Cambridge IGCSE™

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **0478/22**

Paper 2 Problem-solving and Programming **February/March 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.

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 car buying service helps customers calculate the price of buying a new car from Snazzy Autos. There are three models of car available: hatchback, saloon and estate. A car can be bought with optional extras.

Model	Price
Hatchback	Rs 5.35 lakh
Saloon	Rs 4.95 lakh
Estate	Rs 6.25 lakh

Optional extra	Price
Set of luxury seats	Rs 45000
Satellite navigation	Rs 5500
Parking sensors	Rs 10000
Bluetooth connectivity	Rs 350
Sound system	Rs 1000

Snazzy Autos customers buying a new car can trade in an old car. Offers from Rs 10000 to Rs 1.00 lakh can be made for an old car. If an old car is not traded in, then a discount of 5% is taken off the price of the new car and any optional extras. In addition, repeat customers are offered a discount of 10% off any optional extras and 10% off the price of the new car before trade-in. Customers can either pay for the new car in full or make monthly payments over several years.

Write and test a program or programs for the car buying service.

- 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 – Calculate the price

Customers choose the model of car and select any optional extras required. Only valid choices can be accepted for the model and the extras. The customer is asked if they are an existing customer and if they have an old car to trade in. If the customer trades in an old car the amount offered by Snazzy Autos is entered. The price of the new car including model and extras is displayed. Any discount or trade-in amounts are displayed and the total price to pay is calculated and displayed.

Task 2 – Calculate the payment(s)

Extend **Task 1** to display these payment methods:

- full amount paid now customer receives cashback of 1% on the total price
- equal monthly payments are made over four years with no extra charge
- the total price is increased by 5%; equal monthly payments are made over seven years.

For each payment method display: the total amount to pay, the amount of each payment, the number of payments and cashback where appropriate.

Task 3 – New full amount payment offer

Extend **Task 2** to offer customers who decide to pay the full amount now either the 1% cashback or the chosen optional extras free. Customers are shown the offer that is the best value first and the other as an alternative. Details of each offer must be displayed in full.

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

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

Constant name ....

Value ....

Use ....

....
[3]

(b) State the name of the variable that you used to store the amount offered for the trade-in of an old car in **Task 1**. Give the most appropriate data type for this variable. Explain how your program ensured that any data entered for the trade-in offer was valid.

Variable name ....

Data type ....

Validation ....

....

....

....
[4]

(c) Snazzy Autos has decided to add another model to the cars available in **Task 1**. The new model is a convertible, price Rs 6.75 lakh. Explain the changes required to your program for **Task 1**.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [4]

(d) Write an algorithm for **Task 2**, using **either** pseudocode, programming statements **or** a flowchart. Assume that **Task 1** has been completed.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [5]

(e) Explain how your program completed **Task 3**. Assume that **Task 2** has been completed. Any programming statements used in your answer must be fully explained.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [4]

Section B starts on page 8.

Section B

2 **(a)** An algorithm has been written in pseudocode to input 50 numbers. Positive numbers are stored in the array `PosNum[ ]`. Negative numbers are stored in the array `NegNum[ ]`. Zeros are not included in the positive and negative counts.

```
Count ← 0
PosCount ← Count
NegCount ← Count
REPEAT
    INPUT Number
    IF Number > 0
      THEN
         PosCount ← PosCount + 1
         PosNum[PosCount] ← Number
      ELSE
         NegCount ← NegCount + 1
         NegNum[NegCount] ← Number
    ENDIF
    Count ← Count + 1
UNTIL Count >= 50
OUTPUT "There are ", PosCount," positive numbers"
OUTPUT "There are ", NegCount," negative numbers"
```

Describe the error in the pseudocode and write the correction for this error.

Error ....

....

....

....

Correction ....

....

....

....

[4]

(b) The algorithm needs to be changed so there is no limit to how many numbers can be input. When the number 9999 is input, the algorithm stops more numbers being input and outputs the results. The number 9999 is not to be stored nor counted as a positive number.

Explain how you would change the algorithm.

....

....

....

....

....

....

....

....

....

