Cambridge IGCSE™

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

COMPUTER SCIENCE **0478/21**

Paper 2 Problem-solving and Programming **October/November 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 insurance system allows customers to check how much it would cost to buy insurance for a car for twelve months. The starting price of the car insurance is $500. The actual price a customer pays for the car insurance changes depending upon this information:
* engine size of car
* value of car
* where car is kept overnight
* number of kilometres (km) driven a year
* age of driver
* years insured without an insurance claim

Engine size in litres	Price change	Value in $1000	Price change	Kept over-night	Price change	1000 km driven a year	Price change	Age of driver	Price change
<=0.5	–5%	<0.5	–5%	Garage	–5%	<5	–5%	18–20	+100%
>0.5 to 1.0	0%	0.5 to 2	0%	Drive	0%	5 to 20	0%	21–25	+50%
>1.0 to 2.5	+5%	>2 to 10	+5%	Street	+5%	>20	+5%	26–30	+25%
>2.5	+10%	>10 to 20	+10%					31–70	0%
		>20	+15%					71–80	+10%
								>80	+20%

Years without claim	Price change
1	–10%
2	–20%
3	–30%
4	–40%
5	–50%
6	–60%
>6	–70%

Table 1   **Table 2**

The actual price is calculated by:
* finding the total of the percentage changes for the customer using Table 1
* applying this total percentage change to the starting price of the car insurance
* applying the years without claim discount percentage for the customer using Table 2.

Write and test a program or programs to calculate the price for a customer to insure a car.
* 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 to insure a car.

Write a program to obtain the required information from a customer and calculate the price to insure the car. Display the price to insure the car. Display the total percentage change calculated from Table 1, and the years without claim discount percentage from Table 2 separately.

Task 2 – New customer discount.

Extend **Task 1** to include an additional discount of 10% off the price to any new customer who is aged between 26 and 70 inclusive, who also has 2 or more years without a claim. Display the amount of money this would save and the new price.

Task 3 – Adding an extra driver.

Customers can add one extra driver. The age of the extra driver may increase the price. The new price is calculated by applying the percentage price change for the age of the extra driver from Table 1 to the price. Extend **Task 2** to calculate and display the new price including an extra driver if required.

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

    (a) Identify **one** array that you could have used for **Task 1**. Give the data type and state the use of the array.

         Array ....

         Data type ....

         Use ....

         ....

         ....
                                                                                                                            [3]

    (b) Data entered by a customer includes the age of the driver in **Task 1**.
         Identify **two** different validation checks you could use for the age of the driver.
         Justify your choice for each one.

         Validation check 1 ....

         ....

         Justification ....

         ....

         ....

         ....

         Validation check 2 ....

         ....

         Justification ....

         ....

         ....

         ....
                                                                                                                            [4]

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

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [6]

(d) (i) 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]

(ii) The car insurance system needs updating to record the percentage of customers who bought car insurance after obtaining a price to insure a car. Describe the changes required to your program.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [3]

Section B

2   Describe the purpose of variables and constants. Use an example of each in your answer.

....

....

....

....

....

....

....

....

.... [4]

3 This pseudocode algorithm calculates the weight and number of bags in a load of firewood. The weight in kilograms of each bag is input. The algorithm finishes when either 50 bags have been weighed, or as soon as the total weight exceeds 1000 kilograms. Only then are the total weight and the number of bags in the load output.

```
01  TotalWeight ← 1000
02  BagCount ← 0
03  MaxBag ← 50
04  MaxWeight ← 1000
05  REPEAT
06      OUTPUT "Please Enter weight of bag"
07      INPUT Weight
08      TotalWeight ← TotalWeight + Weight
09      BagCount ← BagCount + 1
10      OUTPUT "Number of bags in the load is ", BagCount
11  UNTIL TotalWeight > MaxWeight AND BagCount >= MaxBag
12  OUTPUT "Total weight of the load is ", MaxWeight
```

(a) Give the line number(s) from the algorithm of:

an assignment statement ....

a loop ....

a counting statement ....

a totalling statement ....
[4]

(b) Give the line numbers of the **four** errors in this pseudocode. Suggest a correction for each error.

