCount.

Every time an item is purchased you check if it is healthy or not.

You add this to the corresponding variable

At the end of the loop you check to see if the number of the healthy meals is greater or equal to the unhealthy ones.

If this is the case you multiply 0.90 to the Total price and set this as the new price

Otherwise check if all the items purchased are all healthy items.

If this is the case you calculate a new price as 0.80 multiplied by the Total price.

[4]

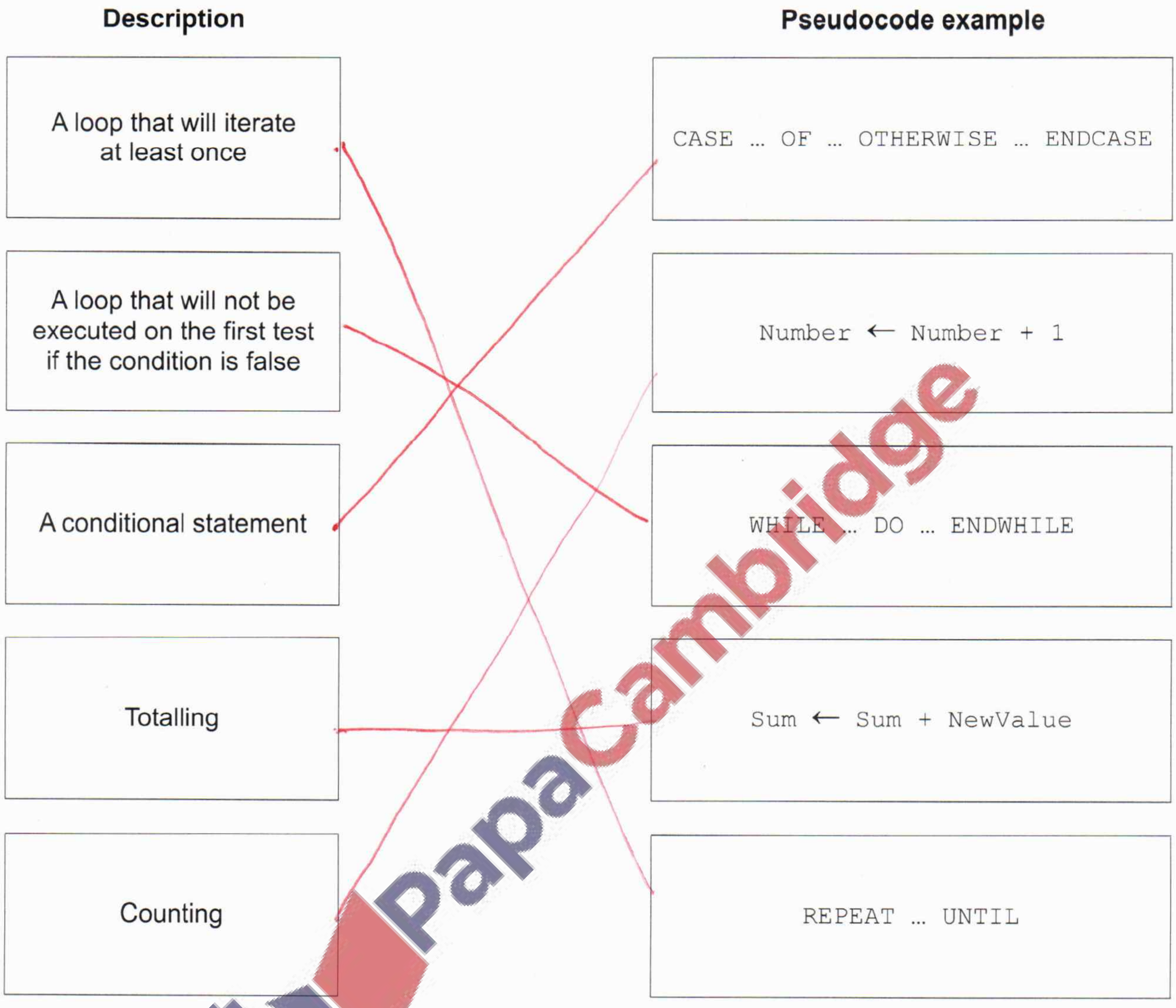
(e) Describe how you could alter your program to restrict the maximum number of items purchased by a customer to **six**.

Add a variable count limit example
max. Set this variable to 0
Everytime an item is purchased, increament
count limit variable max
Using a precondition loop always check
if this variable has reached max
You may also use a post condition
loop to check that the value of the
count limit max has not reached 6
So you can ask again for a new item if
they want

[2]

Section B

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



Description	Pseudocode example
A loop that will iterate at least once	`CASE … OF … OTHERWISE … ENDCASE`
A loop that will not be executed on the first test if the condition is false	`Number ← Number + 1`
A conditional statement	`WHILE … DO … ENDWHILE`
Totalling	`Sum ← Sum + NewValue`
Counting	`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 ....Line 02....

Correction ....REPEAT....

....

Error 2 line number ....Line 04....

Correction ....IF Number <=0 ~~AND~~ OR Number >=499....

....

Error 3 line number ....Line 08....

Correction ....UNTIL Number >=0 AND Number <=499....

....

[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 format check

Check 2 Type check

[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 LL9000 9LL000

Reason This is too long and should be rejected.

Check 2 Invalid Test Data 597967

Reason This data consists of all numbers

[4]

5 This algorithm finds prime numbers.

The pre-defined function `DIV` gives the value of the result of integer division, for example, `y ← 9 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
```

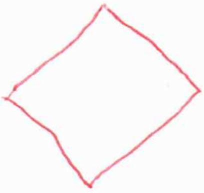
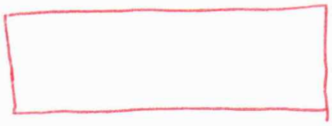
Complete the trace table for the algorithm using the input data:

5, 6, 8, 0, 11, 13

Flag	Number	Divisor	Value	OUTPUT
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]

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
CodeNumber	Primary key to ID each product
Price	Contain the price of the item
Product	Contain the description of the item
NumberInStock	Contain the number of items in stock

[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:	CodeNumber	Product	NumberInStock	
Table:	STOCK	STOCK		
Sort:				
Show:	☑	☑	□	□
Criteria:			<20	
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 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.