.... [4]

3 This flowchart inputs student percentage marks for three examinations. If the average of these marks is 80% or over then a distinction grade is awarded. If the average of these marks is less than 40% then a fail grade is awarded. Otherwise a pass grade is awarded.

```
                    ┌─────────────────┐
                    │      START       │
                    └─────────────────┘
                             │
                             ▼
                    ╱─────────────────╲
                   ╱      INPUT         ╲
                   ╲  Mark1,Mark2,Mark3 ╱
                    ╲─────────────────╱
                             │
                             ▼
                    ┌─────────────────┐
                    │  Total ← Mark1 + │
                    │   Mark2 + Mark3  │
                    └─────────────────┘
                             │
                             ▼
                    ┌─────────────────┐
                    │ Average ← Total / 3 │
                    └─────────────────┘
                             │
                             ▼
                          ◇ IS ◇                    ╱─────────────╲
                       Average >= 80 ?    Yes      ╱   OUTPUT      ╲
                          ◇       ◇  ───────────▶ ╲  "Distinction" ╱
                             │                     ╲─────────────╱
                             │ No                        │
                             ▼                           │
    ╱─────────────╲       ◇ IS ◇                         │
   ╱   OUTPUT      ╲ Yes  Average < 40 ?                 │
   ╲   "Fail"      ╱ ◀──── ◇       ◇                     │
    ╲─────────────╱           │                          │
          │                   │ No                       │
          │                   ▼                          │
          │           ╱─────────────╲                    │
          │          ╱   OUTPUT      ╲                    │
          │          ╲   "Pass"      ╱                    │
          │           ╲─────────────╱                    │
          │                   │                          │
          └───────────────────┼──────────────────────────┘
                              ▼
                    ┌─────────────────┐
                    │      END         │
                    └─────────────────┘
```
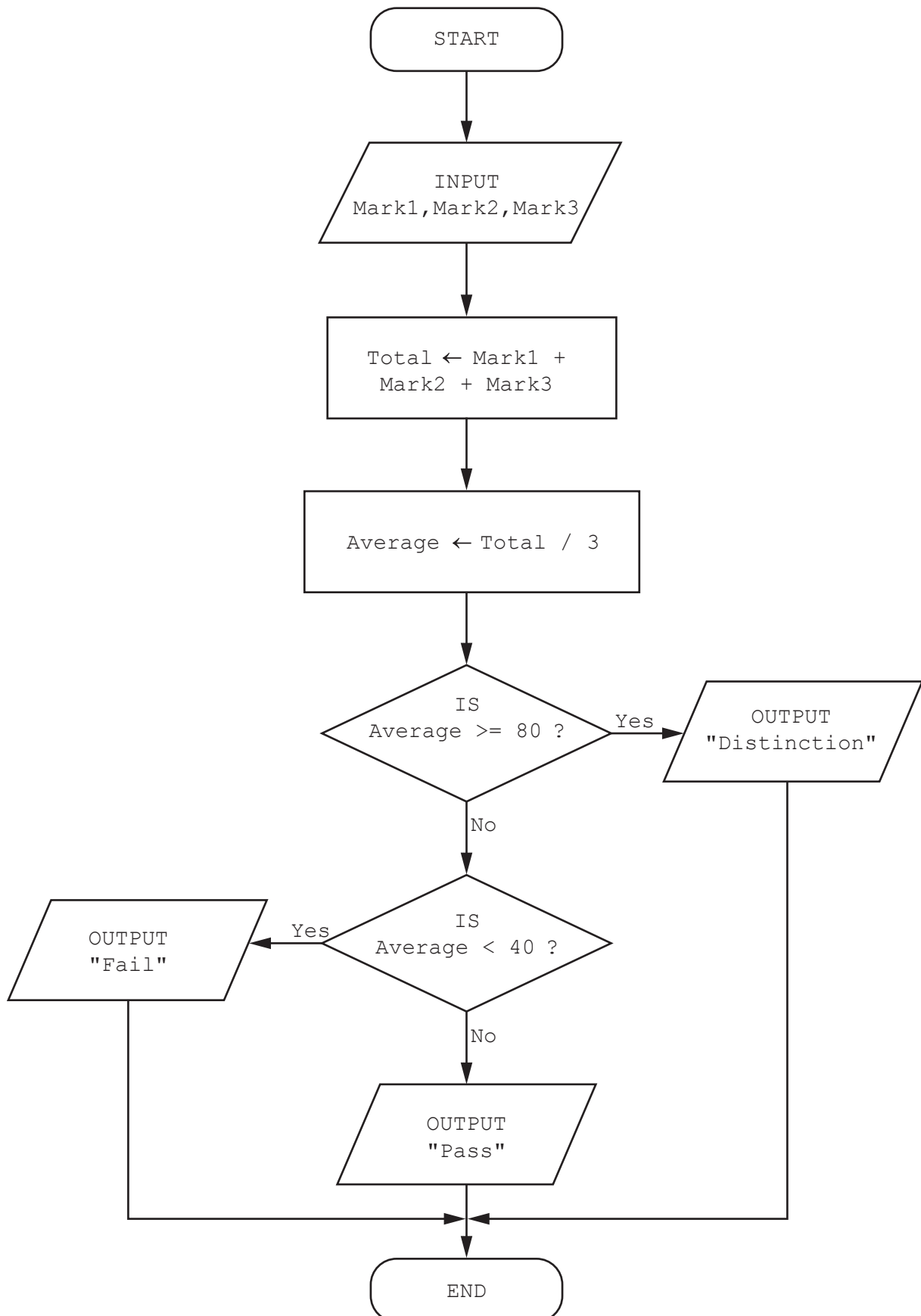
(a) Complete a trace table for each set of input data:

Set 1: 88, 74, 60

Mark1	Mark2	Mark3	Total	Average	OUTPUT

Set 2: 20, 33, 67

Mark1	Mark2	Mark3	Total	Average	OUTPUT

Set 3: 79, 91, 70

Mark1	Mark2	Mark3	Total	Average	OUTPUT

[5]

(b) It has been decided to include an extra grade of Merit when the average of the marks is 60% or more, and less than 80%. Describe the changes that will need to be made to the flowchart.

....

....

....

....

....

....

....

.... [3]

4   Explain why validation and verification checks are needed when data is input.
    Include an example of each type of check in your answer.

    ....

    ....

    ....

    ....

    ....

    ....

    ....

    ....

    .... [4]

Question 5 starts on page 14.

 [Turn over

5 A database table, BOX, is used to keep a record of boxes made and sold by a craftsman. Boxes are categorised by:

- SIZE – small, medium or large
- SHAPE – brief description for example 'star shaped'
- WOOD – maple, beech, walnut or ebony
- PRICE – price in $
- SOLD – whether the box is sold or not.

A database management system uses these data types:

Text   Number   Currency   Boolean

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

SIZE data type ....

Reason ....

....

SHAPE data type ....

Reason ....

....

WOOD data type ....

Reason ....

....

PRICE data type ....

Reason ....

....

SOLD data type ....

Reason ....

....

[5]

(b) (i) Complete the query-by-example grid below to only display the price of small walnut boxes.

Field:				
Table:				
Sort:				
Show:	☐	☐	☐	☐
Criteria:				
or:				

[3]

(ii) The query-by-example grid from part **(b)(i)** needs to be changed to show both walnut and beech boxes and display the wood used. Only one column needs to be changed. Write the changed column.

Field:	
Table:	
Sort:	
Show:	☐
Criteria:	
or:	

[2]

BLANK PAGE

Cambridge IGCSE™

COMPUTER SCIENCE **0478/22**

Paper 2 **March 2020**

MARK SCHEME

Maximum Mark: 50

Published

This document consists of **9** 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
	Section A	
1(a)	Any appropriate constant, for example: **Constant name** `noTradeInDiscount` **Value** 0.05/5 **Use** Storing the percentage discount if no car is traded in	**3**
1(b)	**Variable name** `TradeInValue` // any suitable variable name for trade-in amount **Data type** real/integer **Validation** any **two** from • Use of conditional /loop statement / `IF … THEN` / `REPEAT … UNTIL` / `WHILE` • value input is greater than (or equal to) 10 000/0.1 / less than (or equal to) 100 000/1 / numeric // a value is actually input // error message for rejected input • range check // type check // presence check	**4**
1(c)	Any **four** from: MP1 Display extra model/choice MP2 Increase options available to 4 MP3 Extend validation rule / selection statement(s) for input selection MP4 Store name of extra model e.g. new variable/constant/extra item in array MP5 Store price of extra model e.g. new variable/ constant/extra item in array MP6 Change code to calculate price to pay to include extra model	**4**