Error 1 line number ....

Correction ....

....

Error 2 line number ....

Correction ....

....

Error 3 line number ....

Correction ....

....

Error 4 line number ....

Correction ....

....
[4]

(c) Explain how you could extend the algorithm to calculate and display the average weight of a bag of firewood in the load.

....

....

....

....

....

....

....

....

....

....

.... [4]

4 This flowchart inputs five numbers and performs a calculation.

The predefined function MOD finds the remainder from integer division for example
R ← 25 MOD 11 gives R a value of 3

```
                          ┌──────────┐
                          │  START   │
                          └──────────┘
                               │
                               ▼
                        ╱─────────────╲
                       ╱    INPUT       ╲
                       ╲  V,W,X,Y,Z      ╱
                        ╲───────────────╱
                               │
                               ▼
                         ◇─────────────◇
                        ╱     IS         ╲      Yes      ┌──────────┐
                        ╲    V = 0 ?     ╱─────────────▶ │   END    │
                         ◇─────────────◇               └──────────┘
                               │
                               │ No
                               ▼
           ┌───────────────────────────────────────────┐
           │  A ←  V*2 + W*3 + X*4 + Y*5                │
           └───────────────────────────────────────────┘
                               │
                               ▼
           ┌───────────────────────────────────────────┐
           │            B ←  A MOD 11                   │
           └───────────────────────────────────────────┘
                               │
                               ▼
      ╱──────────────╲   Yes  ◇─────────────◇
     ╱   OUTPUT        ╲◀──────╱     IS       ╲
     ╲   "Valid"       ╱       ╲    B = Z ?   ╱
      ╲───────────────╱         ◇─────────────◇
                                      │
                                      │ No
                                      ▼
             ╱────────────────────────────────╲
            ╱          OUTPUT                   ╲
            ╲          "Invalid"                ╱
             ╲────────────────────────────────╱
```
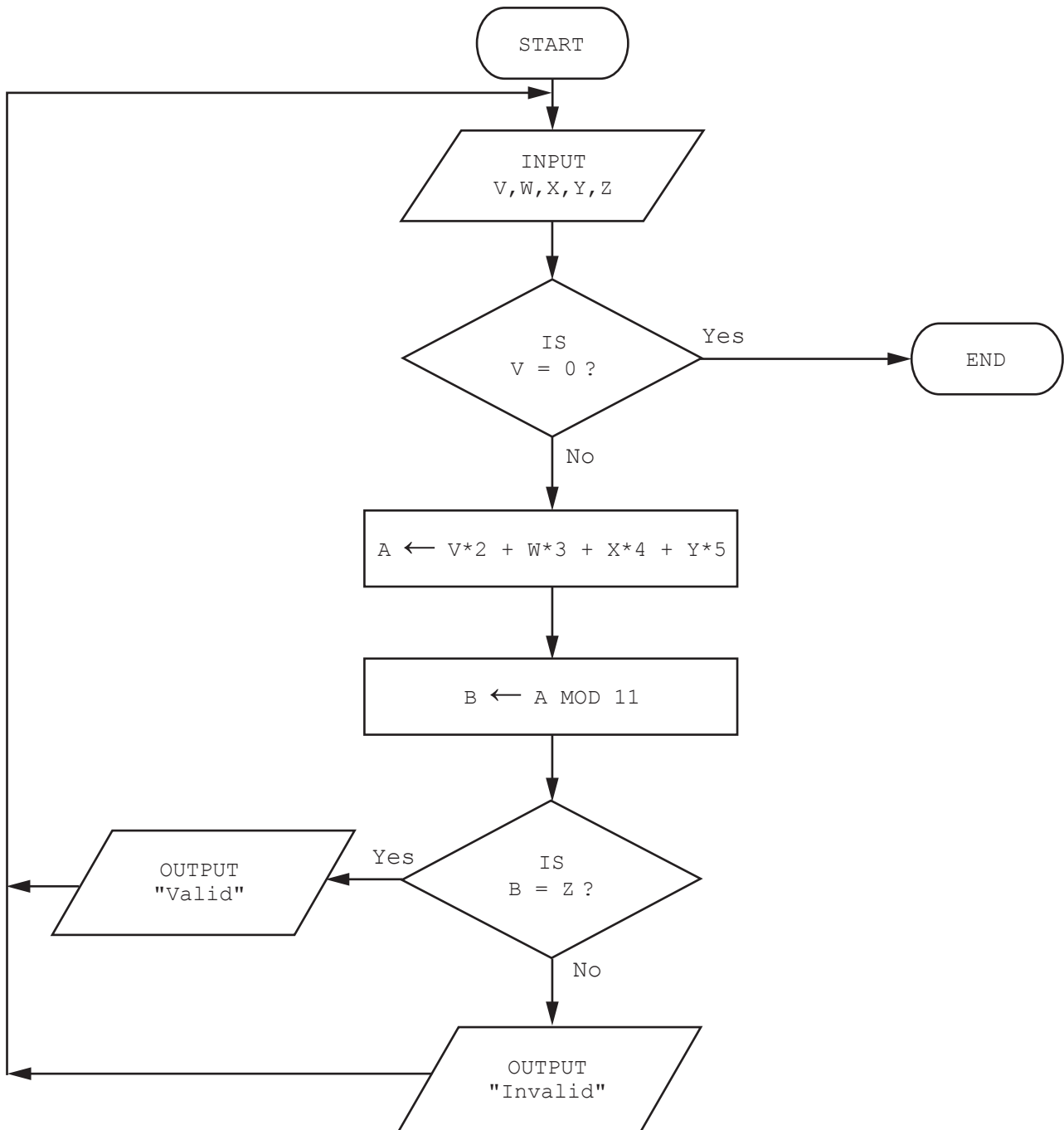
(a) Complete the trace table for this set of input data:
5, 4, 6, 2, 1, 9, 3, 2, 1, 6, 7, 6, 1, 5, 1, 0, 0, 0, 0, 0

