Cambridge Assessment International Education

Cambridge O Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **2210/21**

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 **16** pages. Blank pages are indicated.

[Turn over

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 shop sells a range of mobile devices, SIM cards and accessories as shown in the table:

Category	Item code	Description	Price ($)
Phone	BPCM	Compact	29.99
Phone	BPSH	Clam Shell	49.99
Phone	RPSS	RoboPhone – 5-inch screen and 64 GB memory	199.99
Phone	RPLL	RoboPhone – 6-inch screen and 256 GB memory	499.99
Phone	YPLS	Y-Phone Standard – 6-inch screen and 64 GB memory	549.99
Phone	YPLL	Y-Phone Deluxe – 6-inch screen and 256 GB memory	649.99
Tablet	RTMS	RoboTab – 8-inch screen and 64 GB memory	149.99
Tablet	RTLM	RoboTab – 10-inch screen and 128 GB memory	299.99
Tablet	YTLM	Y-Tab Standard – 10-inch screen and 128 GB memory	499.99
Tablet	YTLL	Y-Tab Deluxe – 10-inch screen and 256 GB memory	599.99
SIM card	SMNO	SIM Free (no SIM card purchased)	0.00
SIM card	SMPG	Pay As You Go (SIM card purchased)	9.99
Case	CSST	Standard	0.00
Case	CSLX	Luxury	50.00
Charger	CGCR	Car	19.99
Charger	CGHM	Home	15.99

Write and test a program or programs for this 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.
Write a program to:

- use appropriate data structures to store the item code, description and price information for the mobile devices, SIM cards and accessories
- allow the customer to choose a specific phone or tablet
- allow phone customers to choose whether the phone will be SIM Free or Pay As You Go
- allow the customer to choose a standard or luxury case
- allow the customer to choose the chargers required (none, one or both may be purchased)
- calculate the total price of this transaction
- output a list of the items purchased and the total price.

Task 2 – Allow a customer to order multiple mobile devices.
Extend **Task 1** to:

- offer the customer the opportunity to purchase an additional mobile device
- if required, perform bulleted steps 2 to 7 of **Task 1** for each additional mobile device and calculate a running total for the customer
- once no further devices are required, output the total the customer will need to pay.

Task 3 – Offering discounts.
Extend the program to allow a discount of 10% off the price of every additional phone or tablet purchased.
Output the new total the customer will need to pay and the amount of money saved.

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

(a) (i) Identify **two** arrays you could have used for **Task 1** and, in each case, state its purpose.

Array 1 ..... PhoneType .....

Purpose ..... To store the types of phones .....
..... available for sale .....
.....

Array 2 ..... TabletTypes .....

Purpose ..... To store the types of tablets .....
..... available for sale. .....
.....

[4]

(ii) Identify **two** variables you could have used for **Task 1** and, in each case, state its purpose.

Variable 1 ..... RunningTotal .....

Purpose ..... To hold the running total cost .....
..... of a phone and accessories that .....
..... are being selected by a user. .....

Variable 2 ..... PurchaseItem .....

Purpose ..... To hold the reply of wether a .....
..... user wishes to purchase the item or .....
..... not. .....

[4]

(b) Explain why the item code data could **not** be stored as a real data type and identify the most suitable data type for the item code data.

..... The code of the item is a mix of .....
..... letters. Real data type requires a .....
..... fractional number. .....
..... The item code is best stored as string .....
.....
..... [2]

(c) Write an algorithm to show how you input your choice of mobile device and SIM card (part of **Task 1**), using **either** pseudocode, programming statements **or** a flowchart. It is **not** necessary to show initialisation or setting up of arrays containing product details.

```
REPEAT
OUTPUT " Select Item Code from below"

FOR index ← 1 to 6
    OUTPUT Phone[Index], PhoneCode[Index]
Next Index
FOR Index ← 1 to 4
    OUTPUT Tablet[index], TabletCode[Index]
Next Index
  Count ← Count +1
INPUT ItemCode.
Found ← False

If found ← false then
    For Index ← 1 to 6
        If phoneCode = ItemCode[index]
            found ← True
        endif
    next Index
    For Index ← 1 to 4
        If TabletCode[index] = ItemCode
            found ← True
        endif
    next index
Until found = True or Count = 10
```

```
If found = false
    Output "your device code does not exist"
else
    If device_type = "Phone" Then
    Output "Would you prefer a Sim Card
        or Pay as you Go Y or N"

    INPUT Answer.

EndIF
```

[5]

(d) Explain how your program completes **Task 3**. Any programming statements used in your answer must be fully explained.