Question	Answer	Marks

1(d)	Any **five** from:
	MP1 Calculate cashback for payment method 1
	MP2 Calculate monthly payments for payment method 2
	MP3 Calculate amount and monthly payments for payment method 3
	MP4 Method 1 – output full amount, one payment and cashback
	MP5 Method 2 – output full amount, full amount divided by 48 and 48 payments
	MP6 Method 3 – output full amount times 1.05, full amount times 1.05 divided by 84 and 84 payments
	MP7 Attempt to output showing all 3 payment methods with appropriate messages (MPs 4,5,6 not awarded)
	
	Sample answer:
	<pre>PRINT "Full Amount ",amountToPay, " to pay. Number of Payments is 1"
	PRINT "Cashback ", amountToPay * 0.01
	PRINT "Over four years ",amountToPay, " to pay. Number of Payments is 48"
	PRINT "Each monthly payment is ", amountToPay / 48
	PRINT "Over seven years ",amountToPay * 1.05, " to pay. Number of Payments is 84"
	PRINT "Each monthly payment is ", amountToPay * 1.05 / 84</pre>
		5

1(e)	Explanation Any **four** from: MP1 Use of selection statement to check if customer chose to pay the full amount MP2 Consideration of special case where there are no optional extras chosen MP3 Comparison of 1% of total price to pay with the total cost of optional extras MP4 Use of selection statement to check for the largest value of cashback or extras /smallest amount to pay MP5 … display the cost of this option first MP6 … then display the cost of the other option	**4**

Question	Answer	Marks
	Section B	
2(a)	Error: • Problem with zero … • … stored in the negative number array // negative number count increases by 1 Correction: • Replace ELSE with IF • `IF Number < 0 (THEN)`	**4**
2(b)	Explanation: • Replace `REPEAT … UNTIL` with `WHILE … DO … ENDWHILE` • Change condition to `WHILE Number <> 9999 DO` • Add / Move `INPUT Number` to before loop // Move / Add extra `INPUT Number` at end of loop • Remove (`Count ← 0` and) `Count ← Count + 1` Or Any **four** from: • Include an `IF` statement after `INPUT Number` / before updating the arrays • `IF Number <> 9999 THEN …` or similar • Move output statements to be executed when `Number = 9999` • Change `UNTIL Count >= 50` to `UNTIL Number = 9999` • Remove (`Count ← 0` and) `Count ← Count + 1`	**4**

Question	Answer	Marks
3(a)	**One** mark for correct input (**all sets**) **One** mark for correct calculations (**all sets**) **One** mark for **each** correct output **Set 1:** 88, 74, 60 **Set 2:** 20, 33, 67 **Set 3:** 79, 91, 70	5
3(b)	Any **three** from: • Add extra decision box … • … in an appropriate position between the average calculation and the output • Check for average greater than or equal to 60 and less than 80 • Output Merit if average greater than or equal to 60 (and less than 80) … • … otherwise continue	3

Set 1: 88, 74, 60

Mark1	Mark2	Mark3	Total	Average	OUTPUT
88	74	60	222	74	Pass

Set 2: 20, 33, 67

Mark1	Mark2	Mark3	Total	Average	OUTPUT
20	33	67	120	40	Pass

Set 3: 79, 91, 70

Mark1	Mark2	Mark3	Total	Average	OUTPUT
79	91	70	240	80	Distinction

Question	Answer	Marks
4	**One** mark for explaining why a validation check is needed when data is input To check that data is sensible / reasonable / meets required criteria **One** mark for explaining why a verification check is needed when data is input To check that data is not changed on entry **One** mark for an example of a validation check Range check // Length check // Type check **One** mark for an example of a verification check Double entry // Visual check	**4**

Question	Answer	Marks
5(a)	One mark for data type and reason SIZE text, expressed as a single word SHAPE text, short phrase required WOOD text, expressed as a single word PRICE currency, needs to be expressed as dollars / may be used in calculations SOLD Boolean, only two choices	**5**

Question	Answer	Marks
5(b)(i)	**One mark** for • correct field and table names must include SIZE, WOOD and PRICE, allow SHAPE or SOLD • correct show for PRICE only • correct criteria to select small and walnut	3

Field:	SIZE	WOOD	PRICE	
Table:	BOX	BOX	BOX	
Sort:				
Show:	☐	☐	☑	☐
Criteria:	='small'	='walnut'		
or:				

Question	Answer	Marks
5(b)(ii)		2

Field:	WOOD	
Table:	BOX	
Sort:		
Show:	☑	(1)
Criteria:	='walnut' OR 'beech'	(1)
or:		

One mark for each correct alteration max 2