V	W	X	Y	Z	A	B	OUTPUT

[4]

(b) Describe the purpose of this flowchart.

....

....

....

.... [2]

5 A database table, AIRLINE, stores data used to compare airlines.

Code	Airline name	Number of employees	Number of countries	Head office	Share price
FJ	FastJet	60000	30	Europe	514.50
MA	MurphyAir	35000	8	Europe	152.67
JS	JetSeven	45000	22	Asia	257.44
K3	Koala3	22000	11	Australia	501.21
NS	NorthState	30000	4	America	108.22
SS	SouthState	30000	4	America	126.35
BJ	BlueJet	15000	7	Africa	215.45
SK	SkyKing	32000	27	Europe	506.12
PF	PandaFly	50000	35	Asia	317.88

(a) State how many fields and how many records are shown in the AIRLINE table.

Number of Fields ....

Number of Records ....

[2]

(b) Show the output that would be given by this query-by-example.

Field:	Airline name	Number of employees	Head office	Share price
Table:	AIRLINE	AIRLINE	AIRLINE	AIRLINE
Sort:				
Show:	✓	☐	✓	☐
Criteria:		< 35000		> 500.00
or:				

....

....

.... [2]

(c) Complete the query-by-example grid to find every airline with a head office in Asia or Africa, and number of countries greater than 4. Only show the airline name and number of countries.

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

[4]

BLANK PAGE

BLANK PAGE

16

BLANK PAGE

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)	For **one** array **One** mark for naming of an array with a suitable data type and **one** mark for its use Array `yearsWithoutClaim` Data type integer Use to store the % discount Many correct answers, this is an example only.	**3**
1(b)	For **each** validation check **One** mark for identification of a check and **one** mark for the reason **max 4** Check presence check Reason need to enter age to calculate price Check type check Reason need to check age is a positive integer Many correct answers, these are examples only.	**4**

1(c)	Any **six** from:
	MP1 Input with prompt if new customer
	MP2 Check age greater than or equal 26
	MP3 Check age less than or equal 70
	MP4 Check number of years without a claim greater than or equal to 2 …
	MP5 … discount 10%
	MP6 Otherwise discount 0%
	MP7 Calculate new price and value of extra discount
	MP8 Output new price and value of extra discount, with suitable message
	