At the end of a purchase of a single phone or tablet ask the user if they would prefer to make another purchase.
If the answer is yes allow the user to select properties of their next phone.
apply a 0.10 discount to this new item price.
Subtract the original cost to this new cost to get the difference. Amount saved.

The above program will use variables NumOfPurchasedItems. and Reply.
It will use a post conditional loop to check if the Reply is 'Yes' that the Customer wishes to make another purchase.

[5]

Section B

2 Tick (✓) **one** box in each row to identify if the statement about structure diagrams is true or false.

Statement	True (✓)	False (✓)
A structure diagram is a piece of code that is available throughout the structure of a program.		✓
A structure diagram shows the hierarchy of a system.	✓	
A structure diagram is another name for an array.		✓
A structure diagram shows the relationship between different components of a system.	✓	

[2]

3 Programs can perform validation and verification checks when data is entered.

(a) Give the names of **two** different validation checks and state the purpose of each one.

Check 1 ..... Range check .....

Purpose ..... To check that data .....

..... falls within a specific range .....

.....

Check 2 ..... Type check .....

Purpose ..... To check that data is of .....

..... a certain type. .....

.....

[4]

(b) Give the name of **one** verification check.

..... Visual check. ..... [1]

(c) Describe the difference between validation and verification.

..... Validation is the automatic check that .....

..... the data is correct .....

..... Verification is to confirm that data entered .....

..... is the one required or has been ..... [2]

entered correctly

4 The pseudocode algorithm shown should allow numbers to be entered and should allow 50 numbers to be stored in an array.

```
Count ← 0
REPEAT
  INPUT Values[Count]
  Count ← Count + 1
UNTIL Count = 0
```

(a) Explain why the algorithm will never end.

As the count increases it will go past 0.
The check at the end of the loop will always return false so the loop will never end.

[2]

(b) Re-write the original pseudocode so that it terminates correctly **and** also prevents numbers below 100 from being stored in the array `Values[ ]`

```
Count ← 0
REPEAT
    INPUT Number
    IF Number >= 100 Then
        Values[Count] ← Number
    ENDIF
    Count ← Count + 1
Until Count = 50
```

[4]

(c) Describe how you could change your pseudocode in **part (b)** so that it prevents numbers below 100 and above 200 from being stored in the array `Values[ ]`

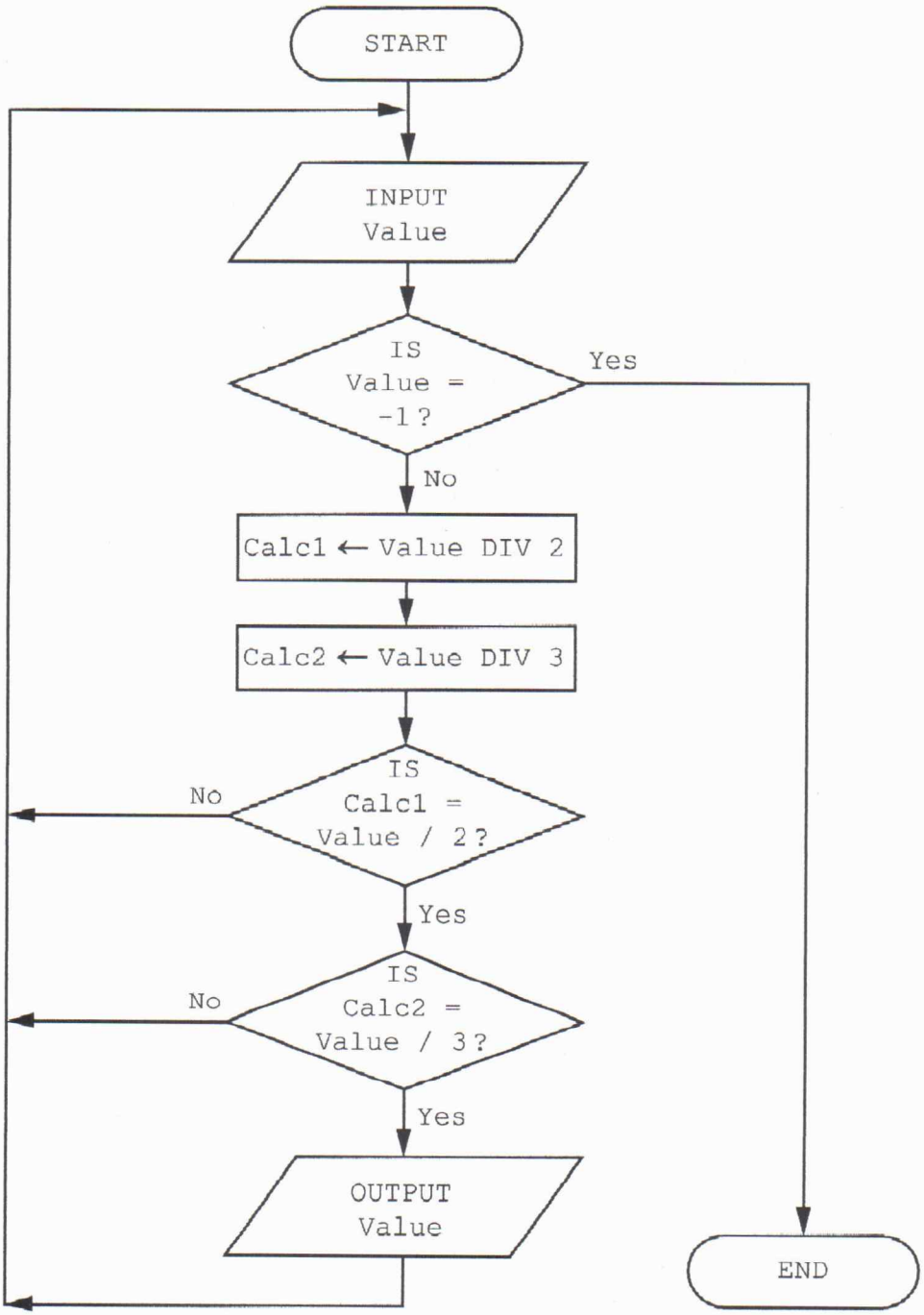
In the check add an extra check to see if the number entered is greater than 200 or below 100
IF Number < 100 or Number > 200