	Sample answer
	`PRINT "Are you a new customer Y/N?"`
	`INPUT new`
	`discount ← 0`
	`IF new = "Y" DO`
	`  IF age >= 26 AND age <= 70`
	`    THEN`
	`      IF yearsWithoutClaim >=2`
	`        THEN`
	`          discount ← price * 0.1`
	`          price ← price * 0.9`
	`      ENDIF`
	`  ENDIF`
	`  PRINT "New price", price`
	`  PRINT "Extra discount for new customer", discount`
	`ENDIF`
		6

Question	Answer	Marks
1(d)(i)	Explanation Any **four** from: MP1 Provide a prompt and input to check if additional driver required MP2 Check response with `IF/REPEAT/WHILE/CASE` statement for yes MP3 … input age of additional driver MP4 … Use of `IF / REPEAT` etc. to validate age MP5 … use the age input to find / look up appropriate price increase MP6 … description of how the price to pay now is calculated MP7 … description of how the price to pay now is displayed including the use of a suitable message For no MP8 … no output required or suitable message output Programming statements can be used but must be explained.	**4**
1(d)(ii)	Description Any **three** from: MP1 Use a variable (quotes) to keep a running total of the number of quotes provided MP2 Ask, using prompt with input, customers if they are going to purchase the insurance MP3 Use another variable (policies) to keep a running total of the number of insurances bought / not bought MP4 Use these to calculate the percentage policies/quotes*100	**3**

Question	Answer	Marks
Section B		
2	**One** mark for description of variable **One** mark for description of constant **One** mark for inclusion of an example variable **One** mark for inclusion of an example constant Example answer: A value that can change during the execution of a program A named value that cannot change during the execution of a program Variable example - using a counter for example `counter ← counter + 1` Constant example – a static value that can be used for checking for example `< maxAge`	**4**

Question	Answer	Marks
3(a)	Line 1/2/3/4/8/9 Lines 5–11 Line 9 Line 8	**4**

Question	Answer	Marks
3(b)	**One** mark for error and correction Line 1 `TotalWeight ← 0` Line 10 **move** `OUTPUT "Number of bags in the load is ",` `BagCount` **to end / after line 11** Line 11 `UNTIL TotalWeight > MaxWeight` **OR** `BagCount >= MaxBag` Line 12 `OUTPUT "Total weight of the load is ", ` **`TotalWeight`**	4
3(c)	Any **four** from: After line 11 Divide `TotalWeight` by ... `BagCount` Assign a new variable `AverageWeight ← TotalWeight / BagCount` Output the result `OUTPUT AverageWeight` With a message `"Average weight of a bag of firewood is "`	4

Question	Answer	Marks
4(a)	**One** mark for correct input columns (V, W, X, Y, Z) **One** mark for correct calculation column A **One** mark for correct calculation column B **One** mark for correct output column	4

V	W	X	Y	Z	A	B	OUTPUT
5	4	6	2	1	56	1	Valid
9	3	2	1	6	40	7	Invalid
7	6	1	5	1	61	6	Invalid
0	0	0	0	0			

Question	Answer	Marks
4(b)	• (Use first four digits input to) calculate a check digit • Check if the check digit input is valid	2

Question	Answer	Marks
5(a)	6 fields 9 records	2
5(b)	Either mark by row or by column but not both **One** each correct row or **one** mark for each correct column max **two** Koala3 Australia SkyKing Europe	2

Question	Answer	Marks
5(c)		**4**

Field:	Airline name	Number of countries	Head office	
Table:	AIRLINE	AIRLINE	AIRLINE	
Sort:				
Show:	☑	☑	☐	☐
Criteria:		> 4	="Asia"	
or:		> 4	="Africa"	

one mark for correct fields and correct table names, must have the three columns required, ignore another column provided the show row is not ticked and there are no criteria set for that column
one mark for correct show row
one mark for correct number of countries criteria
one mark for correct head office criteria