[2]

5 The flowchart represents an algorithm.

The predefined function `DIV` gives the value of the result of integer division, for example, `y ← 9 DIV 4` gives `y` a value of `2`

An input value of –1 ends the algorithm.



(a) Complete the trace table for the input data:

50, 33, 18, 15, 30, –1, 45, 12, 90, 6

Value	Calc1	Calc2	OUTPUT
50	25	16	
33	16	11	
18	9	6	18
15	7	5	
30	15	10	30
-1			

[4]

(b) Describe the purpose of the algorithm.

The programs outputs the values that are divisible by 6.

[2]

6 A garden centre sells garden tools and stores details of these in a database table named TOOLS. **Code** is the primary key in the TOOLS table.

Code	Description	Price ($)	Quantity_Stock	Quantity_Ordered
GFLG	Garden Fork	50.00	1	50
GSLG	Garden Spade	50.00	11	0
GHLG	Garden Hoe	45.00	8	0
HFSM	Hand Fork	9.99	42	0
HSSM	Hand Spade	9.99	40	0
HWSM	Hand Weeder	9.99	11	0
HS20	Hose (20 metres)	45.00	10	0
HS35	Hose (35 metres)	60.00	2	0
HS50	Hose (50 metres)	75.00	20	60
YBLG	Yard Brush	24.99	100	0
LMHD	Lawn Mower	99.99	5	0
LMBT	Lawn Mower (Battery)	249.99	7	0
LMPT	Lawn Mower (Petrol)	349.99	10	25
TRBT	Edge Trimmer (Battery)	79.99	15	0
TRPT	Edge Trimmer (Petrol)	59.99	20	0
SHSM	Shears	40.00	40	0
HCSM	Hedge Clippers	40.00	45	0

(a) State the purpose of the primary key in the TOOLS table.

To uniquely identify each record in the table.

[1]

(b) List the output from the data shown in the table TOOLS that would be given by this query-by-example.

Field:	Code	Description	Price ($)	Quantity_Stock	Quantity_Ordered
Table:	TOOLS	TOOLS	TOOLS	TOOLS	TOOLS
Sort:					Descending
Show:	☑	☑	☐	☐	☑
Criteria:			>40	>0	>0
or:					

HS50 Hose (50 metres) 60

GFLG Garden Fork 50

LMPT Lawn Mower 25

[3]

(c) Complete the query-by-example grid to output the tools where the quantity in stock is below 25. Only show the Code, Description and Quantity_Stock fields in ascending order of Code.

Field:	Code	Description	Quantity_Stock	
Table:	Tools	Tools	Tools	
Sort:	Ascending			
Show:	☑	☑	☑	☐
Criteria:			<25	
or:				

[3]